

UC Merced

UC Merced Electronic Theses and Dissertations

Title

Camera Shake Removal From One Single Image

Permalink

<https://escholarship.org/uc/item/9p46c1k2>

Author

Hu, Zhe

Publication Date

2015

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, MERCED

Camera Shake Removal From One Single Image

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering and Computer Science

by

Zhe Hu

Committee in charge:

Professor Ming-Hsuan Yang, Chair
Professor Roummel F. Marcia
Professor Shawn Newam
Doctor Jue Wang

2015

Copyright
Zhe Hu, 2015
All rights reserved.

The dissertation of Zhe Hu is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Professor Roummel F. Marcia

Professor Shawn Newsam

Doctor Jue Wang

Professor Ming-Hsuan Yang

Chair

University of California, Merced

2015

TABLE OF CONTENTS

	Signature Page	iii
	Table of Contents	iv
	List of Figures	vii
	List of Tables	xii
	Vita and Publications	xiii
	Abstract	xiv
Chapter 1	Introduction	1
	1.1 Overview	1
	1.2 Organization	3
Chapter 2	Literature Review	6
	2.1 Spatially Invariant Blur	6
	2.2 Spatially Variant Blur	8
Chapter 3	Single-Frame Deblurring with Adaptive Dictionary Learning	10
	3.1 Introduction	10
	3.2 Problem Formulation	11
	3.2.1 Estimating Sparse Coefficients	12
	3.2.2 Updating Dictionary	12
	3.2.3 Recovering Latent Image and Estimating Blur Kernel .	13
	3.2.4 Algorithm Analysis	14
	3.3 Experimental Results	15
	3.4 Conclusion	20
Chapter 4	Deblurring Low-light Images with Light Streaks	21
	4.1 Introduction	21
	4.2 Related Work and Context	24
	4.3 Light Streak Detection	26
	4.3.1 Candidate Patch Detection	26
	4.3.2 Finding the Best Light Streak Patch	27
	4.3.3 Finding Additional Light Streak Patches	29

4.4	Blur Kernel Estimation using Light Streaks	29
4.4.1	Updating K	30
4.4.2	Updating D_i	31
4.4.3	Updating I	32
4.4.4	Initialization and Implementation Details	32
4.5	Deconvolution with Ringing Suppression	33
4.6	Experimental Results	36
4.6.1	Comparisons with Hua and Low's Method	36
4.6.2	Comparisons on Real Images	37
4.6.3	Comparisons on Synthetic Images	37
4.6.4	Failure Cases	41
4.7	Non-Uniform Deblurring	45
4.7.1	Estimation of the Blur Parameter τ	46
4.7.2	Estimation of the Latent Image I	47
4.7.3	Light Streak Detection in Non-Uniform Case	47
4.7.4	Comparison Results	48
4.8	Conclusion	48
Chapter 5	Good Regions to Deblur	54
5.1	Introduction	54
5.2	Kernel Similarity	56
5.3	Learning Good Features	60
5.3.1	Learning Framework	60
5.3.2	Image Structure	62
5.3.3	Parameter Learning and Inference	64
5.3.4	Good Regions to Deblur	65
5.3.5	Contribution of Feature Components	66
5.4	Experimental Results	68
5.4.1	Deblurring with Good Regions	69
5.4.2	Contribution of Feature Components	78
5.4.3	Feature Alternatives	78
5.5	Conclusion	78
Chapter 6	Fast Non-uniform Deblurring using Constrained Camera Pose Sub-space	83
6.1	Introduction	83
6.2	Related Work	85
6.3	Non-uniform Blur Model for Camera Shake	86

6.4	Deblurring with Constrained Pose Subspace	87
6.4.1	Initialization using Backprojection	88
6.4.2	Estimating Pose Weights	90
6.4.3	Recovering Latent Image with Deconvolution	93
6.5	Experimental Results	93
6.6	Conclusion	97
Chapter 7	Joint Depth Estimation and Camera Shake Removal from Single Blurry Image	98
7.1	Introduction	98
7.2	Related Work	100
7.3	Depth-Aware Deblurring	102
7.3.1	EM Formulation	103
7.3.2	Image Partition	106
7.3.3	Depth Estimation	106
7.3.4	Constant-Depth Non-Uniform Deblurring	106
7.3.5	Initialization and Implementation Details	109
7.4	Experimental Results	111
7.4.1	Depth Estimation	111
7.4.2	Non-Uniform Deblurring	113
7.5	Conclusion	115
Chapter 8	Conclusion and Future Work	118
8.1	Summary	118
8.2	Future work	119
8.2.1	General Non-Uniform Model	120
8.2.2	Mobile Deblurring System	120
8.2.3	Joint Object Segmentation and Motion Deblurring	121
Bibliography		122

LIST OF FIGURES

Figure 3.1:	Experiments with motion blur kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.	16
Figure 3.2:	Experiments with motion blur kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.	17
Figure 3.3:	Experiments with randomly generated kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.	18
Figure 3.4:	Experiments with randomly generated kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.	19
Figure 4.1:	Deblurring a low-light image. The amount of salient image features that can be extracted from the low-light image is limited. This leads to the failure of the state-of-the-art methods [12, 66] on the image. The proposed method (d) improves the results by detecting and making use of the light streaks (the detected light streaks are shown in the boxes of (a)).	23
Figure 4.2:	Examples of light streak detection. The red box indicates the best light streak patch and the green boxes show additional light streak patches that are automatically identified.	27
Figure 4.3:	Non-blind deconvolution of a synthetic example. The second row displays the zoom-in view of the region in red box. The proposed method extends the Richardson-Lucy framework to suppress ringing artifacts, and handles outliers via pixel-wise weights.	35
Figure 4.4:	A comparison with Hua and Low’s approach [28]. The red box indicates our selected light streak patch and the green box is a manually selected patch.	38
Figure 4.5:	Comparisons with state-of-the-art methods on real examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows.	39

Figure 4.6:	Comparisons with state-of-the-art methods on real examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows.	40
Figure 4.7:	Cumulative error ratio histogram [40] on the synthetic dataset of uniform blur. There are 154 blurry images consisting of 11 low-light images and 14 blur kernels.	42
Figure 4.8:	Comparisons on synthetic examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows. . . .	43
Figure 4.9:	Comparisons on synthetic examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows. . . .	44
Figure 4.10:	Examples of failure cases. (a) a blurry image with failed light streak detection; (b) a deblurred image using the ground truth kernel; (c) a cropped region of (b).	45
Figure 4.11:	Example of light streak detection in non-uniform blur. (a) a spatially variant blurry image with directly applying light streak detection to it; (b) a spatially variant blurry image applying light streak detection and cross-validation.	49
Figure 4.12:	An example from the synthetic dataset of non-uniform blur. The image is generated by applying camera motion obtained from [33] to a low-light image, followed by adding Gaussian noise with 1% variation.	50
Figure 4.13:	Cumulative error ratio histogram [40] on the synthetic dataset of non-uniform blur. There are 165 blurry images generated from 11 low-light images and 15 camera motions from [33].	51
Figure 4.14:	Examples of non-uniform deblurring. (a)(g) blurry images with our detected light streaks; (b)(h) deblurred results by [12]; (c)(i) deblurred results by [59]; (d)(j) deblurred results by [66]; (e)(k) deblurred results by our method; (f)(l) our estimated kernels.	52
Figure 5.1:	Different regions lead to different kernel estimations and deblurred results. The top left image is the input blurry image with three sub-windows selected for estimating kernels. The other three images are the recovered images and estimated kernels from these three sub-windows using [18].	55
Figure 5.2:	Estimated kernels vary from shift to scale, but the dominant parts are the same. These kernels should be considered as the same by a good kernel similarity metric.	57

Figure 5.3:	The RMSE metric is not effective for kernel similarity. (a) the ground truth blur kernel; (b) the smooth estimated kernel, with RMSE 0.0013 compared to the ground truth; (c) the estimated kernel has an extra part, with RMSE 0.0016; (d) the estimated kernel misses some parts, with RMSE 0.0012.	59
Figure 5.4:	Compare RMSE and kernel similarity. (a) ground truth blur kernel; (b) - (d): three estimated kernels, and their RMSEs and kernel similarities compared with ground truth in Table 5.1.	59
Figure 5.5:	Spatial correlation. (a) input blurry image; (b) image reconstruction error map built using the estimated kernels from shifting subwindows (blue to red pixels indicate low to high reconstruction errors compared with the ground truth image); (c) kernel similarity map (blue to red pixels indicate high to low kernel similarity compared with the ground truth kernel).	61
Figure 5.6:	Texture regions are not always effective for kernel estimation. (a) the latent image. (b) the blurred image.	63
Figure 5.7:	Comparison on different window selection strategies for deblurring using [18]. (a) input blurry image and corresponding kernel; (b) & (c) user-selected subwindows (red box) and the deblurred results; (d) subwindow selected by window selector [18] and the deblurred result; (e) subwindow selected by window selector following [63] and the deblurred result; (f) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display).	71
Figure 5.8:	Comparison of different window selection strategies for deblurring using [49]. (a) input blurry image and corresponding kernel; (b) & (c) user-selected subwindows (red box) and the deblurred results; (d) subwindow selected by window selector [18] and the deblurred result; (e) subwindow selected by window selector following [63] and the deblurred result; (f) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display).	72
Figure 5.9:	Comparison on different window selection strategies for deblurring using [12]. (a)(f) input blurry image and corresponding kernel; (b)(g) the kernel estimation and deblurred result using the whole image; (c)(h) subwindow selected by window selector [18] and the deblurred result; (d)(i) subwindow selected by window selector following [63] and the deblurred result; (e)(j) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display). . .	74

Figure 5.10: Comparison on different window selection strategies for deblurring using [12]. (a)(f) input blurry image and corresponding kernel; (b)(g) the kernel estimation and deblurred result using the whole image; (c)(h) subwindow selected by window selector [18] and the deblurred result; (d)(i) subwindow selected by window selector following [63] and the deblurred result; (e)(j) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display). . . .	75
Figure 5.11: Examples of inferred region with top ten good subwindows to deblur using our algorithm. (a)(e) input blurred image and the corresponding blur kernel; (b)(f) the kernel estimation and deblurred result using the whole image; (c)(g) the mask image indicating the inferred good region; (d)(h) the deblurred result using the good region to deblur (best viewed on high-resolution display).	76
Figure 5.12: Cumulative error histograms on deblurred results with different window selection methods including ours, [18] and [63]. We use the algorithm in [12] for kernel estimation and the algorithm in [49] for non-blind deconvolution.	77
Figure 5.13: Inferred top subwindows from our algorithm are different for the images of the same scene but with different blur kernels. (a)(e) input blurry image and corresponding blur kernel; (b)(f) the kernel estimation and deblurred result using the whole image; (c)(g) the mask image indicating the inferred top subwindow; (d)(h) the deblurred result using the good region to deblur (best viewed on high-resolution display). . . .	79
Figure 5.14: Inferred regions from our algorithm are different for the images of the same scene but with different blur kernels. (a)(e) input blurry image and corresponding blur kernel; (b)(f) the kernel estimation and deblurred result using the whole image; (c)(g) the mask image indicating the inferred good region; (d)(h) the deblurred result using the good region to deblur (best viewed on high-resolution display). . . .	80
Figure 5.15: Cumulative error histograms on deblurred results with different feature extraction approaches. We use the algorithm in [14] for kernel estimation and the algorithm in [9] for non-blind deconvolution. . . .	81
Figure 6.1: Main steps of the proposed algorithm.	88

Figure 6.2:	Initialization using backprojection with synthetic data. Light green cubes indicate small weight values and dark blue ones indicate large ones. (a) synthesized blurry image. (b) its corresponding camera motion. (c) initial camera motion using backprojection with ground truth blur kernels at several image sites. (d) initial camera motion using backprojection with blur kernels estimated by a uniform deblurring method [12].	95
Figure 6.3:	Comparison with state-of-the-art single image deblurring methods for spatially variant blur. Top to bottom: <i>Pantheon</i> , <i>Gas station</i> , <i>Books</i> and <i>Butcher shop</i>	96
Figure 7.1:	Depth-aware non-uniform deblurring. (a) blurry image; (b) restored image by the method [24] assuming constant scene depth; (c) estimated depth map by our method; (d) restored image by our method using the estimated depth map.	99
Figure 7.2:	Image partition using matting. (a) blurry image and patches that are difficult for image partitioning, (b) one layer using matting and the corresponding matting patches.	107
Figure 7.3:	Synthetic example of the depth estimation in the first iteration. Dark region indicates small depth. (a) synthetic blurry image; (b) ground truth depth; (c) estimated latent image after smoothing filter; (b) our estimated depth using (c).	110
Figure 7.4:	Comparison with depth-from-single-image method [48] on a real image. (a) blurry image; (b) estimated depth by [48]; (c) our estimated depth.	111
Figure 7.5:	Depth estimation results on synthetic images. (a) sharp image; (b) synthesized blurry image caused by both translational and rotational blur; (c) synthesized blurry image caused by only rotational blur; (d) ground-truth depth; (e) our estimated depth from (b); (f) our estimated depth from (c).	112
Figure 7.6:	Depth-aware non-uniform deblurring on synthetic images. (a) synthetic blurry image; (b) deblurred result of [24]; (c) recovered image using our method; (d) our estimated depth (best viewed on a high-resolution display).	115
Figure 7.7:	Depth-aware non-uniform deblurring on real images. Top to bottom: <i>Books</i> , and <i>Boxes</i> examples (best viewed on high-resolution displays).	116

LIST OF TABLES

Table 4.1:	Quantitative comparisons using kernel similarity (KS).	37
Table 5.1:	Comparison between RMSE and kernel similarity (KS).	58
Table 7.1:	Average error with respect to the minimal number of translated pixels in the camera motion.	113

VITA

2009	B. S. in Mathematics, Zhejiang University, Hangzhou, China
2015	Ph. D. in Electrical Engineering and Computer Science, University of California, Merced

PUBLICATIONS

Zhe Hu and Ming-Hsuan Yang, *Learning good regions to deblur*, in International Journal of Computer Vision 2015.

Jinshan Pan*, Zhe Hu* (* indicates equal contribution), Zhixun Su and Ming-Hsuan Yang, *Deblurring face images with exemplars*, in European Conference on Computer Vision 2014.

Zhe Hu, Sunghyun Cho, Jue Wang and Ming-Hsuan Yang, *Deblurring low-light images with light streaks*, in IEEE Conference on Computer Vision and Pattern Recognition 2014.

Zhe Hu, Li Xu and Ming-Hsuan Yang, *Joint depth estimation and camera shake removal from single blurry image*, in IEEE Conference on Computer Vision and Pattern Recognition 2014.

Jinshan Pan, Zhe Hu, Zhixun Su and Ming-Hsuan Yang, *Deblurring text images via L_0 -regularized intensity and gradient prior*, in IEEE Conference on Computer Vision and Pattern Recognition 2014.

Zhe Hu and Ming-Hsuan Yang, *Good regions to deblur*, in European Conference on Computer Vision 2012.

Zhe Hu and Ming-Hsuan Yang, *Fast non-uniform deblurring using constrained camera pose subspace*, in British Machine Vision Conference 2012.

Zhe Hu, Jia-Bin Huang and Ming-Hsuan Yang, *Single image deblurring with adaptive dictionary learning*, in IEEE International Conference on Image Processing 2010.

ABSTRACT OF THE DISSERTATION

Camera Shake Removal From One Single Image

by

Zhe Hu

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California Merced, 2015

Professor Ming-Hsuan Yang, Chair

Image blur is one of the most fundamental and challenging problems in photography. It causes significant image degradation, especially in the low light conditions where longer exposure time is required. Although the blur effect can be reduced by setting a faster shutter speed, it inevitably gains higher levels of noise. Moreover, most users may not be able to take high-quality pictures due to hand shaking, dim lighting conditions and inappropriate shutter speed selections. Therefore, it is important to develop algorithms to remove image blur computationally.

To remove the image blur, we need to estimate the blur kernel, which depicts how the image is blurred, and recover the sharp image. The image deblurring problem is an ill-posed problem as many pairs of the sharp image and the blur kernel could result in a same blurry image. To distinguish the correct pair from others, additional information is required. We introduce a generic image prior and a specific blur motion prior in this thesis to address this issue. Moreover, we explore the influence of image structures on estimating blur kernels, and provide insights on the design of deblurring systems.

Blur kernel estimation turns out to be especially difficult in non-uniform cases, which are the common situations in practice. The presence of camera rotations during exposure would lead to non-uniform blur effect. Scene depth is another factor of non-uniform blur,

but is often neglected in recent methods. In this thesis, we introduce a unified framework for joint restoration of scene depth and latent image from a single input. To solve the complex model of non-uniform blur, we propose an efficient deblurring algorithm using backprojection and constrained camera pose subspace to facilitate fast convergence and low computation.

Chapter 1

Introduction

This chapter introduces the problems addressed in this dissertation and the approaches presented in my research work.

1.1 Overview

Motion blur is one of the most fundamental problems in computer vision. Motion blur artifacts are generated from relative motion between a camera and a scene during the exposure time. It causes significant image degradation, especially in the low light conditions where long exposure time is required. Although the blur effect can be reduced by setting a faster shutter speed, it inevitably gains higher levels of noise. Therefore, it is important to develop algorithms to remove image blur effectively and efficiently.

There are mainly four sources that could result in blurring: camera shake, object motion, out-of-focus blur and spherical aberration. Camera shake happens during the exposure time and blurs the recorded image according to the camera's movement. The modern camera system adopts a mechanical image stabilization hardware to compensate the movement of the imaging device. However, it is inevitable to have some blurs when taking photos in a dark scene or with a telephoto lens. This dissertation will focus on camera shake

removal with a single input image. The movement of objects in the scene is another source of blur, and the motion of different objects would make it more complicated. Out-of-focus blur happens when the lens has a shallow depth of field and the interest object is not in focus. It is a useful technique in photography and can be synthesized with a given focus setting. Spherical aberration, due to the spherical shape of the lens, would result in less blur at the center of the image and more blur near the border. This can be resolved by pre-calibrating the lens of the device.

Recovering the clear image from one single blurred image has been studied extensively with a rich literature. Early work focuses on spatially invariant blur in which the blur is uniform everywhere in the image. Typically, this blur process is modeled as a latent image convolved with a spatially invariant blur kernel K (i.e., the point spread function):

$$B = K \otimes I + n, \quad (1.1)$$

where B , I and n represent the input blurry image, unblurred (or latent) image and noise respectively. Within this formulation, $\otimes$ denotes the convolution operator and the deblurring problem is thus posed as a deconvolution problem. When the blur kernel K is assumed to be known explicitly prior to the deconvolution procedure, this problem is known as the classical linear image restoration problem and called non-blind deconvolution problem. Existing techniques for non-blind deconvolution can be categorized as inverse filtering, least-square filtering, recursive Kalman filtering and iterative methods. The Wiener filter [62] and Richardson-Lucy algorithm [47, 44] are two widely used non-blind deconvolution methods due to their simplicity and efficiency. Nevertheless, these two methods often introduce serious ringing effects. In most of the real-world cases, we do not know the exact kernel for a blurred image, and the deblurring problem with unknown kernel is posited as a blind-deconvolution problem. Blind deconvolution is a well-known ill-posed problem, and deblurring methods within this formulation aim to recover images with missing high frequency details. Since countless pairs of latent image and blur kernel can result in the blurred image, prior knowledge or additional information are often required for effective

solutions. Priors depict the properties of natural images so that constraints can be used in the optimization formulation to better recover the latent image. Along this research line, I aim to address the two important issues listed below.

- What are good priors for the image deblurring problem?
- How can one determine good image structures for deblurring?

In the thesis, we present three methods related to these two problems. We provide a preview of the proposed approaches in Section 1.2.

The model of spatially invariant blur is only sufficient if the camera shake is caused by in-plane translation (i.e., within the sensor plane) without any out-of-plane rotations. However, this assumption does not hold in general. Camera tilt and rotation happen frequently and the image blur becomes non-uniform, i.e., the blurs at different pixel locations is not the same. Moreover, object movement in the scene also leads to non-uniform blur effect. Blur kernel estimation turns out to be especially difficult in the non-uniform case. Early work addresses this problem by restricting the blur process to simple cases. Recently, a geometric model has been proposed to model general non-uniform blur caused by camera motion at the expense of high computational cost. Although progress has been made, there still remain several problems for non-uniform blur.

- How do we model the spatially varying blur in a more general framework?
- How do we solve the complex model efficiently?

We briefly overview two of the proposed methods along this research line in Section 1.2.

1.2 Organization

The following chapters are organized as follows.

In Chapter 3, we propose an algorithm that enforces sparse constraints and exploits the rich information contained in the blurred image. In our formulation, each image patch

is encoded with sparse coefficients using an over-complete dictionary. The sparsity constraints facilitate the latent image recovery without solving an ill-posed deconvolution problem. In addition, the dictionary is learned and updated directly from the given blurry image without using additional images. The proposed method iteratively utilizes sparsity constraints to recover the latent image, estimate the deblur kernel, and update the dictionary directly from a single image.

In Chapter 4, we make use of a special phenomenon, light streaks, that often appears in low-light images to help with image deblurring. We propose a new deblurring framework to automatically detect light streaks using the spectrum of the blur kernel. The detected light streaks are then posed as the constraints for kernel estimation, and are used to regularize the deblurring process.

Other than image priors for generic cases, it is also worthwhile to pay attention to the influence of image structures on estimating blur kernels. Instead of determining good image structures from empirical understanding, a learning based method is proposed for the task in Chapter 5. We first propose a metric that quantitatively measures the similarity between kernels, which facilitates the process of labeling good estimated kernels. Then we pose the learning problem within a conditional random field framework in order to exploit contextual constraints among image regions.

Blur kernel estimation turns to be especially difficult in the non-uniform case. Recently, a geometric model has been proposed to model general non-uniform blur caused by camera motion at the expense of high computational cost. To alleviate the heavy computational load, we propose a fast single image deblurring method to remove non-uniform blur in Chapter 6. In this work, we initialize the camera motion using backprojection from estimated local kernels at different image regions and estimate the motion weights iteratively by constraining the possible camera poses in a low-dimensional subspace.

Depth is another important factor in non-uniform blurs. However, recent methods often neglect it for simplicity. To address this problem, we propose a non-uniform image deblurring framework to jointly estimate the scene depth and remove the camera shake

in Chapter 7. To this end, we present a unified layer-based model for depth-involved deblurring. We provide a novel layer-based solution using matting to partition the layers and an expectation-maximization scheme to solve this problem.

Chapter 2

Literature Review

In this chapter, we review the literature related to the research work in terms of the blur models, spatially invariant models and spatially variant models.

2.1 Spatially Invariant Blur

The problem of removing spatially invariant blur has been studied extensively and numerous algorithms have been proposed [18, 46, 38, 56, 40, 12, 63, 15, 16]. Since blind deconvolution is an ill-posed problem, prior knowledge or additional information is often required for effective solutions. In the image deblurring literature, two types of additional information are often used: natural image priors and additional image observations.

One line of research focuses on exploring image priors for deblurring. In [18], the heavy-tailed gradient distribution of natural images is exploited as prior information. A mixture of Gaussians is used to fit the distribution of the gradient magnitudes of natural images. A sparse gradient prior is also used to search for blur kernels in [38]. In [30] a method is presented to exploit the underlying relation between motion blur and blurry object boundaries, which is shown to facilitate better kernel estimation. In [49], a deblurring algorithm is proposed in which prior knowledge regarding the gradients of natural images

is used with additional constraints on the consistence of local smooth regions before and after blurring. The consistency constraints are shown to be effective in suppressing ringing effects. A method that uses sparsity constraints for both the blur kernel and the latent image in the wavelet domain is presented in [6]. In contrast to existing methods that exploit the heavy-tailed gradient distribution of natural images, an image restoration algorithm that applies adaptive priors based on texture contents is proposed in [15]. Experimental results on denoising and deblurring show that adaptive priors are important for deblurring results when patches at different image locations are manually selected.

Another line of research tackles image deblurring by leveraging auxiliary information acquired at capture time [46, 69, 39, 9, 4, 55]. With both a low-resolution video camera and a high-resolution still-image camera, an algorithm that utilizes both spatial and temporal information is proposed [3] for effective image deblurring. On the other hand, noisy images also provide useful information for image deblurring. When a pair of blurred and noisy images of the same scene are available, it has been shown that the blur kernel can be estimated using the sharp image structures in the noisy image [69]. Numerous studies focus on exploiting additional information, e.g., priors and image sequences to facilitate image deblurring. Considerably less attention has been paid to exploit image structure for kernel estimation and deblurring.

Recent deblurring algorithms can be categorized into four groups: maximum a posteriori (MAP), variational Bayesian, edge prediction, and power-spectrum methods. The MAP based methods [49, 15, 35, 66] model the problem by maximizing the posterior probability with respect to the latent image and the blur kernel. The priors of the latent image and the blur kernel are then used as regularizations in the objective function. Since multiple solutions exist for the objective function, the success of these methods hinges on the choice of the regularizations. However, MAP based methods may generate trivial solutions (i.e., delta kernel functions) [40] even with the carefully designed priors. The variational Bayesian based method [18, 40] approximates the posterior probability of the unobserved variables and computes the marginal likelihood. This approach

has been shown to be theoretically more robust than the MAP based methods [40]. The edge prediction based methods [32, 12, 63] use explicit or implicit salient edge restoration for kernel estimations, which have been demonstrated as useful tools for alleviating delta kernel solutions with fast convergence. Despite demonstrated success [33], these methods rely largely on heuristic edge selections for kernel estimations. The power-spectrum based algorithms [68, 19, 25] are developed based on the observation that the power spectra of sharp natural images can be well modeled by statistical regularities. Thus, the power spectrum of a blur kernel can be estimated from that of a blurry image. Phase retrieval techniques are used to estimate the phase component of the power spectrum which requires an image region containing a distribution of diverse edge orientations for kernel estimation.

2.2 Spatially Variant Blur

Removing spatially variant blur is a challenging task in which patches are degraded differently in an image. This problem is attracting more attention nowadays due to its wide range of practical applications [2, 13, 50, 56, 31, 60, 21, 24, 57, 29]. Early work splits an image into several regions where each is modeled with a constant PSF and recovered by a uniform deblur algorithm [2, 13]. However, these methods are sensitive to region size which is required to be sufficiently small for the assumption of uniform PSFs to be hold. Hybrid camera systems have also been used to capture a local blur kernel at each pixel location based on optical flow [56, 31]. Information about camera motion has also been obtained using inertial sensors such as gyroscopes and accelerometers, which are integrated into many modern mobile devices. Šindelář and Šroubek [51] proposed a deblurring system for smartphones that synthesizes local blur kernels based on gyroscope readings. The system compares favorably to recent deblurring algorithms, but requires user interaction due to the noisy data and the lack of synchronization between the camera and gyroscope sensors. Joshi et al. [31] built an elaborate DSLR-based camera system

with a gyroscope and accelerometer to record camera motion. The inertial sensors are assumed to have been calibrated and synchronized beforehand. Methods with additional inputs are demonstrated to be effective for image deblurring, but such approaches are not applicable to single image non-uniform deblurring.

Recently, a geometric model has been proposed to model the observed blurry image as the integration of all the intermediate images captured by the camera along the motion trajectory [60, 57]. These intermediate images are modeled as transformed copies (i.e., homographies) of the sharp image. Based on this model, image blur caused by camera motion can be formulated as an optimization problem. It is also possible to remove spatially varying blur by estimating a general camera motion function. A similar model has been used to model three degrees of camera motion (with in-plane translation and rotation) in [21]. To speed up the optimization step, fast patch-based non-uniform deblurring methods have been developed [22, 24, 26]. In [33], a dataset for motion deblurring is constructed by recording and playing back the camera shake with a robot system. The blurry images are obtained by taking photos on sharp pictures which assume constant depth. Notwithstanding the demonstrated success, the above-mentioned non-uniform deblurring methods do not consider the effect of scene depth on camera motion blur or assume constant scene depth for simplicity.

To exploit depth information in the blurry input, recent methods [13, 50, 8] partition an image into foreground/background and remove the blur caused by object motion. A coded aperture method has been proposed for deblurring and decoded images are used to infer pixel-wise depth in a Markov random field [39]. Depth effects on translational blur has also been discussed in [52] where the blur is estimated from the user specified region with approximately constant depth to infer depth. In [64], two blurry images of the same scene are used to infer depth with stereo techniques and further improve the blur removal. Similarly, a non-uniform motion deblurring method has been proposed to exploit bilayer-scene constraints from two blur observations [45].

Chapter 3

Single-Frame Deblurring with Adaptive Dictionary Learning

3.1 Introduction

The main challenge in image deblurring stems from the under constrained formulation. Countless clear image/blur kernel pairs could result in the same blurry image. Recent algorithms focus on exploiting additional information, e.g., priors and image sequences to distinguish the optimal solution from others. Gradient distribution priors of nature images and a coarse-to-fine framework are utilized for kernel estimation in [18]. In [69], a method that utilizes additional information from a pair of blurry and noisy images is proposed to estimate the kernel. Recently, an algorithm that models the deblurred blur kernel and image with sparsity constraints has demonstrated promising results [6]. Nevertheless, this method does not exploit the information contained in the blurred image as generic bases (i.e., curvelets and framelets) are used. Most of the priors introduced to this field model the generic properties of natural images. However, the rich and specific information contained in a blurred image is not exploited for deblurring.

In this chapter, we present an algorithm that enforces sparse constraints and exploits

the rich information contained in the blurred image. In our formulation, each image patch is encoded with sparse coefficients using an over-complete dictionary. The sparsity constraints facilitate recovery of the latent image without solving an ill-posed deconvolution problem. In addition, the dictionary is learned and updated directly from one single frame without using additional images. The proposed method iteratively utilizes sparsity constraints to recover the latent image, estimate the blur kernel, and update the dictionary directly from one single frame.

3.2 Problem Formulation

It is well-known that natural images can be modeled well with sparse representations over an over-complete dictionary. A signal (i.e., an image patch in this work) I can be approximated by a sparse linear combination of atom signals in an over-complete dictionary $D \in \mathcal{R}^{n \times m}$, $n, m \in \mathcal{Z}$ and $n < m$, or a set of overcomplete bases $\{d_j\}_{j=1}^m$ and a coefficient vector $\alpha \in \mathcal{R}^m$, that satisfy:

$$\min \|\alpha\|_0 \quad \text{subject to} \quad I = D\alpha, \quad (3.1)$$

or

$$\min \|\alpha\|_0 \quad \text{subject to} \quad \|I - D\alpha\|_2 \leq \epsilon, \quad (3.2)$$

where I is an image patch, which has the same dimension as the dictionary basis and $\|\cdot\|_k$ is the l_k norm. In other words, any signal can be described using a small number of dictionary bases under the corresponding bases pool.

We pose image deblurring as a joint optimization problem that estimates the blur kernel K and sparse coefficients α with regularization:

$$\min_{K, \alpha} \|B - K \otimes DA\|_2^2 + \lambda \sum \Theta(\alpha), \quad (3.3)$$

where each column of B is a raster scan vector of a block from the blurred image, and each column of A is the sparse coefficient α for a block of the latent image (from (3.1))

or (3.2)). In addition, $\Theta(\alpha)$ is the l_1 regularization term upon the sparse representation coefficient α as it has been proved to be equivalent to the l_0 norm under certain conditions [17]. In (3.3), there are three unknown variables: the kernel K , the coefficient vector α , and the underlying dictionary D . Therefore, a direct approach to solve this optimization problem is likely to suffer from slow convergence and local minimums. As the main goal is to estimate the blur kernel, we propose an iterative method to alternately estimate the unknown variables, one at a time, which divides the optimization problem into several simple ones in each iteration. More importantly, the dictionary D is learned from the input image during this optimization process. The proposed algorithm iteratively optimizes one of K , D , α by fixing the other two, and finally obtains the deblurring kernel. With the estimated kernel, we can apply any standard deconvolution algorithm to recover the latent image.

3.2.1 Estimating Sparse Coefficients

At the beginning of each iteration, we fix K and D to estimate the coefficient α of each image patch, the column of A

$$\begin{aligned}\alpha^{(n+1)} &= \min_{\alpha} \|b - (K^{(n)} \otimes D^{(n)})\alpha\|_2^2 + \lambda\|\alpha\|_1 \\ &= \min_{\alpha} \|b - \hat{D}^{(n)}\alpha\|_2^2 + \lambda\|\alpha\|_1,\end{aligned}\tag{3.4}$$

where $\hat{D}^{(n)}$ is the blurred dictionary at iteration n . Here, we use b to represent a block of the blurry image. At the first iteration, we need to initialize K_0 , which will be discussed in Section 3.3. As K and D are fixed at iteration n , there is only one unknown variable and the optimization problem can be solved efficiently using basis pursuit algorithms [11].

3.2.2 Updating Dictionary

In this step, as the kernel K and coefficients α are fixed, we update the dictionary atoms of the latent image based on the test image itself. However, the extracted information from

the blurred image B cannot be directly applied to update the underlying dictionary of the clear image. Before the update of dictionary D , we reconstruct an image block $\hat{I}$ by deconvolving the blurred image with the estimated kernel $K^{(n)}$ using a non-blind deconvolution algorithm, such as the Richardson-Lucy or Wiener deconvolution algorithms, and then use $\hat{I}$ to update the dictionary D of latent image. Similar to the K-SVD algorithm [1], we decompose the multiplication $D\alpha$ to the sum of rank-1 matrices and each time update a basis d_i of dictionary D :

$$\begin{aligned} d_i^{(n+1)} &= \min_{d_i} \|\hat{I} - D^{(n)}\alpha^{(n+1)}\|_2^2 \\ &= \min_{d_i} \|\hat{I} - (d_i\alpha_i^{(n+1)} + \sum_{j \neq i} d_j^{(n)}\alpha_j^{(n+1)})\|_2^2 \\ &= \min_{d_i} \|E_i^{(n+1)} - d_i\alpha_i^{(n+1)}\|_2^2, \end{aligned} \quad (3.5)$$

where $E_i^{(n+1)} = \hat{I} - \sum_{j \neq i} d_j^{(n)}\alpha_j^{(n+1)}$ is the residual matrix.

3.2.3 Recovering Latent Image and Estimating Blur Kernel

Similar to the ideas proposed in [43], we assume the sparse coefficients α of the latent image I with basis $\{d_i\}_{i=1}^m$ are the same as the ones of blurred image B relative to the blurred basis $\{b_i\} = \{K \otimes d_i\}_{i=1}^m$. Thus, we can reconstruct the latent image as

$$I^{(n+1)} = D^{(n+1)}\alpha^{(n+1)}. \quad (3.6)$$

Next, we can write the convolution $K^{(n+1)} \otimes I^{(n+1)}$ as a matrix multiplication $A \cdot x$ (here x corresponds to $K^{(n+1)}$ and image blocks I correspond to the columns of A) and formulate the kernel estimation problem as

$$\min \|B - Ax\|_2^2. \quad (3.7)$$

Moreover, we introduce additional regularization constraints in order to get a stable solution from Tikhonov regularization,

$$\min \|B - Ax\|_2^2 + \|\Gamma x\|_2^2, \quad (3.8)$$

where Γ is the Tikhonov matrix. With Tikhonov regularization, the numerical solution, denoted by $\hat{x}$, is given by

$$\hat{x} = (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top B. \quad (3.9)$$

Alternatively, we can use objective functions with a l_1 norm penalty [49] or framelet system [6], such as

$$\min \|B - Ax\|_2^2 + \|x\|_1, \quad (3.10)$$

to estimate K . As mentioned above, the final deblurred image can be recovered once the blur kernel is estimated.

3.2.4 Algorithm Analysis

Our algorithm focuses mainly on estimating the blur kernel and then employing a standard non-blind deconvolution algorithm to recover the latent image. The success of our algorithm hinges on the effectiveness and accuracy of the kernel estimation. That is, if the accuracy of the estimated kernel is improved at each iteration, the proposed algorithm will finally find a reasonably good solution. This condition can be achieved under certain situations. For the minimization problem (3.4), we have $\hat{D} = K_e \otimes D_I$, where K_e and D_I represent the estimated kernel and dictionary for the latent image I , and $K_r = K_e + E$, where K_r is the real kernel and E is the kernel error. If this minimization problem is solved precisely, $B = (K_r - E) \otimes D_I \alpha$. Meanwhile, $B = K_r \otimes I$, then

$$\begin{aligned} K_r \otimes I &= (K_r - E) \otimes D_I \alpha \\ \Rightarrow K_r \otimes (I - D_I \alpha) &= E \otimes D_I \alpha, \end{aligned} \quad (3.11)$$

Based on the assumption of the coefficients α stated above, if the dictionary D_I for the latent image is correctly estimated ($D_I \alpha \approx I$), then the error $\|I - D_I \alpha\|_2$ will be decreased. Although both sides of the above equation have convolution operations, from the definition of convolution, we can still believe that the variation trend of $\|E\|_2$ is being decreased. That is, the estimate of the blur kernel is improved over each iteration and thus the latent image can be recovered.

3.3 Experimental Results

We implement our algorithm and carry out experiments in MATLAB. The initial kernel K_0 is set to be the Gaussian kernel with $\sigma = 1$, and λ in (3.4) as well as Γ in (3.9) are set as 1 and the identity matrix I . We use gray-scale images for our experiments and crop a small portion (e.g. 200×200 pixels) of the test image to estimate the kernel using the algorithm listed in Section 3.2. We use the Richardson-Lucy deconvolution algorithm to reconstruct $\hat{I}$ in Section 3.2.2. Each iteration of our algorithm takes approximately 5 minutes on a windows PC with a 2.67 GHz CPU and 4 GB RAM.

We compare our method with two state-of-art algorithms [18, 49] which utilize prior information on image edges. As shown in Figure 3.1 and Figure 3.2, although the deblurred images from [18, 49] have shaper edges, they also contain noise in smooth regions such as walls and floors. In contrast, our algorithm is able to recover the deblurred images effectively in smooth and textured regions (e.g., fluff of the koala, stripes of tablecloth and kerchief) with less ringing effects. One possible explanation for these experimental results is that the assumed prior distribution of edges used in [18, 49] does not hold well for these images whereas our method exploits the information from the blurred image.

We also carry out experiments with images blurred by randomly generated kernel. The existing deblurring algorithms are usually developed to deal with motion blur problems in which the kernels are oriented and simple. However, camera shake is complex and cannot be modeled with simple blur kernels. We evaluate our algorithms against others using randomly generated kernels. Figure 3.3 and Figure 3.4 show that our algorithm is able to recover the latent image with more detail and better contrast than [18] (which does not capture image details) and [49] (which suffers more ringing effects with fewer details). We also provide root-mean-square-error (RMSE) to demonstrate the quantitative comparison among the recovered images by different algorithms.

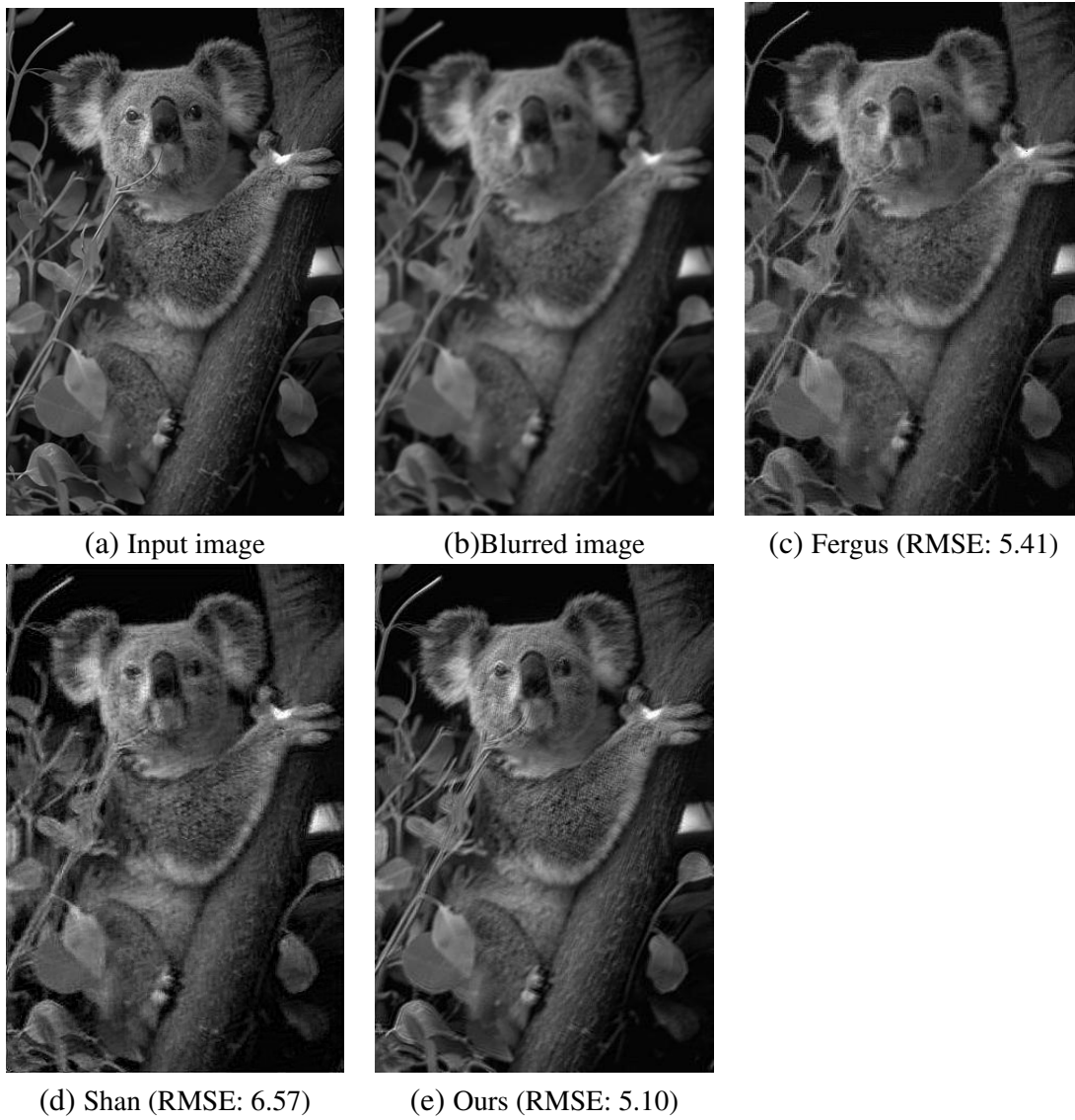


Figure 3.1: Experiments with motion blur kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.



Figure 3.2: Experiments with motion blur kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.



(a) Input image



(b) Blurred image



(c) Fergus (RMSE: 7.87)



(d) Shan (RMSE: 7.46)



(e) Ours (RMSE: 6.73)

Figure 3.3: Experiments with randomly generated kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.

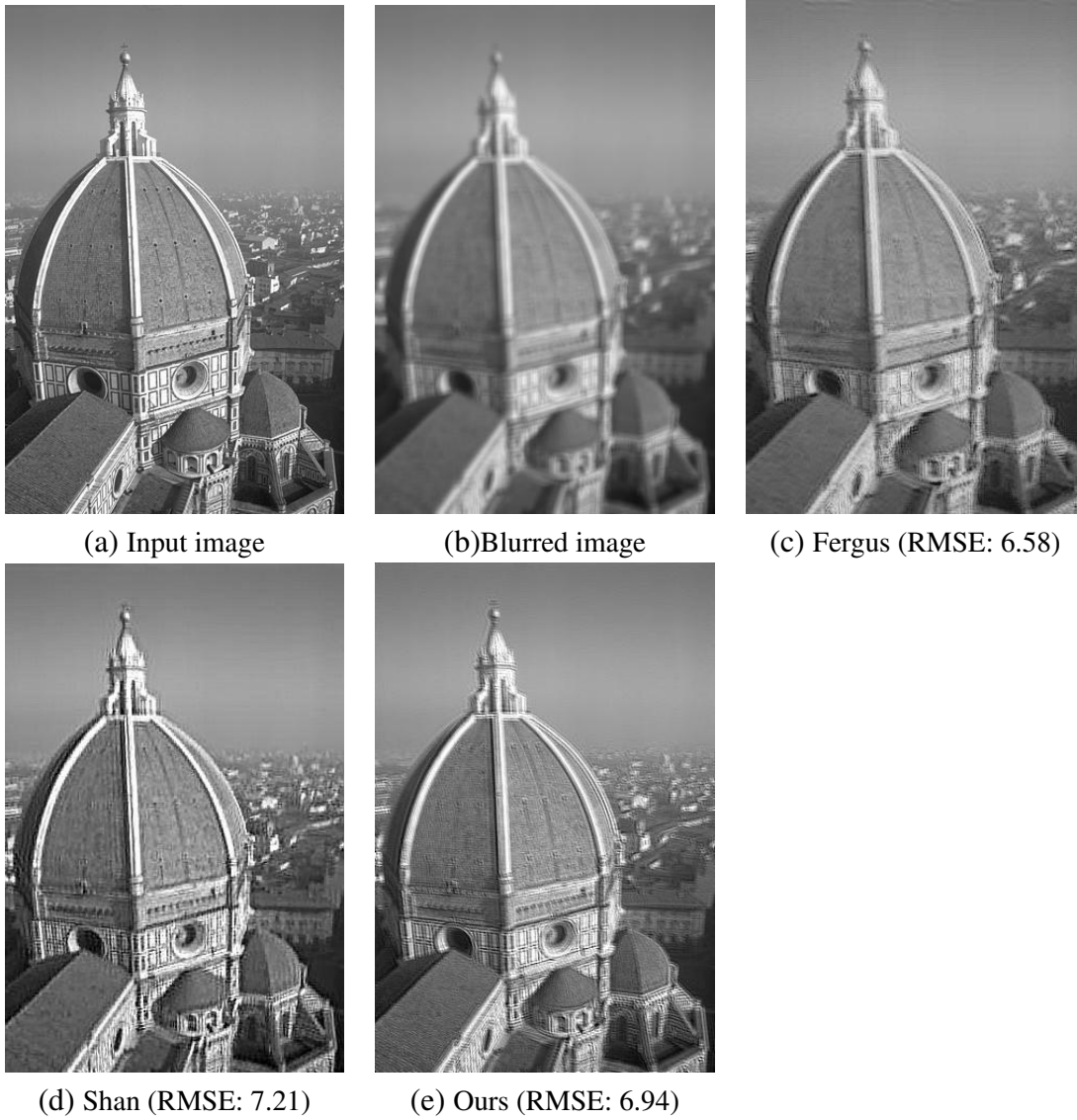


Figure 3.4: Experiments with randomly generated kernel. (a): input image; (b): blurred image; (c): recovered image using approach by Fergus et al. [18]; (d): recovered image using approach by Shan et al. [49]; (e): recovered image by our algorithm.

3.4 Conclusion

In this chapter, we proposed an effective deblurring algorithm with adaptive dictionary learning using a single image. By decomposing the blind deconvolution problem into three simple ones and learning a sparse dictionary from the image, our method is able to estimate the blur kernel and thereby deblur an image. Experimental results show that our algorithm achieves favorable performance against two state-of-the-art algorithms.

Chapter 4

Deblurring Low-light Images with Light Streaks

4.1 Introduction

Taking good pictures in low-light conditions such as at night or in a dark room is perhaps the most challenging task for non-professional photographers. Since a longer exposure time is often required in these cases to produce a well-lit image, the captured image using a handheld camera often ends up being blurry due to inevitable camera shake. It is thus highly desirable to apply image deblurring techniques to recover a sharp image from a blurry low-light image.

Although single image deblurring techniques have been advanced in recent years [18, 69, 49, 12, 40, 35, 7, 19, 66, 61], the state-of-the-art methods often have difficulties handling low-light images. This is because the success of recent approaches hinges on salient image features such as edges [12, 32, 27, 66] for blur kernel estimation. However in low-light images, the amount of salient image features that can be extracted is often limited, as shown in the examples in Figure 4.1. Furthermore, low-light images often have undergone large in-camera non-linear tone mapping, which breaks the linear blur model that most

approaches assume [54].

In this chapter, we examine a special phenomenon that often appears in low-light images to help image deblurring: *light streaks*. Light streaks are caused by blurred light sources such as light bulbs, flash lights, reflected lights, etc., which are very common in both natural (e.g., stars in the sky) and man-made scenes (e.g., street lights). Given that these light sources are small, high-intensity objects in the scene, the light streaks they create roughly depict the shapes of the underlying blur kernels. Intuitively, light streaks contain rich blur information that could potentially help us achieve better deblurring results.

However, without proper treatment, the presence of light streaks can actually jeopardize the performance of existing deblurring approaches and prevent them from estimating a correct blur kernel. The reasons are twofold. Firstly, many blur kernel estimation methods extract and use salient edges for blur kernel estimation [12, 32, 63], but light streaks often contain sharp, strong edges that may be mis-treated as structural edges by these algorithms. Regarding blur kernel estimation, Harmeling et al. [23] and Whyte et al. [59] discard saturated pixels and estimate the blur kernel using remaining pixels. Secondly, light streaks often contain saturated pixels. As shown in [14], without proper handling of saturated pixels, deconvolution of light streak pixels may result in severe ringing artifacts, causing problems for both blur kernel estimation and producing a high quality final output. To suppress artifacts caused by saturated pixels in non-blind deconvolution, Cho et al. [14] and Whyte et al. [59] explicitly model saturated pixels in their optimization processes, which alternately detect saturated pixels and estimate the latent image using non-saturated ones.

In this work, we propose a new deblurring framework that properly uses light streaks as an additional cue for blur kernel estimation. We extend the widely-used linear blur model by explicitly modeling point light sources and light streaks, resulting in a non-linear model that more accurately describes the formation of low-light images that contain light streaks. We then formulate a kernel estimation energy function that takes into account light streaks



Figure 4.1: Deblurring a low-light image. The amount of salient image features that can be extracted from the low-light image is limited. This leads to the failure of the state-of-the-art methods [12, 66] on the image. The proposed method (d) improves the results by detecting and making use of the light streaks (the detected light streaks are shown in the boxes of (a)).

as well as other image structures. Our method also automatically detects “good” light streaks that are useful for kernel estimation. After the blur kernel is estimated, the final output image is obtained by a regularized Richardson-Lucy deconvolution with outlier handling to suppress ringing artifacts.

4.2 Related Work and Context

The problem of removing spatially invariant blur has been studied extensively and significant progress has been made towards this [18, 46, 38, 56, 40, 12, 63, 15, 16]. The spatially invariant blur is usually formulated as a convolution process,

$$B = K * I + N \quad (4.1)$$

where B , I and K are a blurry image, a latent image and a blur kernel respectively, N is noise and $*$ is a convolution operator. Removing blur caused by camera shake becomes a blind deconvolution problem. Since blind deconvolution is an ill-posed problem, prior knowledge is often required for effective solutions. Early work focuses on generic priors obtained from the image statistics of natural images. In [18], the heavy-tailed gradient distribution of natural images is exploited as a constraint for sharp images. A mixture of Gaussians is used to fit the distribution of the gradient magnitudes of natural images. A method that uses constraints on the gradients of the latent image is presented in [49], with additional constraints on the consistence of local smooth regions before and after blurring. In [6], a deblurring method is proposed to exploit sparsity constraints for both the blur kernel and the latent image in the wavelet domain. In contrast to methods that adopt fixed priors, recent methods explore adaptive priors to effectively fit the image content of specific inputs. In [15], an image restoration algorithm is proposed to apply adaptive priors based on texture contents. In [70], the adaptive sparse prior has been used in a multi-image deblurring framework, where the coupled penalty is proved to handle the local minimum problem well. The power-spectrum based algorithms [68, 19, 25] exploit the property

that the power spectra of sharp natural images can be effectively modeled by statistical regularities. Thus, the power spectrum of a blur kernel can be estimated from that of a blurry image.

In this work, we explore a commonly existing phenomenon, light streaks, as specific information for estimating blur kernels. There has been only limited work that considers light streaks for image deblurring. The most related work to ours is an interactive deblurring method proposed by Hua and Low [28], in which the user needs to manually select a light streak region, and the system applies image processing operations to it for extracting the kernel. Since no other image structures are used, the blur kernel generated from a small cropped region may not be optimal for the entire image, especially for non-uniform blur cases. In contrast, our method not only automatically detects useful light streaks, but also incorporates them into a more principled optimization process for estimating more accurate blur kernels.

The spatially invariant blur cases assume that the camera shake only contains translational motion, which does not hold in practice. The real blurry images undergo spatially variant blur in which patches are degraded differently in an image. This problem is attracting more attention nowadays due to its wide range of practical applications [2, 13, 50, 56, 31, 60, 21, 24, 57]. In [60, 57], geometric models have been proposed to model the observed blurry image as the integration of all the intermediate images captured by the camera along the motion trajectory. These intermediate images are modeled as the transformed copies (i.e., homographies) of the sharp image. Based on this model, image blur caused by camera motion can be well formulated as an optimization problem. It is also possible to remove spatially varying blur by estimating a general camera motion function. A similar model has been used to model three degrees of camera motion (with in-plane translation and rotation) in [21]. The optimization problems for removing spatially variant blur often require expensive computational load. To speed up the optimization step, fast patch-based non-uniform deblurring methods have been developed [22, 24]. In [26], the back projection technique has been used to initialize the camera motion from estimated

kernels of several image patches for fast convergence. In this work, we also discuss the efficient non-uniform blur extension of the proposed method with light streaks.

4.3 Light Streak Detection

We now describe how to detect image patches which contain “good” light streaks for kernel estimation, based on pre-defined properties. It is a three-step approach: we first detect a set of candidate image patches that potentially contain light streaks; we then select the light streak that is most similar to the underlying blur kernel using a power-spectrum-based metric; finally, we use the selected best light streak to find additional good light streak patches for kernel estimation.

The motivation for extracting multiple light streaks is that a single light streak may be fully or partially saturated, thus it may contain only limited information about the blur kernel. By using multiple light streaks, we can cumulatively extract more blur kernel information. Furthermore, using multiple light streaks from different parts of the image helps the system achieve better stability against image noise and local distortions.

4.3.1 Candidate Patch Detection

Our method first identifies a set of image patches that may potentially contain light streaks. Given the physical nature of light streaks, we define the following properties as the appearance priors for good light streak patches: (1) pixels covered by a light streak should have relatively high intensities and those in the background have relatively low intensities in a local neighborhood; (2) the high intensity pixels in a light streak patch should have a very sparse distribution; (3) the light streak should be located near the center of a patch; and (4) there should be no other image structures in a light streak patch.

Since the goal of this step is merely removing irrelevant image patches from being considered in the following steps for reducing computational cost, we apply a set of heuristic



Figure 4.2: Examples of light streak detection. The red box indicates the best light streak patch and the green boxes show additional light streak patches that are automatically identified.

filters to quickly achieve it. Firstly, we apply two thresholds on maximum image intensity and gradient magnitude to filter out dark and flat patches. The thresholds are set adaptively based on global statistics, i.e., the top 10% pixels are above the thresholds. According to Property 2, we discard those patches that contain too many high intensity pixels (more than 15%). According to Properties 3 and 4, we divide each patch into two regions: the center region whose size is half of the original patch, and the border region. We then compute how many pixels have either high intensity or high gradient magnitude (above the thresholds) in the border region, and normalize it by the number computed from the center region. If the ratio is higher than 30%, we discard the patch. In this way we can quickly rule out most irrelevant patches and generate only a reasonable number of candidate patches for more careful light streak detection.

4.3.2 Finding the Best Light Streak Patch

From the set of candidate light streak patches, we carefully find the one that best resembles the underlying blur kernel. Intuitively, the best light streak patch should contain a well-lit light trajectory that has roughly the same shape as the blur kernel, on a relatively

clean background. This requires the underlying light source to be in-focus and to have a small size, and to be well separated from other image structures.

Unfortunately the blur kernel is unknown so we cannot directly use it for selecting the best light streak. Recent work [19] shows that we can get a good approximation of the power spectrum of the blur kernel directly from the input image. We thus define a power-spectrum-based metric for selecting the best light streak.

Specifically, the power-law of natural images shows:

$$|\hat{I}(\omega)|^2 \propto \|\omega\|^{-\beta}, \quad (4.2)$$

where $\hat{I}$ is the Fourier transform of an image I , ω is the coordinate in the frequency domain and $\beta \approx 2$. It is well known that a Laplacian filter is a good approximation to $\|\omega\|^{-\beta}$, so that:

$$|\hat{I}(\omega)|^2 |\hat{L}(\omega)| \approx C, \quad (4.3)$$

where L is a Laplacian filter, and C is a constant. For a blurred image $B = K * I + N$, we have:

$$|\hat{B}(\omega)|^2 |\hat{L}(\omega)| \approx |\hat{I}(\omega)|^2 |\hat{K}(\omega)|^2 |\hat{L}(\omega)| \approx C |\hat{K}(\omega)|^2. \quad (4.4)$$

In the spatial domain, we have $B \otimes B * L \approx C(K \otimes K)$, where $\otimes$ is a correlation operator. We thus define our metric as:

$$d(P, B) = \min_C \|B \otimes B * L - C(P \otimes P)\|^2, \quad (4.5)$$

where P is a candidate light streak patch. The optimal C can be found by solving a least square problem. Among all the candidate patches, we select the one with the smallest distance as the best light streak patch, $P_0 = \arg \min_P d(P, B)$. Note that this method also naturally favors unsaturated light streaks, as saturated ones would result in larger metric distances. In Figure 4.2, we show some examples of the best light streak patch selected using this method.

4.3.3 Finding Additional Light Streak Patches

Next, we use the selected best light streak patch to find additional light streak patches from the initial candidate set. This is done by computing the Euclidean distance between the candidate patch P and the best patch P_0 . A histogram equalization operation has been applied to all patches before computing the distances to account for the intensity difference between different light streaks. By setting a threshold, we define a set of detected light streak patches $\mathbb{P} = \{P_i\}_{i=1}^{N_P}$. We use a threshold $0.13S_P \max(P_0)$ where S_P is the patch size in our implementation. Figure 4.2 shows some examples of detected light streak patches.

4.4 Blur Kernel Estimation using Light Streaks

We now describe how to estimate the blur kernel using the detected light streaks as well as other image structures. The traditional linear motion blur model is formulated as $B = K * I + N$. This simple formulation however is insufficient to model the behavior of light streaks. To separately handle image structures and light streak patches, we divide the pixels in the observed image B into three complementary sets B^p , B^r and B^s , which are defined as $B^p = \bigcup P_i$, $B^r = \{x | B(x) \text{ is not saturated} \wedge x \notin P_i \forall i\}$ and $B^s = \{x | B(x) \text{ is saturated} \wedge x \notin P_i \forall i\}$. We assign each B^* a binary mask M^* so that $B^* = M^* \cdot B$. Here $\cdot$ denotes per-pixel multiplication. The three sets can be easily determined after the light streak detection step, before kernel estimation, and they remain fixed in later steps.

We then introduce a more accurate, nonlinear blur model for the input image as:

$$\begin{cases} B^p &= \sum P_i \\ B^r &= M^r \cdot (K * I) + N \\ B^s &= M^s \cdot c(K * I + N) \end{cases}, \quad (4.6)$$

where c is a clipping function defined as $c(v) = v$ if v is in the dynamic range of the camera sensor, and $c(v) = 0$ or 1 otherwise¹.

We use $\hat{P}_i$ to denote the unclipped light streaks such that $P_i = c(\hat{P}_i)$. We also introduce auxiliary variables D_i , for each detected light streak patch P_i , which describe the appearance of the original point light sources that produced the light streaks. We further assume each point light source has a disk shape, but may have a different size and at a different intensity value. In our method, these variables are estimated as well. Specifically, each light streak is modeled as:

$$\hat{P}_i = K * D_i + N. \quad (4.7)$$

Thus, the first line in (4.6) becomes

$$B^p = \sum c(\hat{P}_i) = \sum c(K * D_i + N). \quad (4.8)$$

Given the above model, we seek K , D_i and I that can best describe the observed image and detected light streaks. This is done by adopting the widely-used alternating optimization approach: given the initial values of the three variables, we fix two of them at each time and optimize the one that is left.

4.4.1 Updating K

In this step, we fix D_i and I , and update K by optimizing the following energy function:

$$\begin{aligned} f_K(K) = & \sum_{x \in M^r} |\partial_h B(x) - (K * P_h)(x)|^2 \\ & + \sum_{x \in M^r} |\partial_v B(x) - (K * P_v)(x)|^2 + \lambda \|K\|_1 \\ & + \mu \sum_{P_i \in \mathbb{P}} \sum_{x \in P_i} |(D_i * K)(x) - \hat{P}_i(x)|^2, \end{aligned} \quad (4.9)$$

¹In this chapter, we use the dynamic range normalized into $[0,1]$.

where x is the pixel index. The first two terms on the right side are data terms based on the prediction scheme proposed by Cho and Lee [12]. ∂_h and ∂_v are partial differential operators along the horizontal and vertical axes, respectively, and P_h and P_v are predicted gradient maps along the two axes, respectively. Please refer to [12] for how to compute P_h and P_v using image filtering techniques. Note that the first two terms are applied only on pixels in M^r , so that they are affected by neither saturated pixels nor light streaks. This is in sharp contrast to previous edge-based approaches [12, 66] where edge extraction is inevitably affected by light streaks and saturated regions. The third term is a prior on K . The last term is derived from (4.7).

To obtain the underlying light streak $\hat{P}_i$, we replace the saturated pixels in P_i with the interpolated pixels using spline interpolation (explained in more detail in Section 4.4.4). The computation of $\hat{P}_i$ is pre-defined before the optimization process. Then the energy in (4.9) is minimized using an iterative reweighted least square method (IRLS). Here we set μ as $S_I/(S_P N_P)$ at the beginning, where S_I is the image size. We reduce μ with a factor 0.75 over iterations to rely more on the data error term. We set λ as $\sigma^2 S_P^2/50^2$ with σ denoting the noise deviation of the Gaussian prior.

4.4.2 Updating D_i

In this step, for each selected light streak patch, we estimate its original unblurred light source. As mentioned earlier, we assume that the original point light D_i has a disk shape, and its size and intensity can vary. Thus, we model D_i as a function of two parameters t_i and r_i , which denote the intensity value and the radius of the disk, respectively. Note that t_i is not restricted to the dynamic range of the image. We then derive an energy function for this step as:

$$\begin{aligned} f_{D_i}(t_i, r_i) = & \|D_i(t_i, r_i) * K - \hat{P}_i\|^2 \\ & + \|D_i(t_i, r_i) - I_i\|^2, \end{aligned} \quad (4.10)$$

where I_i is the patch in the latent image I covering the same pixels as P_i . Since we have strong prior knowledge about the light sources, e.g., they are usually very small and have high intensities, we thus sample a discrete set of possible t_i and r_i values, and find out the optimal one that minimizes $f_{D_i}(t_i, r_i)$. In practice, we found that this exhaustive search works well in both synthetic and real examples.

4.4.3 Updating I

In this step, we update the latent image I using the updated blur kernel K and light sources D_i by optimizing the following energy function:

$$\begin{aligned} f_I(I) = & \sum_i \mu \|D_i - I_i\|^2 + \sum_x |B(x) - c(K * I)(x)|^2 \\ & + \gamma \sum_x (|\partial_h I(x)|^\alpha + |\partial_v I(x)|^\alpha), \end{aligned} \quad (4.11)$$

where the third term is the sparse prior term proposed in [39] and $\alpha = 0.8$. γ is a weight for the regularization terms and we use $\gamma = 0.005$ in our experiments. We solve (4.11) using an IRLS method.

4.4.4 Initialization and Implementation Details

To compute $\hat{P}_i$ in (4.9), we calculate the 1D interpolation along horizontal and vertical axes using the Matlab function **interp1** and compute the average to replace the saturated pixels. For updating D_i , we sample a discrete set $\{100/255, 120/255, 140/255, \dots, 1000/255\}$ for t_i and $\{1, 2, 3\}$ for r_i . For the update of I , we refer the readers to [14] for parameter settings.

Our system performs kernel estimation in the original resolution without the coarse-to-fine strategy. In the first iteration, we compute K by taking out the first and second terms in (4.9) and only considering the best detected light streak patch as:

$$\arg \min_K \lambda \|K\|_1 + \mu \sum_{x \in P_1} |(D_1 * K)(x) - \hat{P}_1(x)|^2, \quad (4.12)$$

where we initialize D_i as the point light source with $r_i = 1$ and $t_i = \max \hat{P}_i$. Given the initial K and D_i , we then can compute the initial I , and iteratively update all three.

4.5 Deconvolution with Ringing Suppression

After the blur kernel is estimated, we use a non-blind deconvolution method to restore the latent image. As images taken in low-light conditions often contain lots of saturated pixels, they should be handled properly to avoid generating severe ringing artifacts.

There are some recent approaches for handling saturated pixels in non-blind deconvolution. Cho et al. [14] propose a blur model that explicitly models outliers including saturated pixels, and use an expectation-maximization (EM) method for generating the final image. Whyte et al. [59] propose a modified Richardson-Lucy (RL) method based on a blur model with a saturation function. While both methods use similar blur models, each has its own pros and cons. For example, Cho et al.’s method can handle other types of outliers, while Whyte et al.’s method provides an additional ringing prevention scheme. In our work, we derive a new deconvolution algorithm that combines the advantages of these two methods.

Our approach adopts the basic framework of RL deconvolution, because in practice we found that RL deconvolution is often more effective at suppressing ringing artifacts. In RL-deconvolution, the latent image I is estimated by maximizing a likelihood $p(B|K, I)$, which is defined using Poisson distributions. The update equation of RL deconvolution is derived by differentiating the log-likelihood with respect to I :

$$I^{t+1} = I^t \odot K \otimes \frac{B}{I^t * K}, \quad (4.13)$$

where $\odot$ and $\otimes$ mean per-pixel multiplication and correlation operations, respectively. Division is also pixel-wise.

To better handle outliers and saturated pixels, following Cho et al.’s work [14], we

formulate non-blind deconvolution as a MAP problem:

$$p(I|B, K) \propto \sum_{M \in \mathbb{M}} p(B|M, K, I)p(M|K, I)p(I), \quad (4.14)$$

where M is a mask for specifying inliers and outliers, i.e., $M(x) = 1$ if $B(x)$ is an inlier, and $M(x) = 0$ if $B(x)$ is an outlier. $\mathbb{M}$ is a set of all possible M . We assume Poisson distributions for inliers and uniform distributions for outliers. Then, the likelihood term $p(B|M, K, I)$ is defined as $P(B(x)|M, K, I) = \mathcal{P}(B(x)|K * I(x))$ if $M(x) = 1$ and $P(B(x)|M, K, I) = C$ otherwise, where $\mathcal{P}$ is a Poisson distribution, and C is a constant defined as the inverse of the width of the dynamic range. We define $P(M|K, I)$ and $P(I)$ in the same way to Cho et al.'s approach, as detailed in [14].

Given above formulations, we derive an EM-based regularized RL deconvolution method. The E-step computes per-pixel weights W^t at t -th iteration as:

$$W^t = \frac{\mathcal{P}(B|K * I^t)P_{\text{in}}}{\mathcal{P}(B|K * I^t)P_{\text{in}} + C(1 - P_{\text{in}})}, \quad (4.15)$$

where $P_{\text{in}} \in [0, 1]$ is the probability that B_x is an inlier. The M-step updates the latent image I as:

$$I^{t+1} = \frac{I^t}{1 + \lambda \rho(I^t)} \odot K \otimes \left(\frac{B \odot W^t}{I^t * K} + 1 - W^t \right), \quad (4.16)$$

where $\rho(I)$ is the derivative of a sparse prior:

$$\rho(I) = \text{sign}(\partial_h I) \alpha |\partial_h I|^\alpha + \text{sign}(\partial_v I) \alpha |\partial_v I|^\alpha. \quad (4.17)$$

We set $\alpha = 0.8$. Then deconvolution is done by alternatingly solving (4.15) and (4.16).

In our implementation, for better computational efficiency we approximate (4.15) using Gaussian distributions. Then, (4.15) becomes:

$$W^t = \frac{\mathcal{N}(B|K * I^t, K * I^t)P_{\text{in}}}{\mathcal{N}(B|K * I^t, K * I^t)P_{\text{in}} + C(1 - P_{\text{in}})}. \quad (4.18)$$

We typically run 40 iterations to obtain the deconvolution result. To further suppress ringing artifacts, we also adopt the ringing prevention scheme in [59], which decomposes an image into unsaturated and saturated regions and performs deconvolution separately. Figure 4.3 shows an example of our non-blind deconvolution.

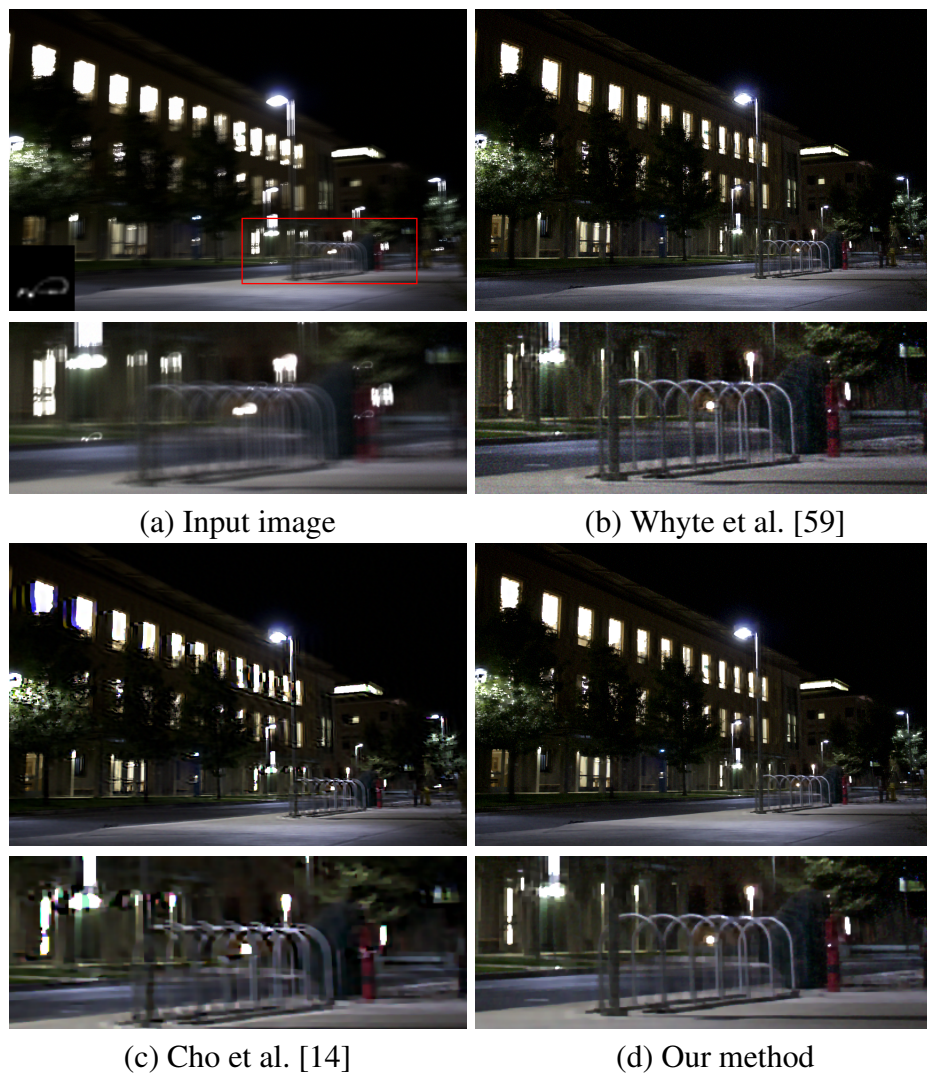


Figure 4.3: Non-blind deconvolution of a synthetic example. The second row displays the zoom-in view of the region in red box. The proposed method extends the Richardson-Lucy framework to suppress ringing artifacts, and handles outliers via pixel-wise weights.

4.6 Experimental Results

We implemented our method in MATLAB and conducted experiments on a PC with an 1.73 GHz Core i7 CPU and 8 GB RAM. For an image of size 700×1000 , the light streak detection step takes about 2 seconds, and the kernel estimation step takes around 5 minutes with our unoptimized implementation.

To evaluate the effectiveness of the proposed light streak detection method on real-world images, we collected 40 natural low-light images that contain light streaks, and applied our method to them. We then visually examined the extracted light streak patches in each image to see if the selected best light streak patches really contain light streaks. We found that in 35 out of 40 images (87.5%), our method successfully extracted correct light streaks. Figure 4.2 shows examples of our light streak detection.

4.6.1 Comparisons with Hua and Low’s Method

We first compare our method with Hua and Low’s approach [28], which estimates the blur kernel from a manually-selected light streak patch. Since this method only uses a single light streak patch, for a fair comparison we also limit our system to select the same patch. In this experiment, to test the robustness of the two approaches, for each input image we tried two different light streaks: the best patch suggested by our method, and a manually selected patch that is visually obvious to the user, resulting in two deblurring results for each method. Figure 4.4 shows the results on one example.

The results suggest that Hua and Low’s method are sensitive to the input patch: it works reasonably well with the automatically-detected, non-saturated light streak, but fails badly with the other one. This is because it only relies on the light-streak patch for extracting the blur kernel, thus it is sensitive to saturation which causes information loss. On the contrary, our method is more robust and works well even with the saturated patch, as we also rely on other image structures for kernel estimation. Furthermore, with the non-saturated patch, our method generates higher quality results than Hua and Low’s method

	[12]	[35]	[41]	[66]	Ours
KS	0.5323	0.5449	0.5298	0.5312	0.7069

Table 4.1: Quantitative comparisons using kernel similarity (KS).

due to the proposed optimization scheme. The results also demonstrate that our automatic selection method selects good light-streak patches for deblurring. More comparisons with Hua and Low’s method on real examples are shown in Figure 4.5 and Figure 4.6.

4.6.2 Comparisons on Real Images

Here we qualitatively compare our method with the state-of-the-art general image deblurring methods [12, 66] that performed well on the recent benchmark test [33], on some real-world examples. As shown in Figure 4.5 and Figure 4.6, previous methods performed poorly on these low-light images since they do not contain sufficient salient edges for kernel estimation, and their estimated kernels tend to be close to a delta function. In contrast, our method obtained more accurate kernel estimation and produced higher quality outputs.

4.6.3 Comparisons on Synthetic Images

We also prepared a synthetic dataset to quantitatively compare our method against others. To build the dataset, we first captured 11 low-light images in RAW format covering a variety of scenes, using a Canon Rebel Xsi camera with an EF-S 18-55 mm lens. For each image we then applied 14 different blur kernels, followed by adding Gaussian noise with 1% variation. This gave us 154 test images in total.

Figure 4.8 and Figure 4.9 show some representative examples of this dataset, as well as the deblurred results using different algorithms. For a fair comparison, we use the non-blind deconvolution method described in Section 4.5 for all methods. The results suggest that our results have significantly higher visual quality than those generated by other methods, due to more accurate kernel estimation.

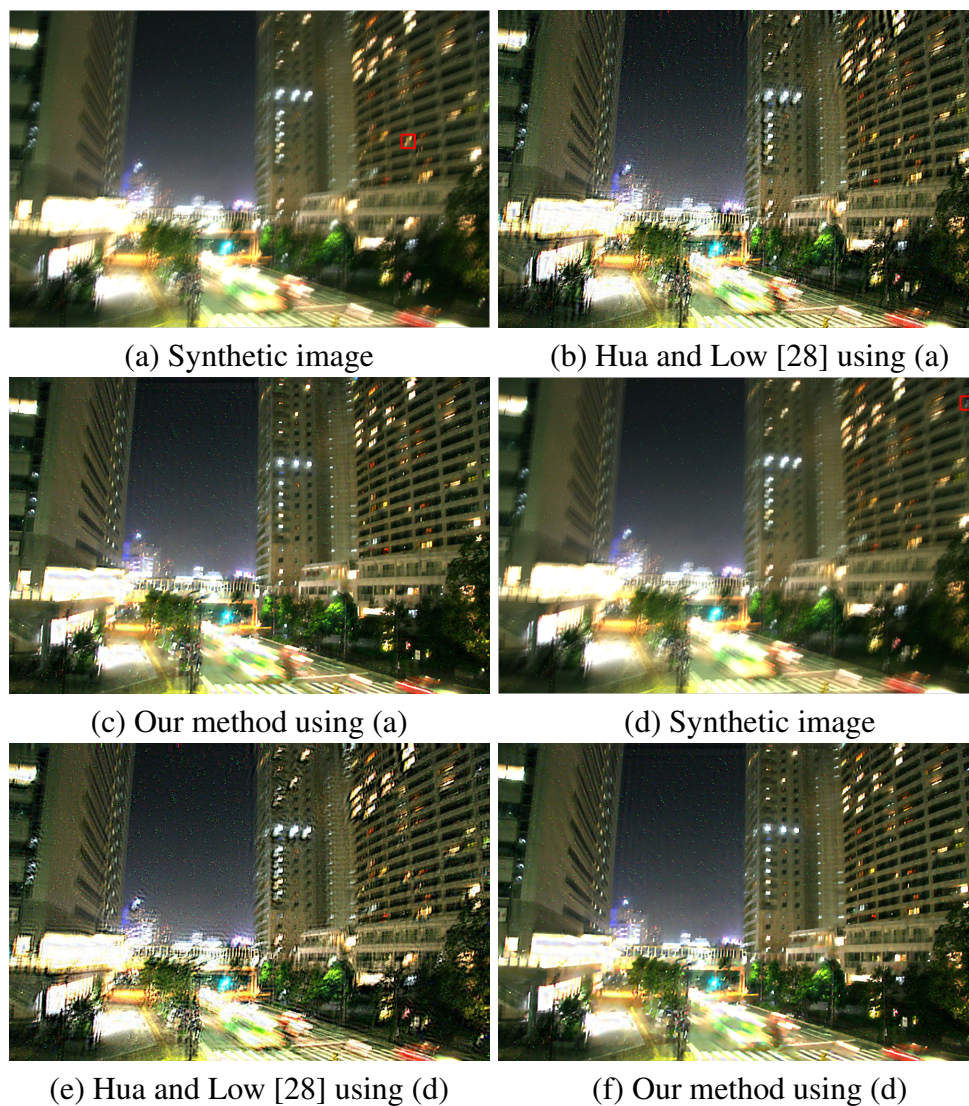


Figure 4.4: A comparison with Hua and Low's approach [28]. The red box indicates our selected light streak patch and the green box is a manually selected patch.



Figure 4.5: Comparisons with state-of-the-art methods on real examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows.



Figure 4.6: Comparisons with state-of-the-art methods on real examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows.

We compare our method with the state-of-art uniform deblurring methods [12, 63, 35, 41]. For quantitative evaluation, we use the metric proposed in [40] to describe how well the estimated kernel reconstructs the image compared to the ground truth kernel. This metric computes the difference between a recovered image I_r and the known ground-truth sharp image I_g , over the difference between the deblurred image I_{k_g} with the ground truth kernel k_g and the ground-truth sharp image as $\|I_r - I_g\|^2 / \|I_{k_g} - I_g\|^2$. The success rate of reconstructed error ratio is then used to evaluate the effectively of the algorithm, and the results are shown in Figure 4.7. We also compute the average kernel similarity [27], which describes the similarity between the estimated kernel and the ground-truth kernel, and the results are shown in Table 4.1. These results demonstrate that the proposed method outperforms previous general image deblurring approaches on deblurring low-light images, due to the explicit light streak modeling.

4.6.4 Failure Cases

The proposed method also fails in some cases. One scenario is when the underlying light sources of the light streaks are large and no longer point lights, as shown in Figure 4.10(a). This type of light streak may be selected by our algorithm and could lead to an erroneous kernel estimation. This problem can be partially alleviated with user guidance by manually selecting a better patch, if it is available in the image.

Another difficulty concerns non-blind deconvolution on a large saturated region as shown in Figure 4.10(b). For this synthetic example, even if we supply the ground truth blur kernel, the proposed method and other non-blind deconvolution methods can not generate satisfactory results due to severe information loss, as shown in Figure 4.10(c).

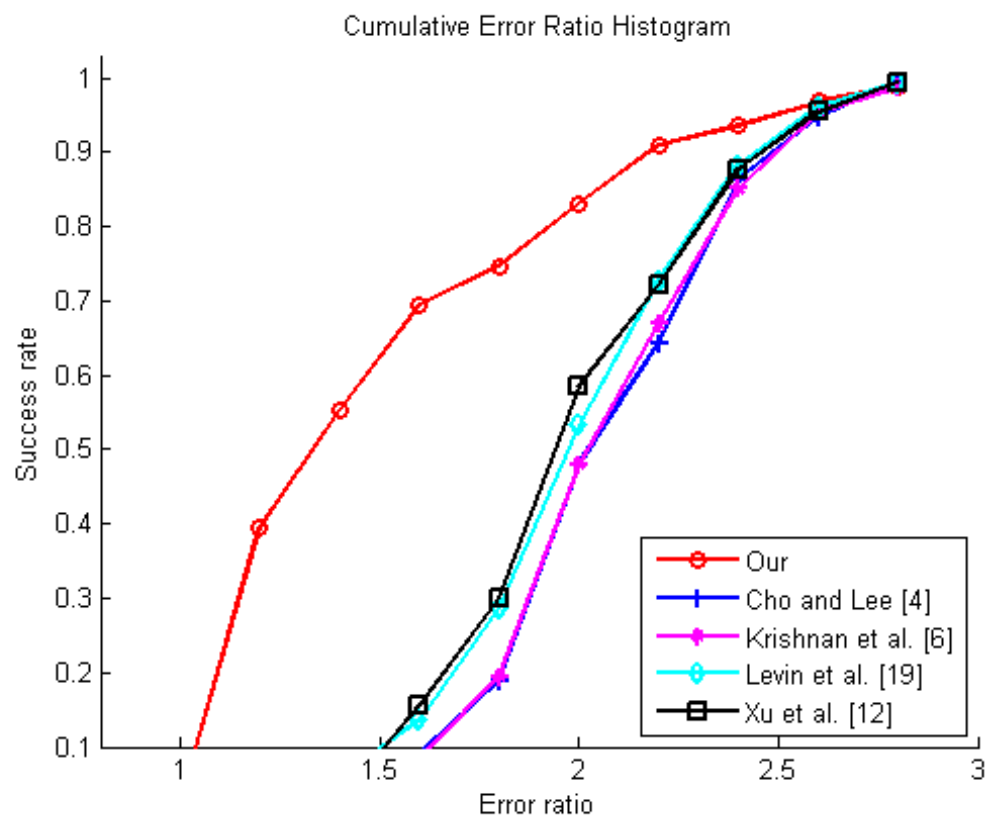


Figure 4.7: Cumulative error ratio histogram [40] on the synthetic dataset of uniform blur. There are 154 blurry images consisting of 11 low-light images and 14 blur kernels.

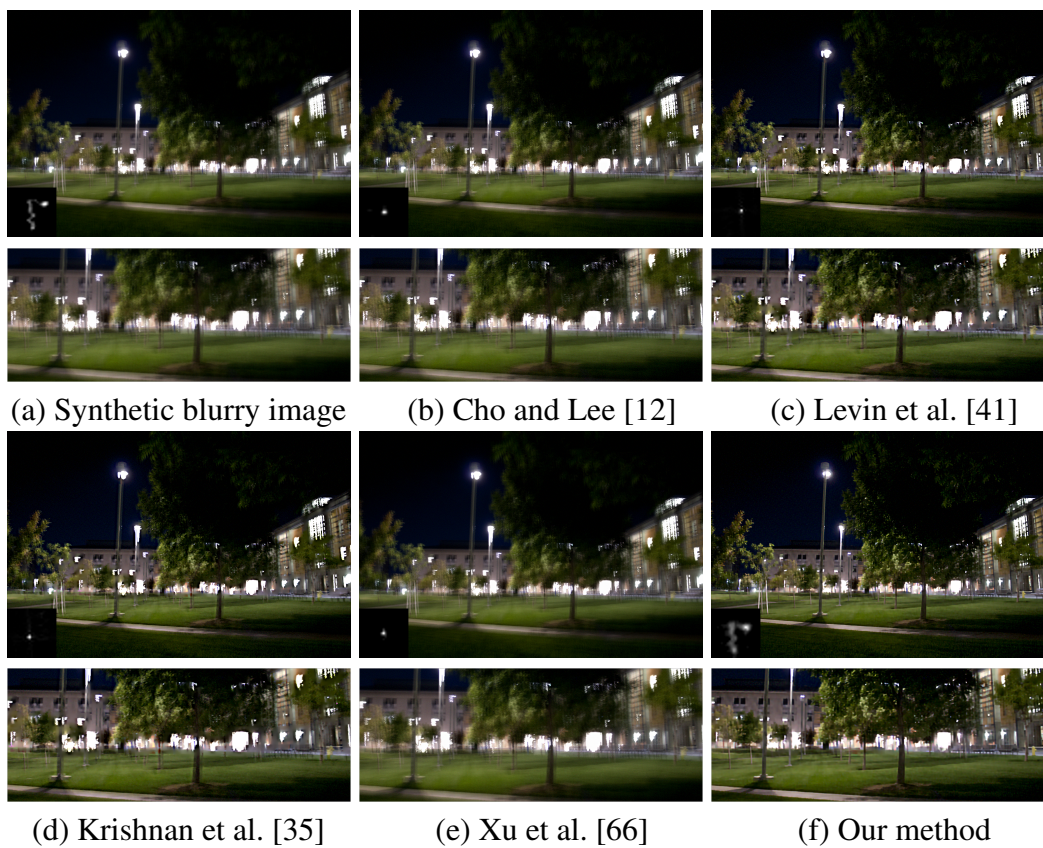


Figure 4.8: Comparisons on synthetic examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows.

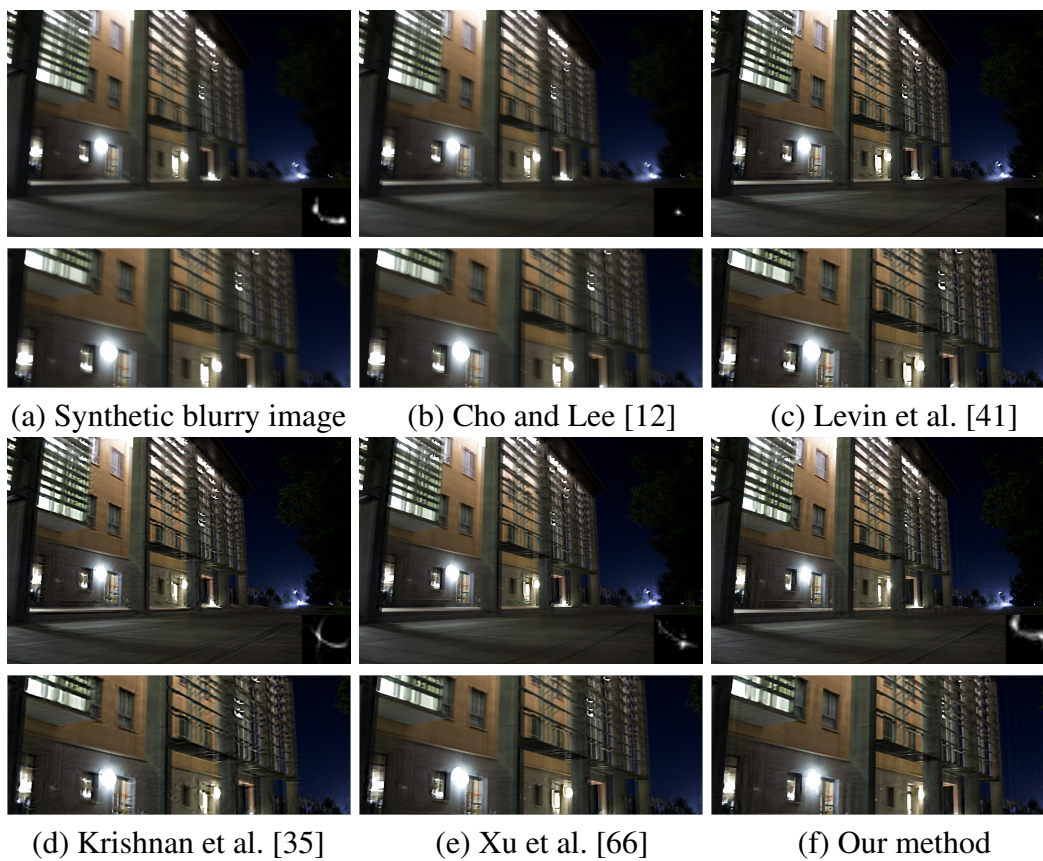


Figure 4.9: Comparisons on synthetic examples. The images in the second and fourth rows are zoom-in views of that in the first and third rows.

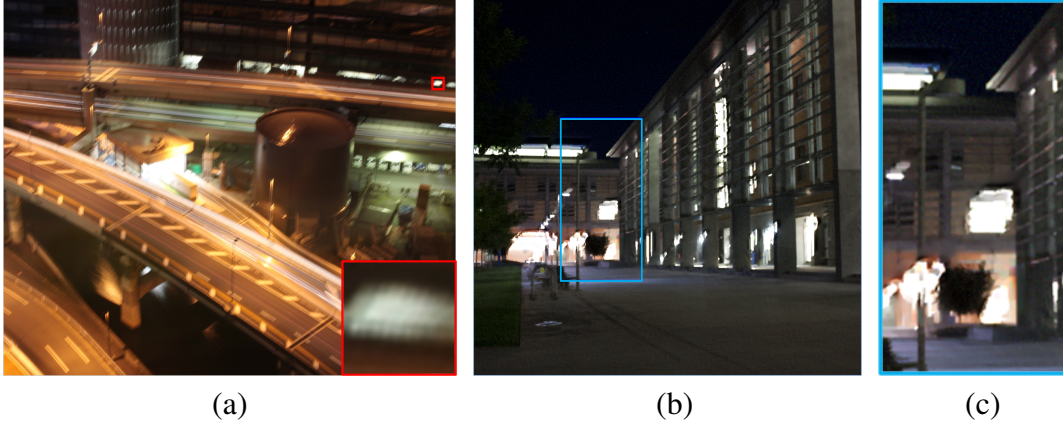


Figure 4.10: Examples of failure cases. (a) a blurry image with failed light streak detection; (b) a deblurred image using the ground truth kernel; (c) a cropped region of (b).

4.7 Non-Uniform Deblurring

In this section, we show that the proposed region selection method can also be used in non-uniform deblurring problems. In practice, the camera shake results in spatially variant blur, in which the blur shape at different locations is different. For efficiency, we approximate the spatially variant blurry image as the sum of several blurry patches following the fast forward model [24] as

$$B = \tau \diamond L = \sum_r K^r * (\tau^r \odot I) + n, \quad (4.19)$$

where $\tau \diamond L$ denotes the fast forward model, K^r represents the blur kernel for the r -th region, and τ^r is the weighting parameter indicating r -th region. The kernel K^r is constrained within the geometric model as $K^r = \sum_{\theta} \tau_{\theta} b_{\theta}^r$, with b_{θ}^r and τ_{θ} denoting the kernel basis for the r -th region and its corresponding weight. Here each kernel basis b_{θ} is calculated by applying the corresponding homography to a grid of single-pixel dots, each one located at the center of a region. The weight τ_{θ} indicates the relative exposure time

the camera stays at the pose θ . Then the model can be written as

$$B = \sum_r \left(\sum_{\theta} \tau_{\theta} b_{\theta}^r \right) * (w^r \odot I) + n. \quad (4.20)$$

Similar to (4.6), we replace the uniform mode with the fast forward model as

$$\begin{cases} B^p &= \sum P_i = \tau \diamond D \\ B^r &= M^r \cdot (\tau \diamond I) + N \\ B^s &= M^s \cdot c(\tau \diamond I + N) \end{cases}, \quad (4.21)$$

where $D = \sum D_i$ is the dot-pattern image.

4.7.1 Estimation of the Blur Parameter τ

To estimate τ , we optimize the following energy function similar to (4.9),

$$\begin{aligned} f_{\tau}(\tau) &= |\partial_h B - \partial_h(\tau \diamond I)|^2 + |\partial_v B - \partial_v(\tau \diamond I)|^2 \\ &\quad + |B^p - (\tau \diamond D)|^2 + \lambda |\tau|^2, \end{aligned} \quad (4.22)$$

where ∂_h and ∂_v are partial differential operators along the horizontal and vertical axes, respectively. The function utilizes the detected light streaks as the constraints to regularize the camera motion weight estimation. Since the model is linear with respect to τ , there exists the matrix-vector-multiplication expression $A_h \tau = \partial_h(\tau \diamond I)$ and $A_v \tau = \partial_v(\tau \diamond I)$ with respect to fixed I , and $A_D \tau = \tau \diamond D$ with respect to a fixed D . Thus, the energy function (4.22) can be written as

$$\begin{aligned} f_{\tau}(\tau) &= |\partial_h B - A_h \tau|^2 + |\partial_v B - A_v \tau|^2 \\ &\quad + \mu |B^p - A_D \tau|^2 + \lambda |\tau|^2. \end{aligned} \quad (4.23)$$

We minimize the equation using gradient descent and the gradient of f_τ is derived

$$\begin{aligned} \frac{\partial f_\tau(\tau)}{\partial \tau} = & 2(A_h^\top A_h + A_v^\top A_v + \mu A_D^\top A_D + \lambda)\tau \\ & -2(\partial_h B + \partial_v B + \mu B^p). \end{aligned} \quad (4.24)$$

4.7.2 Estimation of the Latent Image I

The model (4.20) can also be written in matrix form with respect to τ . Since it is not guaranteed that there exist light streaks in each region, we do not impose light streak constraints in the latent image estimation. Thus, the process becomes normal latent image estimation with non-uniform blur [24, 66].

4.7.3 Light Streak Detection in Non-Uniform Case

In Section 4.3, we introduced the method to determine light streaks, by comparing the power spectrum of the whole image with the interest patches. This method works well for the uniform blur situation. However, there are problems when directly applying this to non-uniform blur cases. The non-uniform blur causes the blur kernels to appear differently in different locations, thus the power-spectrum of each varies among locations. To deal with the issue, we split the image into a grid of regions same as the regions mentioned in Section 4.7. Each region is assumed to be affected by a uniform blur kernel and we can apply the power spectrum method to detect light streaks in each region. However, this process leads to a high false positive rate of detection as some of the regions do not have much image structure as shown in Figure 4.11(a). To make the process more robust, we utilize cross-validation as we assume similarity among detected light streaks although they appear spatially variant. We make use of the kernel similarity [27] to measure the similarity between the detected light streaks, and set a threshold of 0.6 to determine whether two light streaks have strong correlation. If a detected light streak is found to have less than 2 strong

correlations, it is considered false alarm. With the validation step, we achieve a light streak detection with a low false positive rate, as shown in Figure 4.11(b).

4.7.4 Comparison Results

We first compare our method with the state-of-art methods including uniform [12, 35, 66] and non-uniform [24, 59] methods on real images. As shown in Figure 4.14, those methods fail to deal with low-light images. Although these methods prove to be effective for generic deblurring, they are more likely to converge to trivial solutions (i.e., delta kernel) for low-light images as there are not many salient edges in the images. The proposed method detects the light streaks accurately and estimates the non-uniform blur well with the constraints of the light streaks.

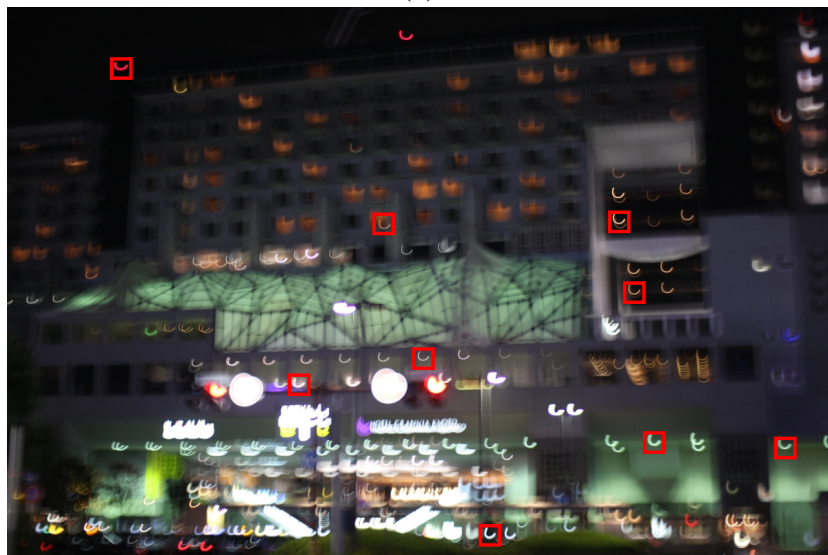
To provide comprehensive comparison, we compare our method with the state-of-the-art methods on a synthetic dataset based on [33]. Since the images provided in [33] are in good illumination, we generate 165 low-light images consisting of 11 clear images captured at night and 15 camera motions from [33], followed by adding Gaussian noise with 1% variation. We compare our method with the state-of-art uniform deblurring methods [12, 35, 66] and non-uniform deblurring methods [59, 24] on the dataset. We use the metric of cumulative error ratio [40] to evaluate the efficiency of the methods and the results are shown in Figure 4.13. One example from the dataset is given in Figure 4.12.

4.8 Conclusion

In this chapter we propose a new deblurring method that explicitly models light streaks for low-light image deblurring. Our method detects the light streaks appearing in the blurry image and incorporates them into an optimization framework, which jointly considers light streaks and other image structures for kernel estimation. We propose a non-blind deconvolution step to carefully suppress the ringing artifacts caused by light streaks. We



(a)



(b)

Figure 4.11: Example of light streak detection in non-uniform blur. (a) a spatially variant blurry image with directly applying light streak detection to it; (b) a spatially variant blurry image applying light streak detection and cross-validation.

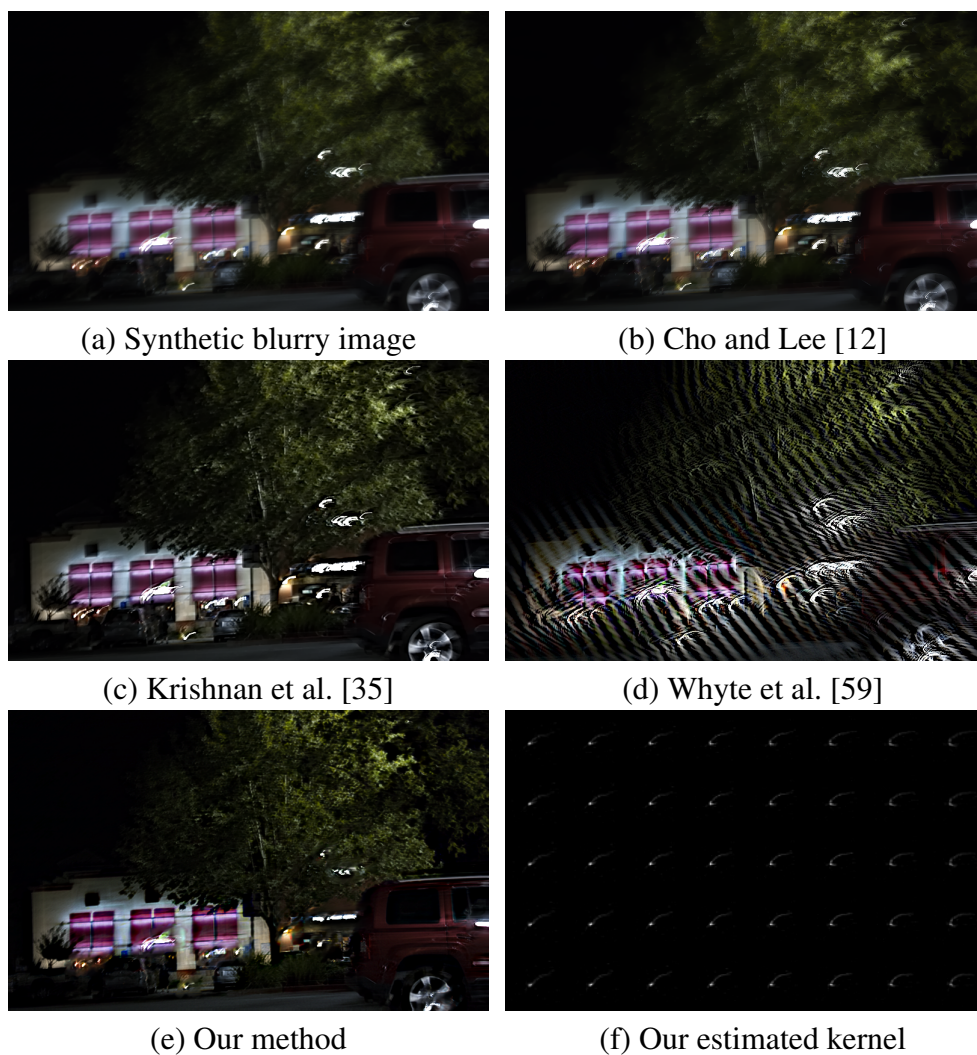


Figure 4.12: An example from the synthetic dataset of non-uniform blur. The image is generated by applying camera motion obtained from [33] to a low-light image, followed by adding Gaussian noise with 1% variation.

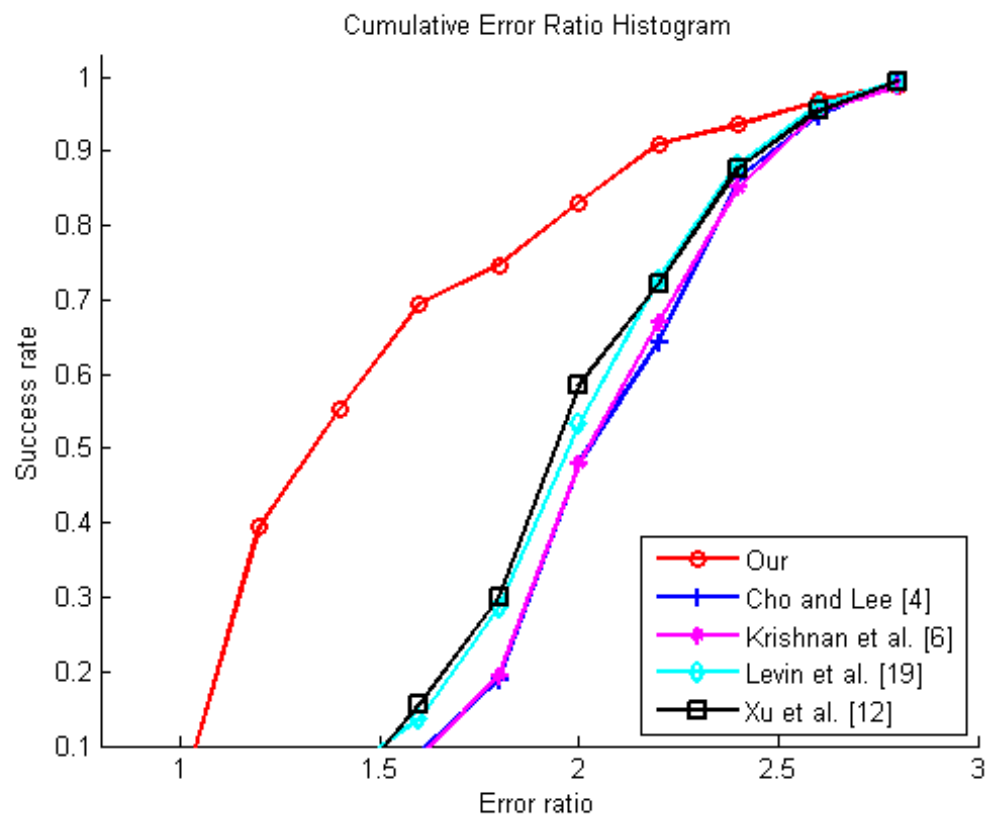


Figure 4.13: Cumulative error ratio histogram [40] on the synthetic dataset of non-uniform blur. There are 165 blurry images generated from 11 low-light images and 15 camera motions from [33].

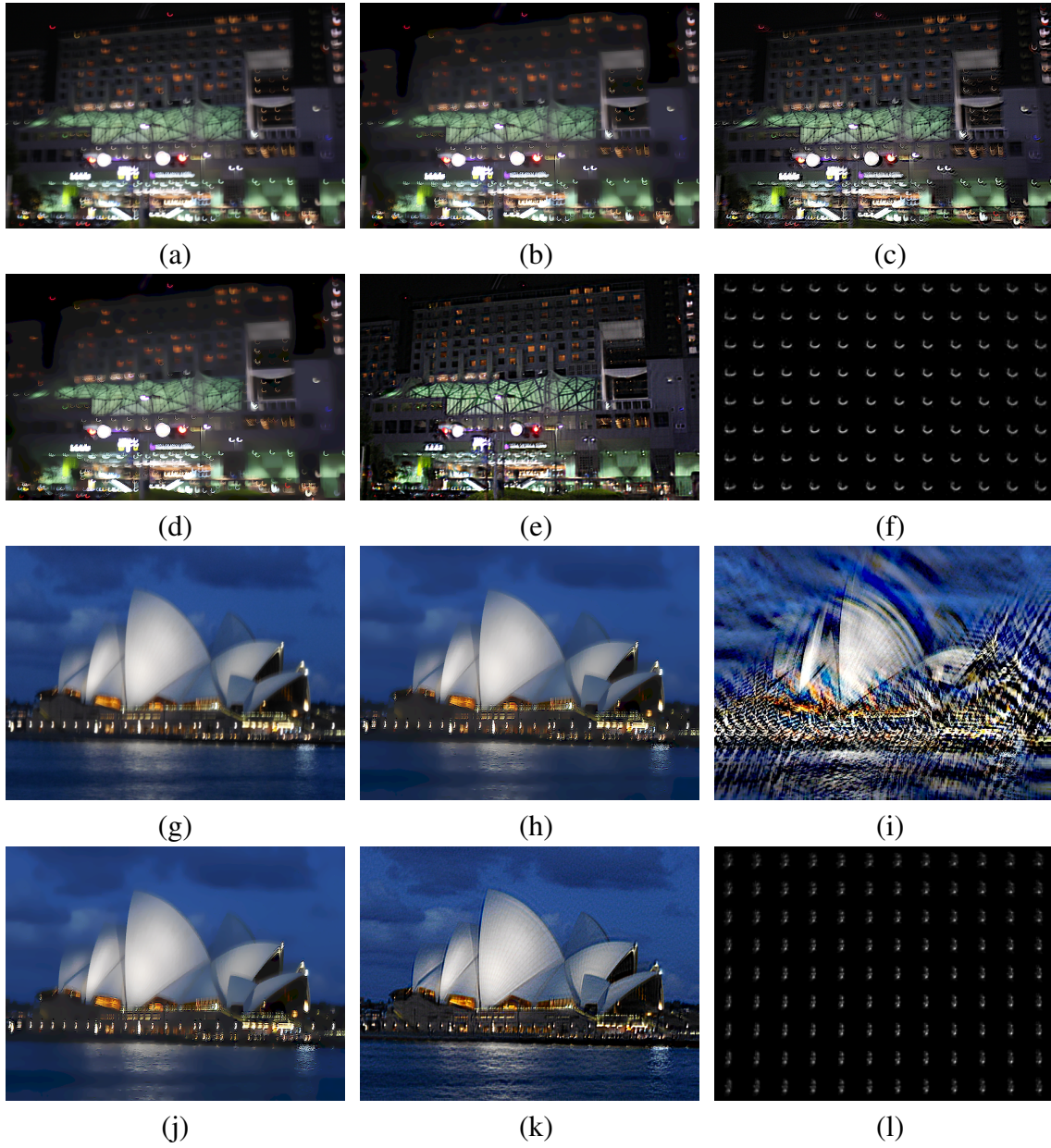


Figure 4.14: Examples of non-uniform deblurring. (a)(g) blurry images with our detected light streaks; (b)(h) deblurred results by [12]; (c)(i) deblurred results by [59]; (d)(j) deblurred results by [66]; (e)(k) deblurred results by our method; (f)(l) our estimated kernels.

also show that the proposed method can be extended to the spatially variant deblurring case for a more general framework. Experimental results show that our system can obtain decent results on the challenging images that previous methods can not handle.

Chapter 5

Good Regions to Deblur

5.1 Introduction

For the single-frame deblurring problem, it is usually advantageous to make full use of the blurry image. However, not all pixels of the blurry image are informative. Smooth regions, for example, do not contribute much for estimating the blur kernel. This encourages us to ask “What kind of image features or structure of the blurry image really help in blur kernel estimation?” By detecting good image features for blur kernel estimation, an accurately estimated blur kernel can then be used to recover a latent clear image with high visual quality. In [18, 40], it is demonstrated that regions with strong edges tend to yield better deblurring results. Some of the gradient-based methods favor salient edges with gradients of specific patterns [32, 15, 63]. On the other hand, based on 1D signal examples, it is demonstrated that edges of short length could adversely affect the deblurring results [63]. In other words, if the whole image is used for image deblurring without deliberate selection of good features, negative impacts are likely to result in inferior results. For this reason, along with computational efficiency, it is preferable to determine a comparatively small region, rather than the whole image, for estimating the blur kernel. Figure 5.1 illustrates that different regions may lead to completely different kernel esti-

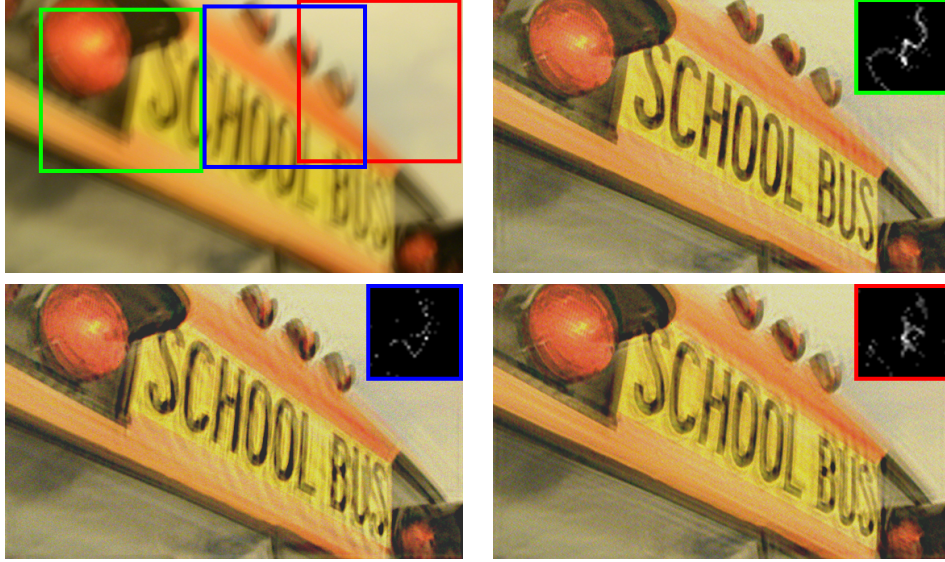


Figure 5.1: Different regions lead to different kernel estimations and deblurred results. The top left image is the input blurry image with three subwindows selected for estimating kernels. The other three images are the recovered images and estimated kernels from these three subwindows using [18].

mation results, and thereby different deblurred images. This problem can often be partly alleviated by manual selection and visual inspection of the results. However, this requires tedious human input and thus cannot yield fully automatic deblurring algorithms. In addition, questions regarding which regions or what image structures are crucial for accurate blur kernel estimation remain unanswered.

In this chapter, we address these questions for effective and efficient image deblurring. We first propose a metric that quantitatively measures the similarity between kernels, which facilitates the process of labeling good estimated kernels effectively. Instead of determining good image structures from empirical understanding and prior knowledge, we resort to learning for this task based on a collection of labeled data with the proposed kernel similarity measure. We pose the learning problem within a conditional random field

(CRF) [37] framework in order to exploit contextual constraints among image regions. In addition, we explore the contribution of different features with structured output. We construct a dataset which covers sufficient variability of image structure and blur kernels following the technique described in [40] for evaluation, and apply the learned models to select good image regions for deblurring. The proposed algorithms select good regions automatically for deblurring with efficient and favorable results, thereby relieving users from tedious trials for selecting image patches. Also the empirical results also provide some insights about the kinds of image structures or features that are useful for deblurring algorithms.

5.2 Kernel Similarity

Existing methods mostly resort to the visual quality of deblurred images for empirical evaluation. While it is important to recover high visual quality images, it is neither reliable nor effective to evaluate recovered results since human vision is sensitive to noise and ineffective in telling minute differences. As suggested in [40], it is preferable to separate the image deblurring problem into two steps: 1) blur kernel estimation and 2) non-blind deconvolution. If the blur kernel can be accurately estimated, then the deblurred images can be then easily recovered with non-blind deconvolution algorithms. Therefore, the ensuing question is how to identify the kernels effectively.

The difficulty of comparing kernels arises when kernels vary in terms of shift and range (i.e., the size of the kernel). Two kernels K_1 and K_2 are considered shift and range invariant if the dominant (i.e., non-zero) parts of them are the same, regardless of differences in locations and sizes of the kernel windows. Figure 5.2 shows three kernels that vary in shift and range. Thus, a good metric for kernel similarity should be shift and range invariant.

The commonly used root-mean-square-error (RMSE) metric is not effective in computing the similarity between two kernels. Typically, smooth kernels are favored by RMSE, due to its 2-norm form and the fact that the entries of blur kernel sum up to one. Figure 5.3

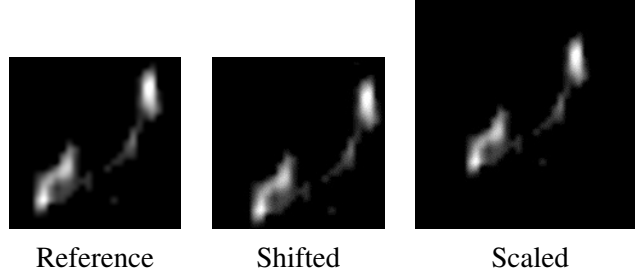


Figure 5.2: Estimated kernels vary from shift to scale, but the dominant parts are the same. These kernels should be considered as the same by a good kernel similarity metric.

shows several examples how RMSE performs with the ground truth blur kernel. Visually, the second estimated kernel (c) with a small extra part is closest to the ground truth kernel, either with respect to the shape or the image intensity. The first estimated kernel (b), though over-smoothed, looks more similar to the third one (d), which misses some parts and focuses more on the bottom section. In this case, the estimated kernels with low RMSEs do not necessarily indicate good results, while the estimated kernels with slightly higher RMSEs in fact match better with the ground truth kernels.

To deal with the above-mentioned problems, we propose a kernel similarity metric to effectively compare estimated kernels with the ground truth. We utilize the maximum response of normalized cross-correlation to represent the blur kernel similarity $S(K, \hat{K})$ of two kernels, K and $\hat{K}$,

$$S(K, \hat{K}) = \max_{\gamma} \rho(K, \hat{K}, \gamma), \quad (5.1)$$

where $\rho(\cdot)$ is the normalized cross-correlation function and γ is the possible shifts between the two kernels. Let τ represent element coordinates, $\rho(\cdot)$ is given by

$$\rho(K, \hat{K}, \gamma) = \frac{\sum_{\tau} K(\tau) \cdot \hat{K}(\tau + \gamma)}{\|K\| \cdot \|\hat{K}\|}, \quad (5.2)$$

where $\|\cdot\|$ is the 2-norm, and $K(\tau)$ and $\hat{K}(\tau)$ are zero when τ is out of the kernel range. The maximum response of the normalized cross-correlation, similar in nature to

	Figure 5.3		Figure 5.4	
	RMSE	KS	RMSE	KS
kernel in (b)	0.0013	0.8178	0.0090	0.5676
kernel in (c)	0.0016	0.8791	0.0090	0.7521
kernel in (d)	0.0012	0.8106	0.0089	0.7473

Table 5.1: Comparison between RMSE and kernel similarity (KS).

the convolution, can handle the shift and scale problems mentioned above. Moreover, using the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \sum_{\tau} K(\tau) \cdot \hat{K}(\tau + \gamma) &\leq \|K\| \cdot \|\hat{K}\| \\ \implies S(K, \hat{K}) &\leq 1, \end{aligned} \tag{5.3}$$

The equality holds if and only if the non-zero part of K and $\hat{K}$ are linearly dependent. Since for any blur kernels, all the entries of the kernel sum to one, the condition is equivalent to the non-zero parts of K and $\hat{K}$ being the same.

With this metric, the larger kernel similarity values reflect more accurate kernel estimation results. To demonstrate the effectiveness of the kernel similarity metric, we compare it with RMSE on the kernels of Figures 5.3 and 5.4, and the results provided in Table 5.1. In Figure 5.3, low RMSE does not necessarily indicate good results as we discussed before while the kernel similarity values matches the observation. For Figure 5.4, the kernel in (c) and the kernel in (d) maintain more kernel structures and are more similar to the ground truth. In this case, the RMSEs of the estimated kernels are of the same level while the KS values of the kernel in (c) and the kernel in (d) are higher than that of the kernel in (b). As seen from the comparison, the kernel similarity values agree with the visual quality and our understanding of blur kernels.

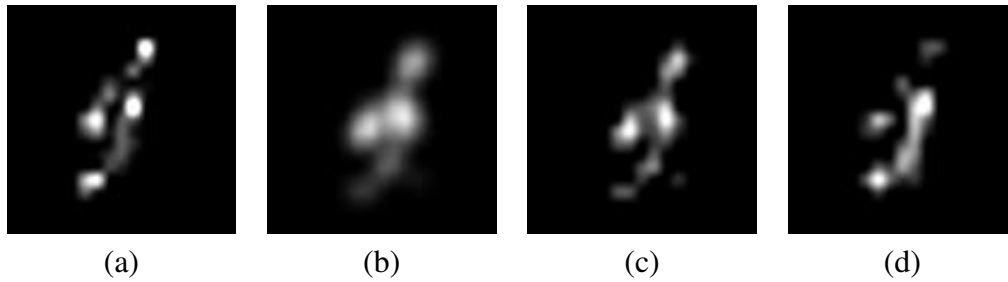


Figure 5.3: The RMSE metric is not effective for kernel similarity. (a) the ground truth blur kernel; (b) the smooth estimated kernel, with RMSE 0.0013 compared to the ground truth; (c) the estimated kernel has an extra part, with RMSE 0.0016; (d) the estimated kernel misses some parts, with RMSE 0.0012.

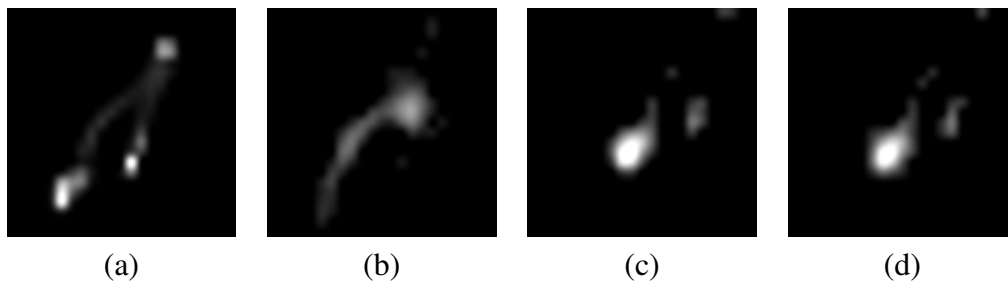


Figure 5.4: Compare RMSE and kernel similarity. (a) ground truth blur kernel; (b) - (d): three estimated kernels, and their RMSEs and kernel similarities compared with ground truth in Table 5.1.

5.3 Learning Good Features

Existing deblurring methods in the literature discuss some potential features that may be useful for kernel estimation mainly based on empirical experimental results. In this chapter, we address this problem by learning good image regions for deblurring.

5.3.1 Learning Framework

To determine the good image regions for deblurring, we analyze the image structure by small subwindows. The subwindows within the image, in the context of kernel estimation and the recovered image, are spatially dependent. Two closely overlapping subwindows (e.g., shifted by a few pixels in either direction) share similar image structures. In addition, it is reasonable to expect that other subwindows nearby a potential good subwindow for kernel estimation, contain useful image structures for deblurring. Consequently, the deblurred results for these subwindows should be similar, which are also observed empirically in image deblurring results. Figure 5.5 shows one example where we estimate a blur kernel from each subwindow of size 200×200 and apply it to recover the whole image. With all the estimated kernels from subwindows and recovered images, we construct an image reconstruction error map (Figure 5.5(b)) and a kernel similarity map (Figure 5.5(c)) by comparing with the ground truth clear image and blur kernel. The value at each pixel of these maps is computed by averaging the reconstruction errors or kernel similarity values from all the subwindows containing it. The image reconstruction map illustrates that deblurred results using subwindows for kernel estimation are spatially correlated. In the meanwhile, high kernel similarity values match low image reconstruction errors which demonstrates the effectiveness of kernel similarity as a metric for evaluating deblurred results. Thus, we label the training data using kernel similarity and pose the problem of learning good regions within a CRF framework [37] as it encourages spatial correlation.

Let S and i represent the set of nodes and node index. Given the labels $\mathbf{y}$ and the observations $\mathbf{x}$, the conditional distribution $P(\mathbf{y}|\mathbf{x})$ is

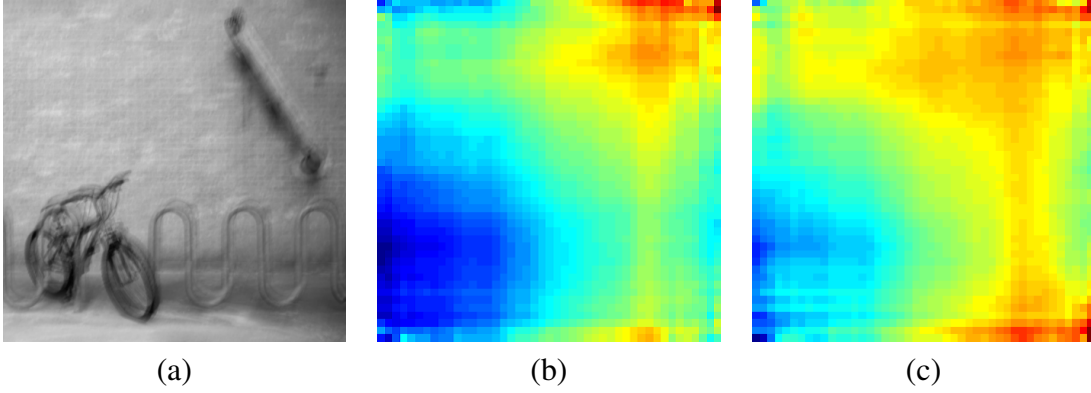


Figure 5.5: Spatial correlation. (a) input blurry image; (b) image reconstruction error map built using the estimated kernels from shifting subwindows (blue to red pixels indicate low to high reconstruction errors compared with the ground truth image); (c) kernel similarity map (blue to red pixels indicate high to low kernel similarity compared with the ground truth kernel).

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp(E(\mathbf{y}|\mathbf{x})), \quad (5.4)$$

where Z is a normalization term also known as the partition function. The energy E is

$$E(\mathbf{y}|\mathbf{x}) = \sum_{i \in S} A_i(y_i, \mathbf{x}) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I_{ij}(y_i, y_j, \mathbf{x}), \quad (5.5)$$

where A_i and I_{ij} denote the association and interaction potentials, respectively. The association potential $A_i(y_i, \mathbf{x})$ measures how likely the node of index i would be labeled y_i given the observation $\mathbf{x}$. Meanwhile, the interaction potential $I_{ij}(y_i, y_j, \mathbf{x})$ determines how the label y_j at node j affects the one y_i at node i . Here $\mathcal{N}_i$ represents the neighborhood of node i .

In this chapter, the subwindows in the input blurred image are considered as nodes in the CRF model similar to the formulation of a discriminative random field [36]. Hence, we formulate the association potential with the log-likelihood of local discriminative model using the logistic function,

$$A_i(y_i, \mathbf{x}) = \log P_1(y_i | h_i(\mathbf{x})), \quad (5.6)$$

where $h_i(\cdot)$ denotes the feature vector of the local region at node i and the first element is set as 1 to accommodate the bias term. The conditional probability $P_1(y_i | h_i(\mathbf{x}))$ of class y_i at node i is defined based on the logistic function:

$$P_1(y_i | h_i(\mathbf{x})) = \sigma(y_i \mathbf{w}^\top h_i(\mathbf{x})), \quad (5.7)$$

where $\mathbf{w}$ are parameters of the logistic function σ .

Similar to the association potential, the interaction potential is given by

$$I(y_i, y_j, \mathbf{x}) = \log P_2(y_i, y_j | \mu_{ij}(\mathbf{x})), \quad (5.8)$$

and

$$P_2(y_i, y_j | \mu_{ij}(\mathbf{x})) = \sigma(y_i y_j \mathbf{v}^\top \mu_{ij}(\mathbf{x})), \quad (5.9)$$

where $\mathbf{v}$ are the parameters of the logistic function and μ_{ij} denotes the feature vector for pair (i, j) . We adopt the difference of feature vector f between node i and j , with 1 as the first element, to express μ_{ij} , $\mu_{ij}(\mathbf{x}) = [1, |h_i(\mathbf{x}) - h_j(\mathbf{x})|]^\top$. Since we do not encourage negative interaction for two nodes of different appearance or at image discontinuities, the term $\mathbf{v}^\top \mu_{ij}(\mathbf{x})$ is set to be non-negative. That is, we use the value $\max(0, \mathbf{v}^\top \mu_{ij}(\mathbf{x}))$ for $\mathbf{v}^\top \mu_{ij}(\mathbf{x})$.

5.3.2 Image Structure

Numerous prior work has shown that smooth image regions do not provide sufficient information for kernel estimation, and instead textured regions are often selected. Nevertheless, the estimation results may still be poor even when textured regions are used [15]. Indeed, regions full of repetitive edges may not make good contributions to the kernel estimation when the blur movement (trajectory of camera movement) occurs in the similar direction as the edges. As shown in Figure 5.6, the stripes on the table cloth and scarf, which share the same direction as the blur motion, remain almost the same before and after blurring. In this case, the homogeneous textures do not help in kernel estimation.

Therefore, gradient orientations in textured regions should also be considered for effective kernel estimation.



Figure 5.6: Texture regions are not always effective for kernel estimation. (a) the latent image. (b) the blurred image.

To estimate blur kernels, recent algorithms focus on the use of sharp edges or edge distribution [32, 15]. Analogous to the problems with textured regions, sharp edges can be of great value for image deblurring under proper assumptions. The underlying assumption for effective use of sharp edges is that regions with high contrast in the original image maintain informative structure after motion blur. However, not all the sharp edges are effective for kernel estimation. Recently, it has been shown in [63] that edges of smaller size than the blur kernel may have adverse effect on kernel estimation, and consequently edge maps of sufficient size are used for deblurring.

Taking all these factors into consideration, we present a method to extract features from regions for image deblurring. We use the responses of a Gabor filter bank $f(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})]$ to represent the oriented textures of an image region $\mathbf{x}$. Here n denotes the bin number of the Gabor filters and $f_i(\mathbf{x})$ represents the proportion that the i -th orientation is the dominant direction within the observation $\mathbf{x}$. The image gradient histogram $g(\mathbf{x}) = [\text{hist}(g_1(\mathbf{x})), \text{hist}(g_2(\mathbf{x}))]$ is used to capture the distribution of edges,

where $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are the first-order derivatives of the image $\mathbf{x}$ along the vertical and horizontal directions.

To rule out potential negative effects from small edges, we use the mask $M(\mathbf{x}) = H(r(\mathbf{x}) - \tau)$ proposed by [63], where $H(\cdot)$ is the Heaviside step function whose value is zero for negative argument and otherwise one, and τ is the threshold. For each pixel $p \in \mathbf{x}$, $r(p)$ measures the usefulness of gradients by

$$r(p) = \frac{\| \sum_{q \in \mathcal{N}_s(p)} \nabla \mathbf{x}(q) \|}{\sum_{q \in \mathcal{N}_s(p)} \| \nabla \mathbf{x}(q) \| + 0.5}, \quad (5.10)$$

where $\mathcal{N}_s(p)$ is an $s \times s$ window centered at pixel p .

The feature vector $h(\mathbf{x})$ is then formed by concatenating the above-mentioned local image features,

$$h(\mathbf{x}) = [f(\mathbf{x}), g(\mathbf{x}), f(M(\mathbf{x})), g(M(\mathbf{x}))] \quad (5.11)$$

with varying parameters n , τ and s . We compare the proposed feature vectors with some alternatives in Section 5.4.2.

5.3.3 Parameter Learning and Inference

Let θ denote the set of parameters in the CRF model, $\theta = \{\mathbf{w}, \mathbf{v}\}$. The maximum-likelihood estimates of the model parameters θ are computed with the pseudo-likelihood to approximate the partition function Z ,

$$\hat{\theta} = \arg \max_{\theta} \prod_m \prod_{i \in S_m} P(y_i^m | \mathbf{x}^m, \mathbf{y}_{\mathcal{N}_i}^m, \theta), \quad (5.12)$$

where m represents the index of the training image and S_m is the graph generated from the m -th image. Based on this formulation, we have

$$P(y_i | \mathbf{x}, \mathbf{y}_{\mathcal{N}_i}, \theta) = \frac{1}{z_i} \exp(A_i(y_i, \mathbf{x}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{x})), \quad (5.13)$$

and the partition function can be written as

$$z_i = \sum_{y_i \in \{-1,1\}} \exp(A_i(y_i, \mathbf{x}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{x})). \quad (5.14)$$

To balance the effect between association and interaction potentials, we add in a penalty term $\frac{1}{2\phi^2} \mathbf{v}^\top \mathbf{v}$, where the variable ϕ is pre-defined in this work. We solve the optimization problem in the log pseudo-likelihood form,

$$\begin{aligned} \hat{\theta} = \arg \max_{\theta} \sum_m \sum_{i \in S_m} [\log \sigma(y_i \mathbf{w}^\top h_i(\mathbf{x})) + \\ \log \sum_{j \in \mathcal{N}_i} \sigma(y_i y_j \mathbf{v}^\top \mu_{ij}(\mathbf{x})) - \log z_i] - \frac{1}{2\phi^2} \mathbf{v}^\top \mathbf{v}. \end{aligned} \quad (5.15)$$

Since Eqn. 5.15 is a convex optimization problem with respect to the parameter θ , the initial setting of θ does not affect the final results. We use the BFGS method to solve the optimization problem and obtain parameter $\hat{\theta}$. Loopy belief propagation (LBP) [67] is then utilized for inference.

5.3.4 Good Regions to Deblur

Given a high resolution blurred image (e.g., 3000×2000 pixels), it is rather time consuming to apply deblurring algorithms with the whole image (about 65 seconds using [12]). Even with a fast deblurring algorithm [12], it may not be the best choice to use the whole image for kernel estimation for the reasons discussed in Section 5.3.2. We will demonstrate in Section 5.4 that using a region of the blurred image for kernel estimation may render better deblurred result rather than using the whole image. Clearly, one immediate solution for these problems is to select a region within the input image to estimate the blur kernel, and then apply a non-blind deconvolution algorithm to the whole image.

In this work, we use principal component analysis to reduce the dimensionality of the feature vectors for learning the model parameters, and LBP to classify the subwindows as good regions or not for blur kernel estimation. The label of each node (subwindow)

in the training image is determined by comparing the estimated kernel and the ground truth kernel. The reason we use LBP for inference instead of graph cuts here is that we would like to obtain labels with confidence values so that it is more convenient to develop a region selection strategy. We select the top ranked subwindow to estimate the kernel for simplicity although other weighted approaches may be used. Given the window size, the proposed method of selecting a good subwindow to deblur does not require manual selection which can be problematic and time-consuming. We show that the proposed method can effectively select the optimal subwindow for deblurring in Section 5.4.

Moreover, this window selection method can also be applied to non-uniform (spatially variant) blur cases which require large memory space and significant computation if the whole image is used to estimate the camera motion (e.g., [12]). For instance, the non-uniform deblur algorithm [21] employs a RANSAC-based scheme to select a set of patches for alternatively optimizing camera motion and reconstructed patches. With our method, it would be easy and effective to choose a set of good patches for deblurring.

5.3.5 Contribution of Feature Components

Within the model presented in Section 5.3, we use a feature vector consisting of several components. In this section, we determine the most important component for image deblurring via their structured output. In addition, the approach we use can also be extended to determine other effective features. We use a slightly different energy function and consider each feature component independently. That is, we decompose the associate potential $A_i(y_i, \mathbf{x})$ into a combination of feature components:

$$A_i(y_i|\mathbf{x}) = \sum_c \alpha_c \log P_1^c(y_i|h_i^c(\mathbf{x})), \quad (5.16)$$

where h_i^c represents different components of the feature vector, e.g., $f(x)$ and $f(M(x))$, with weights α . The conditional probability $P_1^c(y_i|h_i^c(\mathbf{x}))$ is defined based on the same logistic function as Eqn. 5.7. Since we aim to determine the weights of feature components

rather than the parameters of the logistic functions, the parameters of logistic functions are assumed to be known (e.g., learned with a linear support vector machines). For the interaction potential, we use $I(y_i, y_j, \mathbf{x}) = \beta \delta(y_i - y_j)$ to encourage labeling smoothness, where $\beta > 0$ is the weight and $\delta(\cdot)$ denotes the Kronecker function.

The objective here is to maximize the probability of labels. In the training phase, we compute the weights $\Lambda = (\alpha, \beta)$ that assign the training labels $\hat{\mathbf{y}}^m$ higher or equal probabilities than any other labels $\mathbf{y}^m$ of the training image m ,

$$P(\hat{\mathbf{y}}^m | \mathbf{x}^m) \geq P(\mathbf{y}^m | \mathbf{x}^m), \forall \mathbf{y}^m \neq \hat{\mathbf{y}}^m. \quad (5.17)$$

We can cancel the normalization term Z from both sides of the constraints and express the constraints in terms of energies,

$$E(\hat{\mathbf{y}}^m | \mathbf{x}^m) \geq E(\mathbf{y}^m | \mathbf{x}^m), \forall \mathbf{y}^m \neq \hat{\mathbf{y}}^m. \quad (5.18)$$

We take a max-margin approach to compute the weights that satisfy the inequalities with the largest energy margin γ ,

$$\max_{\Lambda: \|\Lambda\|=1} \gamma \quad \text{s. t.} \quad E(\hat{\mathbf{y}}^m | \mathbf{x}^m) - E(\mathbf{y}^m | \mathbf{x}^m) \geq \gamma, \forall \mathbf{y}^m \neq \hat{\mathbf{y}}^m. \quad (5.19)$$

The weights Λ are constrained to have unity norm to prevent them from growing without bound. Similar to [53], we use the transformation $\|\Lambda\| \leftarrow \|\Lambda\|/\gamma$ and add slack variables ξ_m to make the program more robust to noisy training data. To enforce a relatively larger margin for labels that are far from the training labeling, we rescale the margin as proposed in [58] with a Hamming loss $\Delta(\hat{\mathbf{y}}^m, \mathbf{y}^m) = \sum_i \delta(\hat{y}_i^m, y_i^m)$ and rewrite the objective function as

$$\begin{aligned} \min_{\Lambda} \quad & \frac{1}{2} \|\Lambda\|^2 - \frac{C}{M} \sum_m \xi_m \quad \text{s. t.} \quad \forall m \\ & E(\hat{\mathbf{y}}^m | \mathbf{x}^m) - E(\mathbf{y}^m | \mathbf{x}^m) \geq \Delta(\hat{\mathbf{y}}^m, \mathbf{y}^m) - \xi_m, \\ & \xi_m \geq 0, \forall \mathbf{y}^m \neq \hat{\mathbf{y}}^m \\ & \Lambda \geq 0 \end{aligned} \quad (5.20)$$

where C denotes the slack penalty parameter and M is the number of training images. Here we enforce non-negative weights to compare the contribution of different feature components for image deblurring.

Since the Hamming loss can be decomposed across nodes, it is absorbed into the node energies into a refined energy function $E'(\mathbf{y}^m|\mathbf{x}^m) = E(\mathbf{y}^m|\mathbf{x}^m) + \Delta(\hat{\mathbf{y}}^m, \mathbf{y}^m)$. To learn the weights, we iteratively find the minimum energy labeling $\mathbf{y}_\star$ of training image m using graph cuts [5] with the refined energy. If $\mathbf{y}_\star \neq \hat{\mathbf{y}}^m$, we update the weights Λ to enforce the margin. The update process terminates when $\mathbf{y}_\star = \hat{\mathbf{y}}^m$ is satisfied for all training images.

5.4 Experimental Results

We evaluate the inferred subwindows using three state-of-the-art deblurring algorithms [18, 49, 12] and compare the performance using the error metric introduced in [40]. This metric computes the difference between a recovered image I_r and the known ground-truth sharp image I_g , over the difference between the deblurred image I_{k_g} with the ground truth kernel k_g and the ground-truth sharp image as $\|I_r - I_g\|^2 / \|I_{k_g} - I_g\|^2$. The cumulative histogram of reconstructed error ratio (briefly cumulative error histogram for convenience) is then used to evaluate the efficiency of an algorithm. Since the dataset from [40] has limited variability of image structures and blur kernels, we use the same technique to collect 400 blurred image using 20 sharp images (with image size around 500×500 pixels) and 20 blur kernels for training. Furthermore, we construct another set of 120 challenging blurred images using 10 sharp images and 12 blur kernels for tests. Our experiments are carried out on a machine with a 3.40 GHz CPU and 16 GB RAM. To infer an image of 450×450 pixels takes around 5 seconds with our MATLAB implementation.

5.4.1 Deblurring with Good Regions

With the CRF model described in Section 5.3, we learn the parameters using all the 400 training images for inferring good regions to deblur. For each image, we build the graph model with overlapping subwindows of 200×200 pixels as nodes and shifts of 20 pixels. The size of the subwindows is determined empirically as it is large enough to estimate a kernel of size smaller than 29×29 pixels and all the blur kernels in this data set are like that. In our experiments, there are at least 484 subwindows extracted from each image as the nodes for training. We collect the estimated kernel from each subwindow using [18] as it is one of the best kernel estimators [40]. With the estimated kernels and ground truth kernels, we label node i to be 1 or -1 using the proposed kernel similarity $S(K_i, K)$ between an estimated kernel K_i and the ground truth kernel K with the threshold λ of 0.6 (empirically set for effectiveness and efficiency), where K_i represents the estimated kernel for node i . The feature parameters n , τ and s in this work are set to be 8, 0.6 to 0.9 with increments of 0.1, and 9 to 24 with increments of 4 respectively. During the inference process, the size of subwindows is set proportional to the user-defined kernel size. That is, for a larger blur kernel, the size of subwindows selected to estimate the kernel should also be increased proportionally.

Comparison with User-selected Region

To demonstrate that user selection can be replaced by our algorithm, we compare our top inferred subwindows with user-selected regions. The users tend to choose regions with the most salient edges and variation as shown in Figure 5.7(b)(c) and Figure 5.8(b)(c). The user selection strategy works well in some situations but usually requires several trials to obtain a good result. On the contrary, our proposed method does not require user guidance and the deblurred results by the inferred subwindows of our algorithm outperform that using user-selected regions.

Comparison with Region Selection Algorithms

We compare the proposed algorithm with two other region selection methods for deblurring [18, 63]. The automatic subwindow selector by [18] searches for the regions with high variance and low saturation for kernel estimation. We implement another subwindow selection method by following the idea in [63] that the image deblurring process may be adversely degraded by the negative effect of small edges. In this selector, we remove small edges using the mask $M(\mathbf{x})$ and then search for the subwindow with the most salient edges.

In this experiment, we select the top ranked subwindow from our inference algorithm for blur kernel estimation for simplicity although other alternatives may be used. We apply other methods to select one subwindow from each image for kernel estimation and then apply two state-of-the-art algorithms [18, 49] to recover the whole image. Figure 5.7, Figure 5.8 and Figures 5.9-5.10 illustrate the comparison using the deblurring algorithm of [18], [49] and [12] respectively. The deblurring results on subwindows inferred by our algorithm are better than those from the other two subwindow selection strategies.

Comparison with Deblurring Using the Whole Image

We also compare with the deblurred results by [12] which uses the whole image for kernel estimation. In this experiment, we test two region selection strategies based on the inference results. One is to select the top ranked subwindow to estimate the blur kernel (as shown in Figure 5.9 and Figure 5.10), and another is to combine the top ten good subwindows. To combine the top ten subwindows, we choose the smallest rectangle which covers all the subwindows as the region for simplicity. We note that the top ranked subwindows are usually clustered due to the use of the CRF model which encourages spatial correlation, and thus the rectangular region is still of small size compared with the whole image as shown in Figure 5.11. Compared to the results obtained from the whole images, our algorithm with the inferred region generates comparable or superior kernel

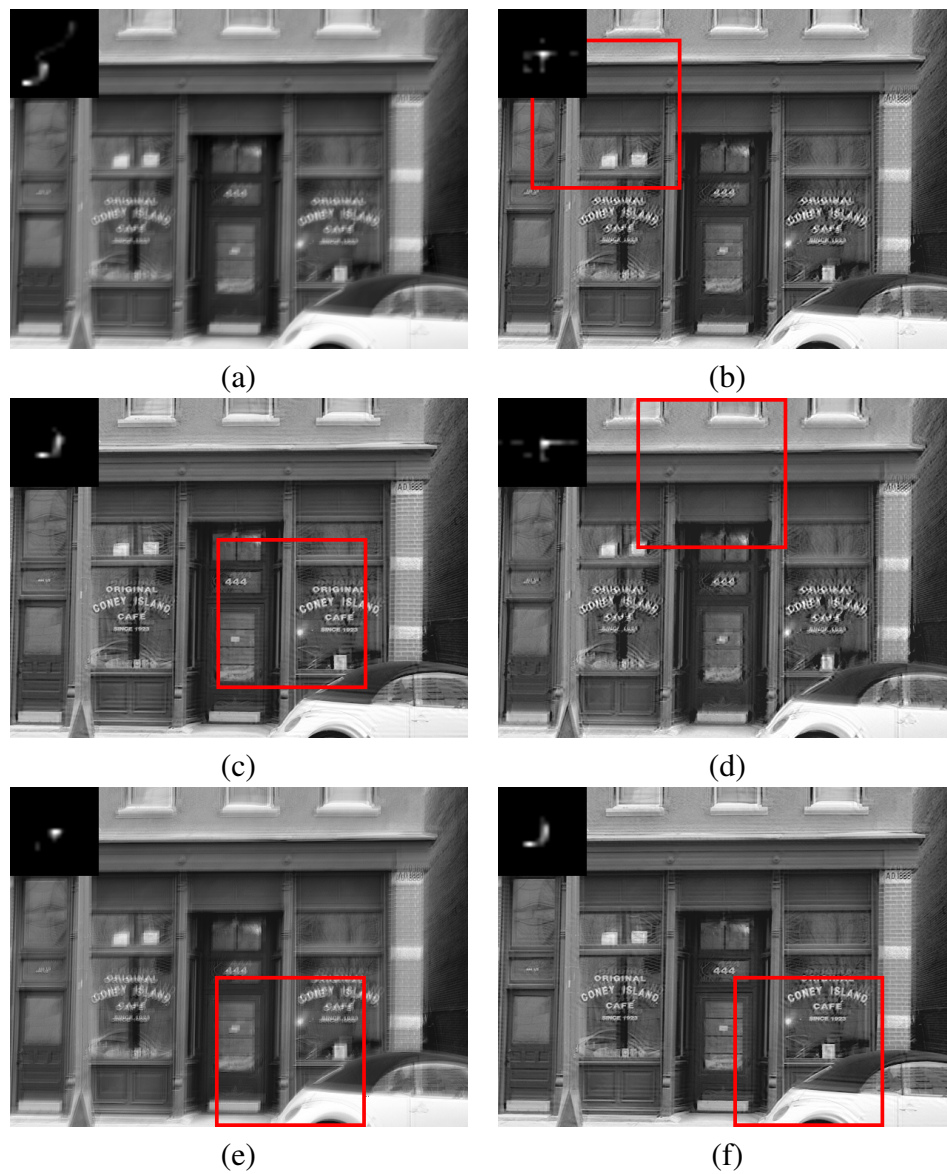


Figure 5.7: Comparison on different window selection strategies for deblurring using [18]. (a) input blurry image and corresponding kernel; (b) & (c) user-selected subwindows (red box) and the deblurred results; (d) subwindow selected by window selector [18] and the deblurred result; (e) subwindow selected by window selector following [63] and the deblurred result; (f) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display).

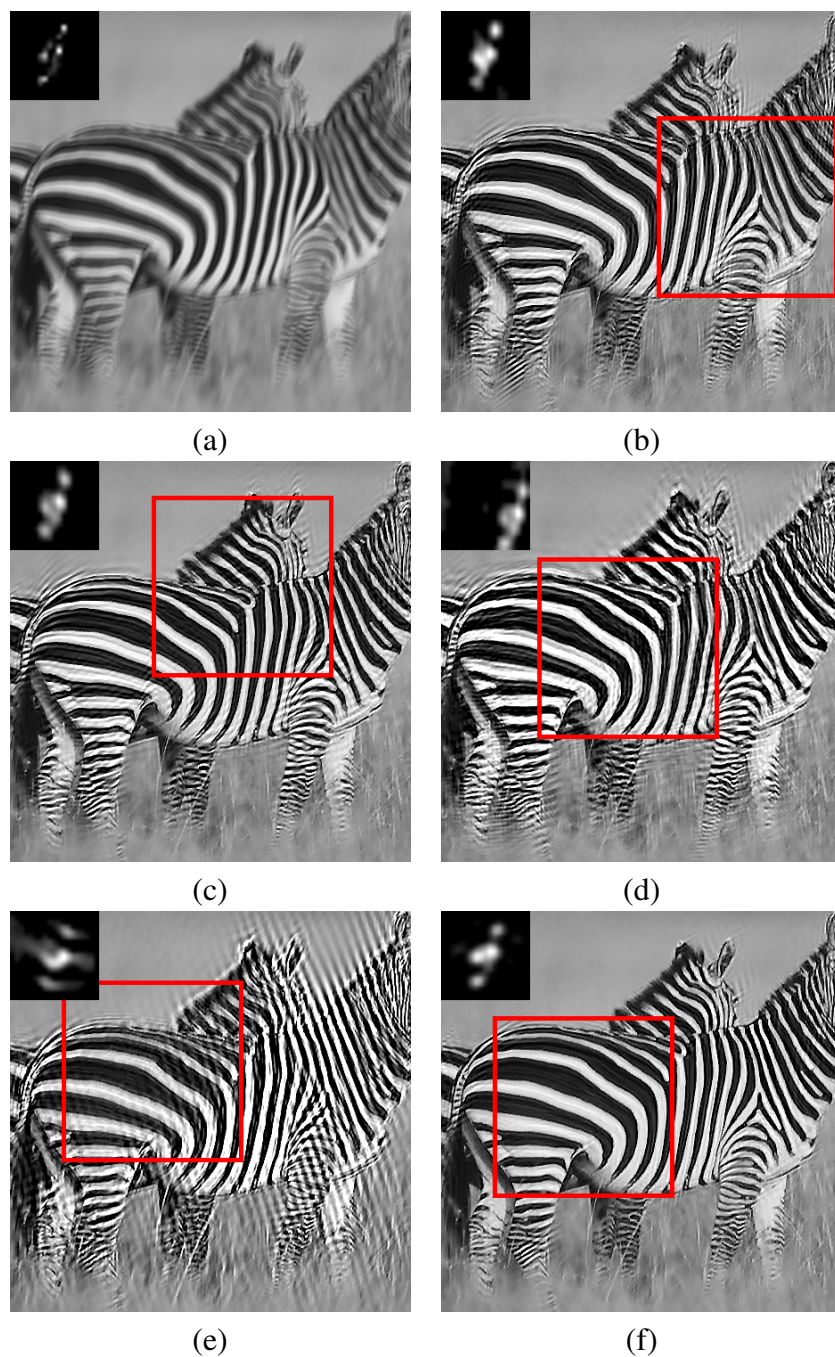


Figure 5.8: Comparison of different window selection strategies for deblurring using [49]. (a) input blurry image and corresponding kernel; (b) & (c) user-selected subwindows (red box) and the deblurred results; (d) subwindow selected by window selector [18] and the deblurred result; (e) subwindow selected by window selector following [63] and the deblurred result; (f) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display).

estimation and reconstructed images.

Quantitative Comparison

We conduct extensive comparisons using 120 challenging test images and present their cumulative error histograms. Given an inferred region by the proposed algorithm, we use a fast algorithm [12] to estimate a blur kernel, and a non-blind deconvolution algorithm [49] to recover the latent image (similar to [12]). For thorough evaluations, we compare the results using the above-mentioned region selection strategies [18] and the whole image [12]. In addition, we compare with [63] using a region or the whole image for kernel estimation. As shown in Figure 5.12, the curve using our top ten subwindows generally performs better than other algorithms [12, 63] using the whole image. The results also show that not all the information in a blurred image is useful and using the whole image for kernel estimation may not be the best choice. We note that the reconstructed results are visually plausible even when the error ratio is around 5, which is different from the observations in [40]. The reason is that we employ different non-blind deconvolution algorithms and larger test images.

Analysis

Our empirical results also provide some insights about the kinds of image structures or features that are favored by different deblurring algorithms. In the *zebra* image, the selected window by our algorithm contains relatively strong edges, as suggested in [18, 40]. However, small or detailed edges may not help kernel estimation as shown in Figure 5.8(b). Furthermore, the subwindow with various oriented edges is selected rather than the regions with repetitive textures. The empirical results and observations bear out our intuition and analysis in Section 5.3.2.

To further provide insight on what structures within image patches constitute good regions for deblurring, we provide more results of inferred good regions in the same scene

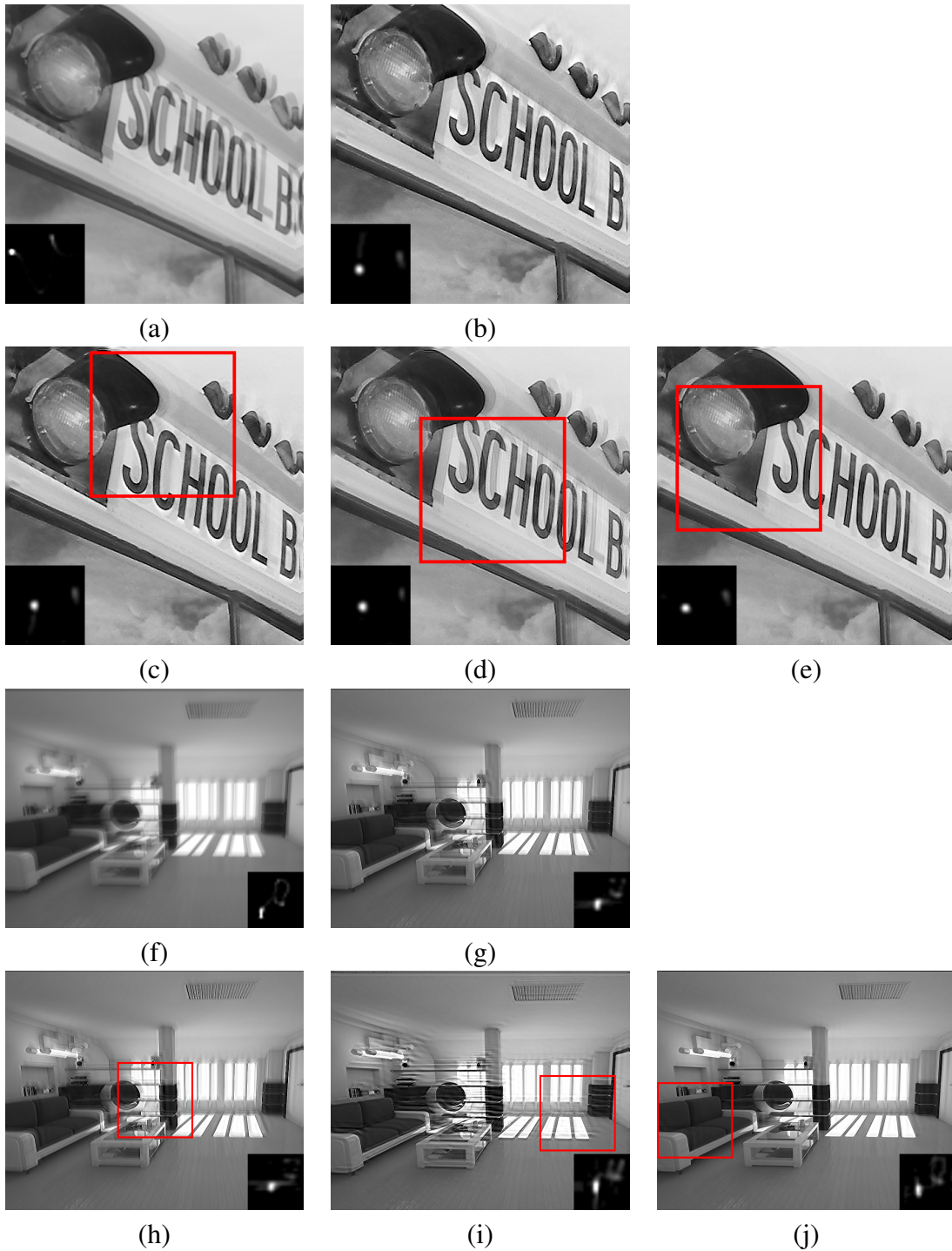


Figure 5.9: Comparison on different window selection strategies for deblurring using [12]. (a)(f) input blurry image and corresponding kernel; (b)(g) the kernel estimation and deblurred result using the whole image; (c)(h) subwindow selected by window selector [18] and the deblurred result; (d)(i) subwindow selected by window selector following [63] and the deblurred result; (e)(j) subwindow selected by our algorithm and the deblurred result (best viewed on high-resolution display).

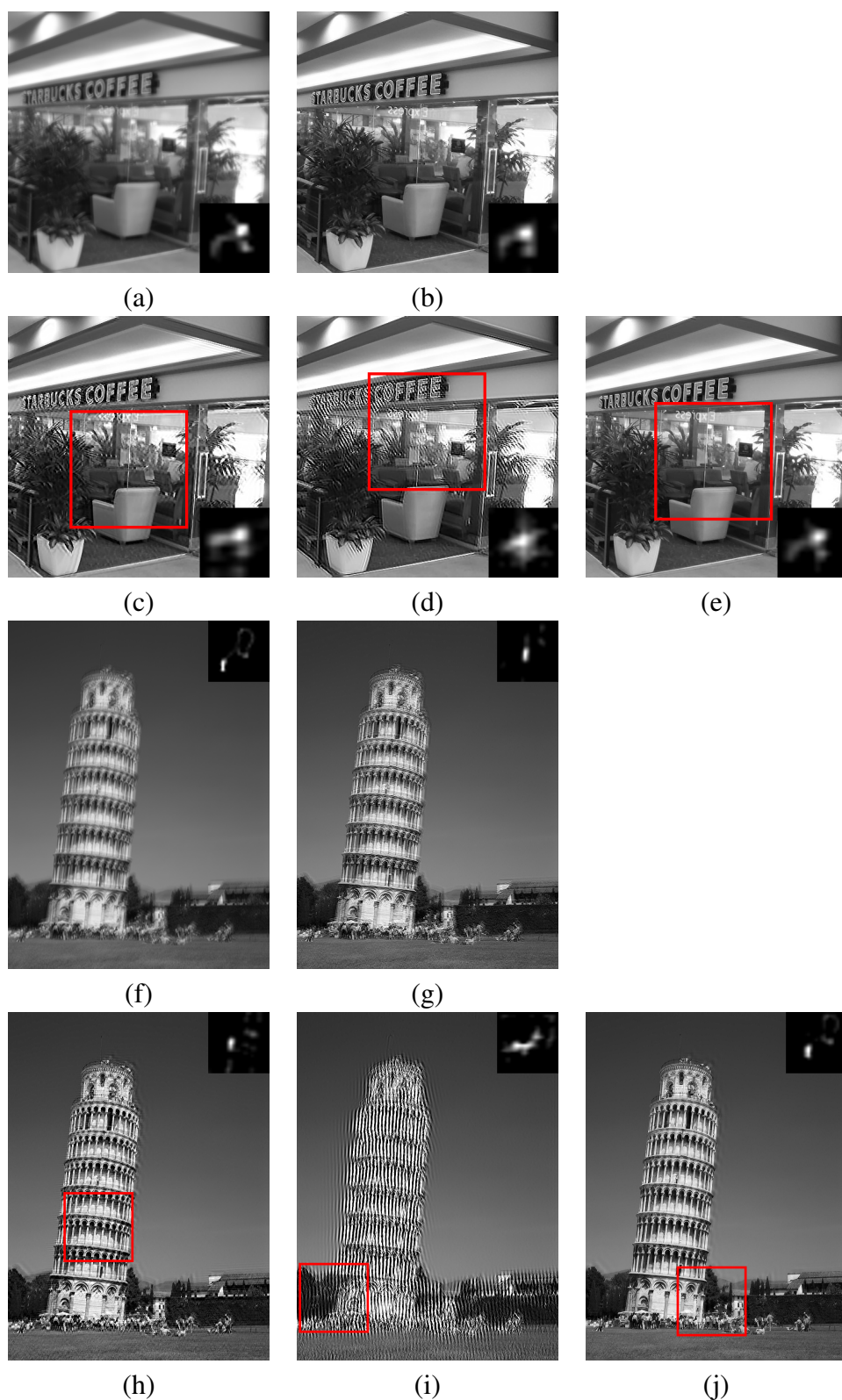


Figure 5.10: Comparison on different window selection strategies for deblurring using [12]. (a)(f) input blurry image and corresponding kernel; (b)(g) the kernel estimation and deblurred result using the whole image; (c)(h) subwindow selected by window selector [18] and the deblurred result; (d)(i) subwindow selected by window selector fol-

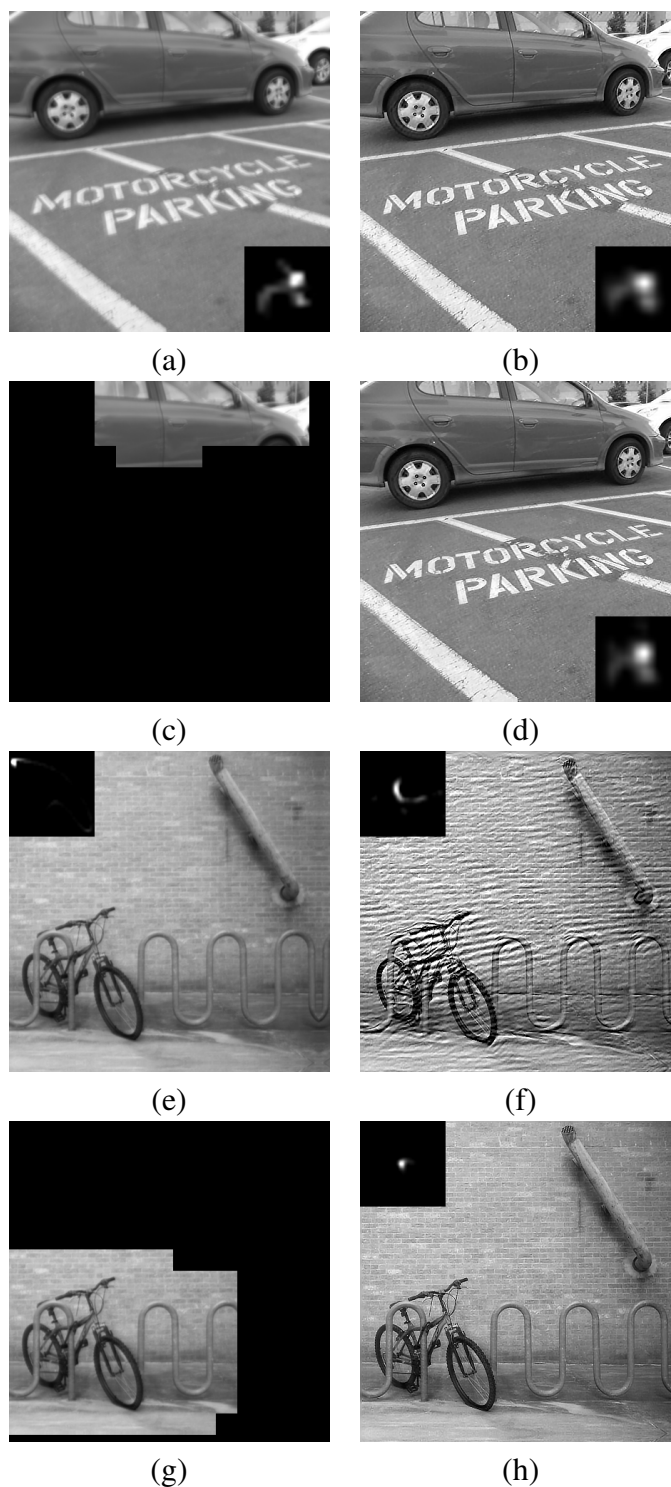


Figure 5.11: Examples of inferred region with top ten good subwindows to deblur using our algorithm. (a)(e) input blurred image and the corresponding blur kernel; (b)(f) the kernel estimation and deblurred result using the whole image; (c)(g) the mask image indicating the inferred good region; (d)(h) the deblurred result using the good region to deblur (best viewed on high-resolution display).

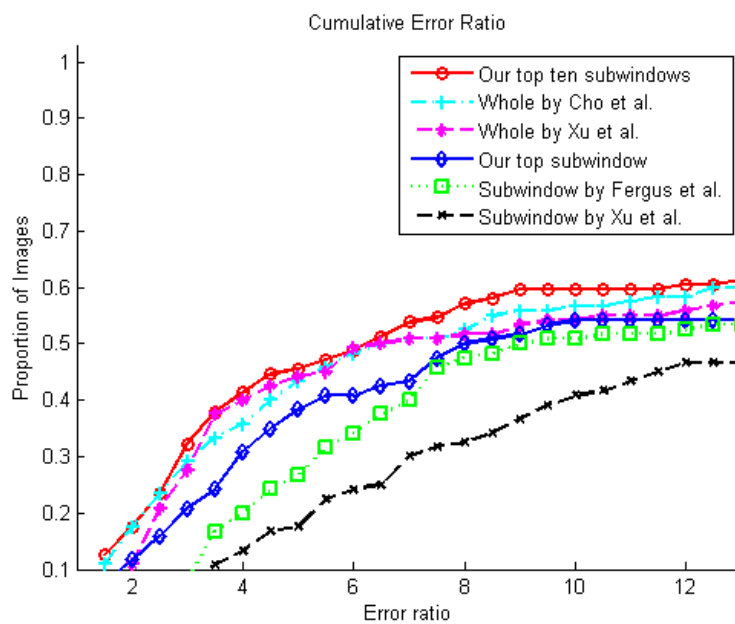


Figure 5.12: Cumulative error histograms on deblurred results with different window selection methods including ours, [18] and [63]. We use the algorithm in [12] for kernel estimation and the algorithm in [49] for non-blind deconvolution.

but different blur kernels in Figure 5.13 and Figure 5.14. The inferred regions in the same scene vary a bit as a result of different blur kernels. This can be explained by the fact that different blur kernels change the image structure in different ways.

5.4.2 Contribution of Feature Components

With the model described in Section 5.3.5, we learn the weights Λ from a set of 400 images. The results show that the weights α_c for feature components $f(\mathbf{x})$ and $f(M(\mathbf{x}))$ of Eqn. 5.11 have the highest values. This indicates that Gabor filter responses are more discriminative for this task, while using the original blur image is more effective than using the image after removing small edges.

5.4.3 Feature Alternatives

We evaluate alternative features, i.e., gradient histogram and the histogram of Gabor filter responses, that can be used to describe local image structures for kernel estimation. We follow the same process as we describe in Section 5.4.1 of the manuscript, but replace the proposed feature with the alternative features. Figure 5.15 shows the cumulative error histograms of methods with different features, and the results with our proposed feature (Section 5.3.2) are better than those with alternative features. In addition, the proposed algorithm performs better than the method [12] using the entire image.

5.5 Conclusion

Recent work focuses on introducing priors or additional information to facilitate deblurring, and considerably less effort has been made to study the structural information of the input blurry image itself. We address this issue and exploit informative image structure for deblurring. In this chapter, we propose a learning-based approach for selecting good features and good patches for blur kernel estimation. Towards this, we introduce a kernel

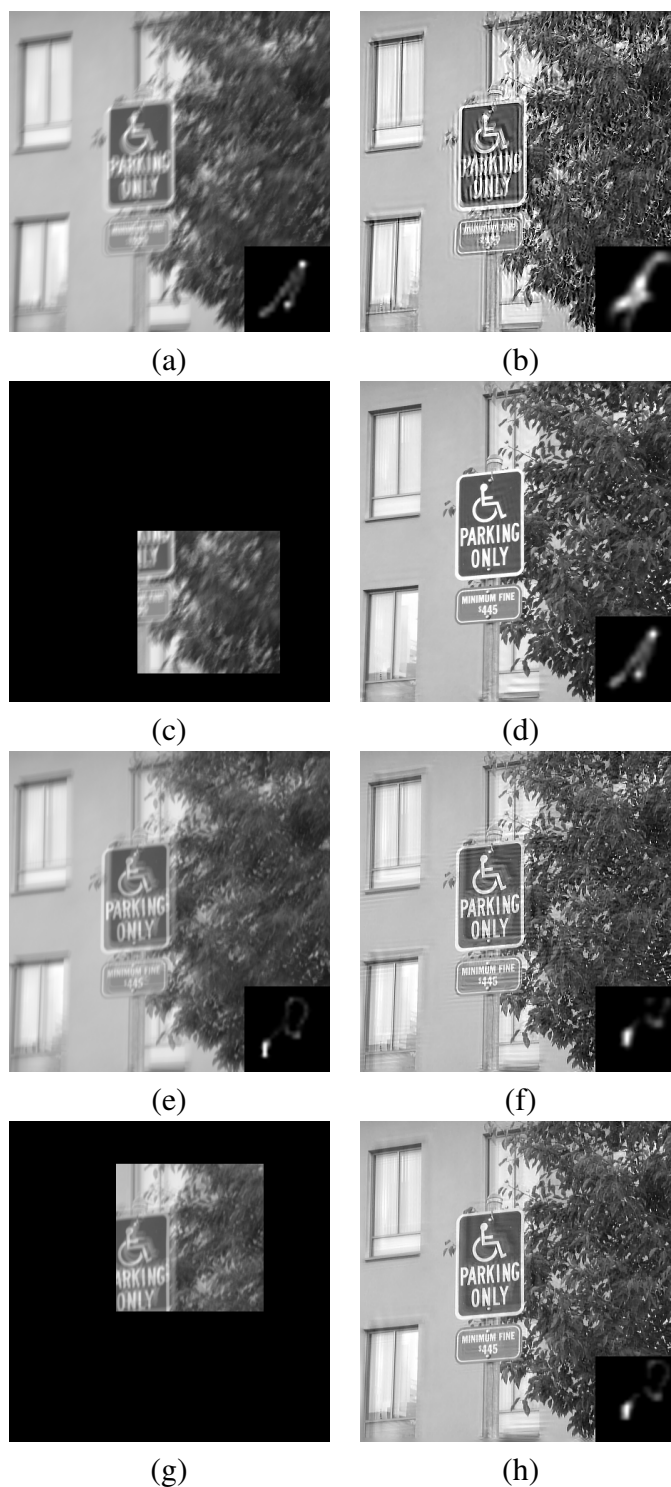


Figure 5.13: Inferred top subwindows from our algorithm are different for the images of the same scene but with different blur kernels. (a)(e) input blurry image and corresponding blur kernel; (b)(f) the kernel estimation and deblurred result using the whole image; (c)(g) the mask image indicating the inferred top subwindow; (d)(h) the deblurred result using the good region to deblur (best viewed on high-resolution display).

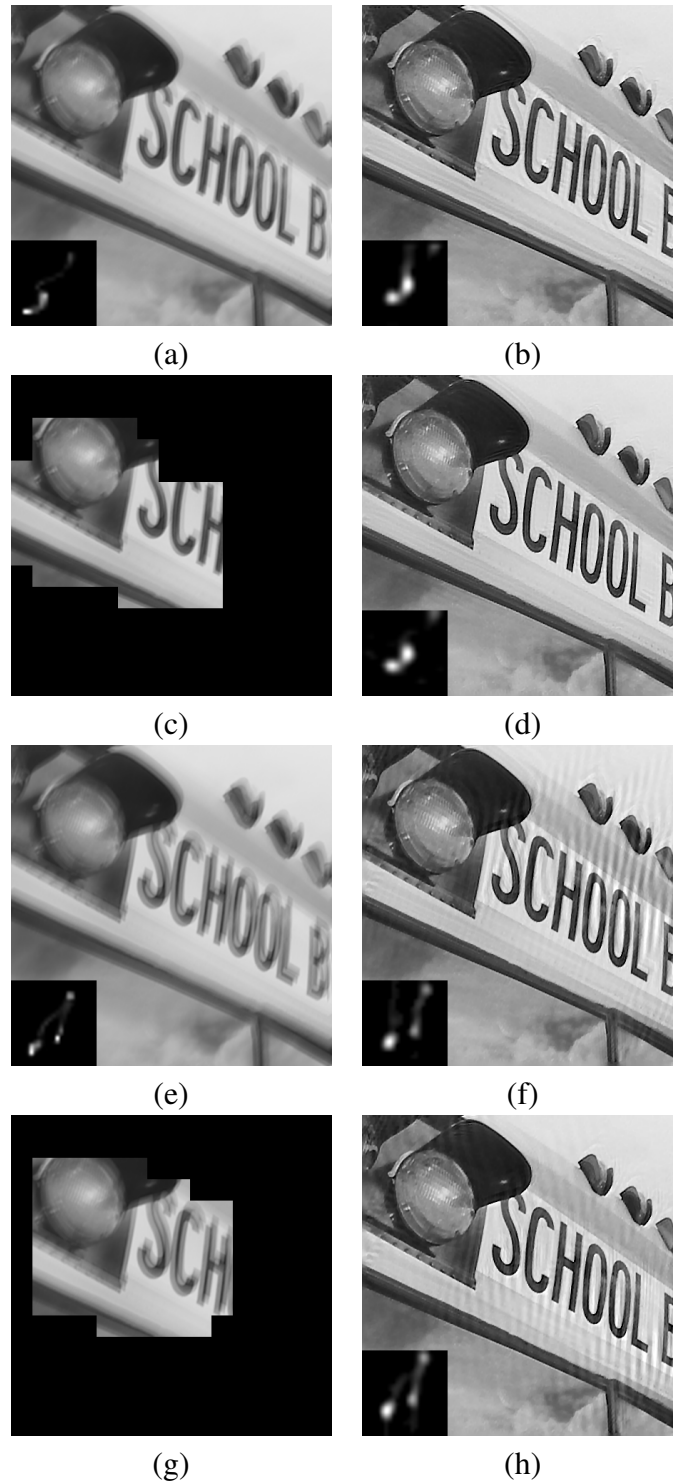


Figure 5.14: Inferred regions from our algorithm are different for the images of the same scene but with different blur kernels. (a)(e) input blurry image and corresponding blur kernel; (b)(f) the kernel estimation and deblurred result using the whole image; (c)(g) the mask image indicating the inferred good region; (d)(h) the deblurred result using the good region to deblur (best viewed on high-resolution display).

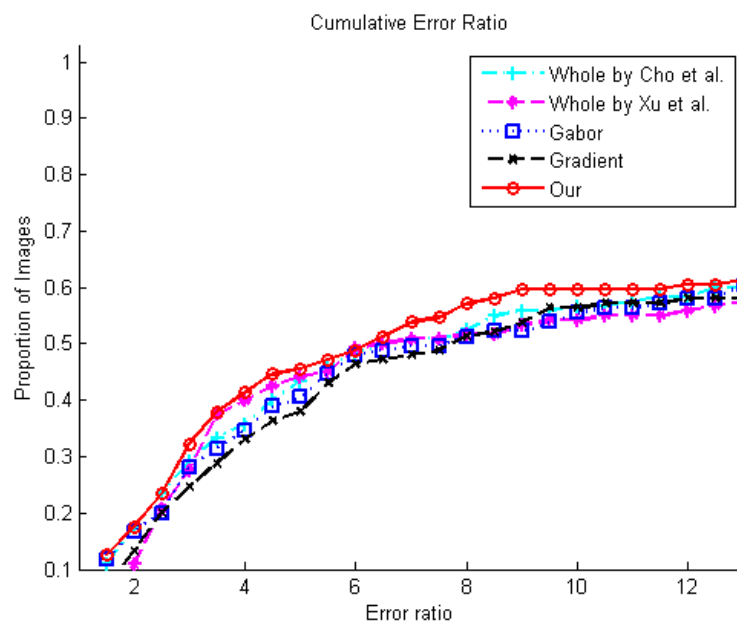


Figure 5.15: Cumulative error histograms on deblurred results with different feature extraction approaches. We use the algorithm in [14] for kernel estimation and the algorithm in [9] for non-blind deconvolution.

similarity metric for effective comparison of kernels. The proposed algorithm selects good regions automatically for deblurring with efficient and favorable results, thereby relieving users from tedious trials for selecting image patches.

Chapter 6

Fast Non-uniform Deblurring using Constrained Camera Pose Subspace

6.1 Introduction

Image deblurring is one of the most fundamental problems in computer vision that has been studied extensively. Image blur often results from relative motion between a camera and the scene (e.g., camera shake) during the exposure time. It causes significant image degradation, especially in low light conditions where longer exposure time is required. Although the blur effect may be reduced by setting a faster shutter speed, it inevitably generates a significant amount of noise.

Most existing algorithms formulate image blur as a convolution process with a uniform blur kernel or point spread function (PSF) on a latent input image. The uniform blur model is sufficient if the camera shake is caused only by in-plane translation (i.e., within the sensor plane) without any out-of-plane rotations. However, this assumption does not hold in general. Camera tilt and rotation happen frequently and the image blur becomes non-uniform, i.e., blurs at different pixel locations are not the same. Moreover, object movement in the scene also leads to non-uniform blur effects. Thus, blur kernel estimation

is especially difficult in the non-uniform case.

Early work addresses the non-uniform blur problem by restricting the blur process to simple cases [2, 13, 50]. Recently, a geometric model has been proposed to describe the camera motion using the weighted sum of homographically transformed copies of the sharp image [60]. The transformed copies can be viewed as the intermediate images observed by the camera during the camera shake process. This model imposes a global constraint on local blur kernels, and the estimation step becomes computing the weights of different camera poses rather than directly recovering spatially varying blur kernels at different pixel locations. The main disadvantage of this model is the heavy computational load due to dense sampling of poses in the high dimensional camera motion space. To make this problem solvable in practice, recent methods approximate the six-dimensional motion space with a three-dimensional subspace (roll and in-plane translation) and coarse-grained sampling. Nevertheless, this simplification inevitably introduces reconstruction errors. Most recently, an approach that uses efficient filter flows to address the computational issues was proposed [24]. However, this algorithm aims to approximate local blur kernels without exploiting camera motion structure (e.g., camera motion trajectory is sparse in the camera motion space).

In this chapter, we propose a single image deblurring algorithm to remove non-uniform blur. We first introduce an initialization method that facilitates convergence and avoids local minimums of the formulated optimization problem. We then propose a new camera motion estimation method which optimizes a small set of pose weights of a constrained camera pose subspace at a time rather than using the entire space. We develop an iterative method to refine the camera motion estimation and introduce perturbation at each iteration to obtain robust solutions. The proposed method also enables us to use dense sampling and the six-dimensional motion space without heavy computational load. We evaluate our method with the state-of-the-art algorithms and the experimental results show that the proposed approach is not only efficient but also effective for non-uniform image deblurring.

6.2 Related Work

The problem of removing spatially invariant blur has been studied extensively and numerous algorithms have been proposed [18, 46, 38, 50, 56, 40, 12, 63, 15, 16]. Since blind deconvolution is an ill-posed problem, it is often necessary to exploit prior knowledge and additional information to obtain effective solutions. For single image deblurring, recent approaches focus on exploring image priors for deblurring. Image priors are usually formulated as constraints or regularization terms in the optimization function so that satisfactory results can be achieved without searching for the entire solution space. In [18], the heavy-tailed gradient distribution of natural images is exploited as prior information. The recovered latent image is constrained to fit the distribution of gradient magnitudes of natural images described by a Gaussian mixture. The underlying relation between motion blur and object boundaries is analyzed and exploited for deblurring [30]. In [49], a deblurring algorithm is proposed in which prior knowledge regarding the gradient distribution of natural images is used along with consistency constraints of local smooth regions before and after blurring. The consistency constraints are shown to be effective in suppressing ringing effects. A method that uses sparsity constraints of both the blur kernel and latent image was presented in [6]. Most recently, a fast method with carefully designed regularizations and optimization procedures was proposed for image deblurring [12].

Removing spatially variant blur is a challenging task in which regions are degraded differently, thereby entailing non-uniform models over the entire image. This problem has recently attracted more attention due to its wide range of practical applications. One approach to handle spatially varying blur is to segment the blurry image into multiple regions where each one is modeled by a uniform blur kernel [2, 13]. This approach is sensible and well-known uniform image deblurring algorithms can be used once the segments are determined. Such segmented regions, however, often need to be small for the assumption of using uniform blur kernels to hold. Another approach is to model or capture the camera motion during the blur process and derive spatially varying blur kernels at different sites.

In [50], an algorithm to account for blurs caused mainly by the in-plane rotational motion from a single image is proposed. Specially designed imaging systems (i.e., hybrid cameras and sensors) have also been used to estimate a local blur kernel at each pixel location based on optical flow [56, 31]. The use of additional hardware to facilitate image deblurring is demonstrated to be effective. Nevertheless, it is not clear how such approaches can be applied to single frame image deblurring in the general setting.

Recently, a geometric model has been proposed to model the observed blurry image as the integration of all the intermediate images captured by the camera along the motion trajectory [60, 57]. These intermediate images are modeled transformation (i.e., homographies) of the sharp image. Based on this model, image blur caused by camera motion can be formulated as a single optimization problem. It is also possible to remove spatially varying blur by estimating a general camera motion function. A similar model has been used to model three degrees of camera motion (with in-plane translation and rotation) in [21]. In [24], a method that employs an efficient filter flow algorithm to speed up the optimization step is developed. However, these approaches entail high computational cost as the high-dimensional camera motion and latent image have to be computed during the iterative optimization process.

6.3 Non-uniform Blur Model for Camera Shake

In this section, we describe the non-uniform blur model caused by camera shake for the scene with constant depth that can be modeled in a way similar to [21]. The geometric model we use is based on [60] which formulates the blurry image B as the integration of the transformed copies of latent image L ,

$$B = \int f(H_\theta, L)w(\theta)d\theta + n, \quad (6.1)$$

where θ denotes the camera pose characterized by 6 parameters, H_θ is the homography induced by camera pose θ , and $f(H_\theta, L)$ represents the transformed image. The blurry

image can be viewed as a combination of the images the camera captures with observation noise n . Similar to [21], we address this problem in a discrete space with

$$B = \sum_{\theta \in S} f(H_\theta, L) w_\theta + n = \sum_{\theta \in S} w_\theta (K_\theta L) + n, \quad (6.2)$$

where K_θ is the matrix that warps the latent image L to the transformed one at a sampled pose θ and S denotes the set of sampled camera poses. In this formulation, K_θ can also be viewed as a blur kernel basis where row i is the weight vector representing how pixel i of the latent image L contributes in the blurry image. We note that K_θ is a sparse square matrix which can be stored in memory efficiently. Thus, we model the blur process as

$$B = \mathcal{K}L + n, \quad (6.3)$$

where $\mathcal{K} = \sum_{\theta \in S} w_\theta K_\theta$ represents a blur matrix with each row as a point spread function (PSF) at a pixel location. For spatially invariant blur, each row of $\mathcal{K}$ shares the same PSF but with location shifts. We denote W as the set of weights w_θ .

To reduce the dimensionality, one approach is to approximate the six-dimensional camera pose space by three-dimensional rotations without considering translations [60]. In [21], it is shown that the out-of-plane rotation can be well approximated by x, y translation when the focal length is sufficiently long, which is usually the case for camera settings. In this work, we approximate the camera motion with a three-dimensional pose space as in [21] (x, y translation and z rotation) and focus on the estimation of pose weights.

6.4 Deblurring with Constrained Pose Subspace

The purpose of single image deblurring is to recover the sharp image L given the blurry observation B . To solve L in Eqn. 6.3, we seek to minimize the energy function over the variables L and W with kernel basis K_θ ,

$$\min_{(L, W)} ||(\sum_{\theta \in S} w_\theta K_\theta)L - B||^2 + \Phi_1(L) + \Phi_2(W), \quad (6.4)$$

where Φ_1 and Φ_2 represent the regularization terms on the latent image L and weights W respectively. The regularization terms will be explained in the following sections.

In this work, we develop an iterative method to solve L and W . We first initialize W with patch-based uniform deblurred results, and then estimate L and W via iterative optimization. At each iteration, we restrict the solution space in a small set of camera pose space which the camera motion trajectory is most likely to belong to. Figure 6.1 summarizes the main steps of our method and the technical details are presented in the following sections.

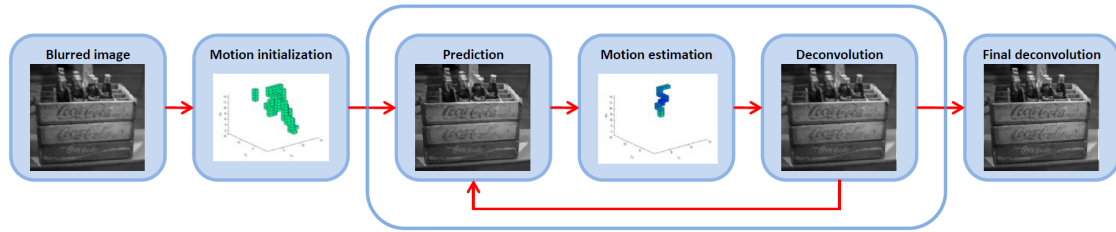


Figure 6.1: Main steps of the proposed algorithm.

6.4.1 Initialization using Backprojection

To solve the ill-posed problem, a good deconvolution method needs to deal with local minimums and multiple feasible solutions (pairs of sharp image and camera motion) for a given blurry image. Recent algorithms employ coarse-to-fine approaches to alleviate these problems, but require computing the kernel basis and estimating pose weights at each scale. In this work, we propose an initialization method of pose weights to address this problem. The proposed initialization method also facilitates the convergence of the optimization process.

The blur kernel at a specific pixel location can be viewed as the projection of the camera motion trajectory or a tomographic scan over the camera motion space. We denote $p(\theta, l)$ as the mapping indicating how camera pose θ affects pixel l in the image, which

can also be viewed as the l -th row of kernel basis K_θ . Thus, we express the blur kernel at site l as

$$k_l = \sum_{\theta} w_{\theta} p(\theta, l). \quad (6.5)$$

Motivated by the backprojection techniques in image processing [20] which are used to reconstruct the two-dimensional signal from its one-dimensional projections, we develop a method to reconstruct the camera motion from multiple blur kernels. We first collect the estimated blur kernels from several image patches (e.g., from 3 by 3 grid) using a fast uniform deblurring algorithm [12]. Although the blur kernels are not precisely estimated due to the camera rotational motion and the estimation errors, they capture the overall shape of the blur kernels (i.e., coarse camera motion) and thus can be used for initialization. From multiple estimated blur kernels of different regions, one approach to approximate the camera motion is to use the inverse transformation by duplicating the two-dimensional blur kernels across the camera motion space. That is, for each entry in k_l , we determine the possible camera poses whose projection $p(\theta, l)$ at this site l is the interest entry and then duplicate the weight of this entry across all the possible camera poses. This procedure is called backprojection commonly used in image reconstruction [20] and we denote $bp(k_l, l)$ as the backprojected value by

$$bp(k_l, l) = \sum_i \sum_{\{\theta | p(\theta, l) = i\}} k_l(i) \Gamma(\theta), \quad (6.6)$$

where i is an index for any location of blur kernel k_l and $\Gamma(\theta)$ is an indicator function of camera poses with value 1 for pose θ and 0 for others. If a pose of interest coincides with the camera motion trajectory (we refer to this pose as an inlier and outlier pose vice versa), its amplitude will be doubled ideally once two blur kernels at different locations are backprojected (which is known in image reconstruction from projections [20]). As the number of the blur kernels increases, the strength of outlier pose decreases with respect to the strength of the inlier pose in which multiple backprojections intersect. In this way, we can approximate the weights W using backprojection $\hat{W} = \frac{1}{n} \sum_l bp(k_l, l)$ with the normalization term n .

The approach with direct backprojection may not render precise camera motion estimation for several reasons. First, a good approximation of camera motion requires a large number of blur kernels to rule out outlier poses. However, it is time consuming to compute a large number of blur kernels at different locations and thus we use only a few blur kernels for initialization. Second, the estimated blur kernel itself is not precise. Since we use a uniform deblurring method to estimate the blur kernels from image patches, the results are not accurate as in-plane camera rotation of camera shake is likely to occur. Within an image, the patches near the optical center are affected with less blur caused by in-plane camera rotation compared to those on the boundary.

To address these problems, we formulate the initialization estimation with an optimization problem and enforce sparse constraints of the weights to get better reconstruction results due to the fact that the camera motion trajectory is sparse in the motion space. Moreover, we assign each backprojection function $bp(k_l, l)$ a confidence value $a(l)$ based on the distance of the site l to the optical center which can usually be assumed to be the center of the image. Thus, the initial estimation of weights W is formulated as,

$$\hat{W} = \arg \min_W ||W - \sum_l a(l)bp(k_l, l)||^2 + ||W||_1, \quad (6.7)$$

where the confidence value needs to be normalized across all the sites, i.e., $a(l) \leftarrow \frac{a(l)}{\sum_m a(m)}$. We show that the proposed initialization method provides a good approximation to the camera motion trajectory in Section 6.5.

6.4.2 Estimating Pose Weights

In the weight estimation step, we compute W given the current recovered latent image L based on Eqn. 6.4.

Image Prediction: To obtain more accurate estimates, we take a prediction step to approximate the latent image similar to [12]. In this step, a bilateral filter is first applied to suppress noise and small edges, and a shock filter is then used to restore strong edges

(more details of the settings can be found in [12].). Each predicted image $\tilde{L}$ has fewer artifacts and sharp edges. More importantly, the predicted image $\tilde{L}$ resembles the latent image even when the estimated L from the previous iteration is blurry, and thus this step facilitates faster convergence.

Weight Estimation on Constrained Pose Subspace: Since image derivatives have been shown to be effective for reducing ringing effects in the optimization process for deblurring [49, 12], we formulate the weight estimation problem as

$$\min_W \sum_{\partial_*} \alpha_* \left\| \sum_{\theta \in S} w_\theta \partial_*(K_\theta \tilde{L}) - \partial_* B \right\|^2 + \Phi_2(W), \quad (6.8)$$

where $\partial^* \in \{\partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$ denotes the partial derivative in two directions of first and second order, and $\alpha_* \in \{\alpha_1, \alpha_2\}$ is the weight for partial derivative of different order as [49].

In this formulation, a direct optimization on the whole pose set S is computationally expensive. We note that the discretized poses along the camera motion trajectory form a connected one-dimensional path in the camera motion space. Moreover, the one-dimensional path constitutes a sparse subset of the space. That is, the majority of poses are outliers (i.e., not on the trajectory) and their weights should be zero. In this work, we propose to optimize Eqn. 6.8 only on a sparse subset of poses referred as an active set $A \subset S$. The active set represents the set of poses which are likely to lie on the motion trajectory.

We note that the construction of initial weights $\hat{W}$ indicates the likelihood a pose lies on the motion trajectory. Thus, the first active set can be obtained by the initial weights with a threshold ε . We use the same threshold strategy to determine the active set at each iteration. Suppose $A^{(i)} = \{\theta_1, \dots, \theta_p, \theta_{p+1}, \dots, \theta_{p+q}\}$ is the active set at iteration i , and the corresponding weights $W_A^{(i)} = \{w_1, \dots, w_{p+q}\}$ are sorted in the decreasing order $w_1 \geq \dots \geq w_p \geq \varepsilon \geq w_{p+1} \geq \dots \geq w_{p+q}$. At iteration $i + 1$, we set the new active set to be $A^{(i+1)} = \{\theta_1, \dots, \theta_p, \hat{\theta}_{p+1}, \dots, \hat{\theta}_{p+q}\}$ by deleting the smallest q weights and adding

new poses $\{\hat{\theta}_{p+1}, \dots, \hat{\theta}_{p+q}\}$ as perturbations. We obtain the new poses by sampling based on the previous active set $A^{(i)}$ using a Gaussian distribution (with small variance), since the poses close to the members of the active set are most likely to be inliers. The perturbation step makes the proposed algorithm more robust.

Once we determine the active set A , we assign the weights for the poses of the inactive set to be 0, and estimate the weights of poses in the active set with ℓ_2 regularization and non-negative constraints,

$$\min_{W_A} \sum_{\partial_*} \alpha_* \left\| \sum_{\theta \in A} w_\theta \partial_*(K_\theta \tilde{L}) - \partial_* B \right\|^2 + \beta \|W_A\|^2, \quad \text{s.t.} \quad W_A \geq 0. \quad (6.9)$$

where 0 is a zero vector and $W_A \geq 0$ denotes each weight in W_A is non-negative. We solve this optimization problem using an interior point method with small β (0.1 in our experiments). Since the active set is already sparse with respect to the whole space, it is not necessary to introduce a sparse regularization term in the above optimization problem.

For parameter settings, we conduct experiments and find that the index number (i.e., the number of elements in a set) of the active set is sensitive when it is smaller than 60 and stable otherwise. The reason is that a very small number of sampled poses may not be enough to cover the motion trajectory so that we could not obtain an accurate weight estimation on such active set. We set the index number to be 60 in this work as it is sufficient to generate good results while the number of samples required to fill the discretized whole motion space is over 1000 even for a small blur kernel size. In [60, 21], since the whole pose space is used for optimization, only coarse-grained sampling is adopted for computational reasons. As we optimize the problem within the active set (which is close to camera motion trajectory), only a small number of poses is required, thereby enabling us to use fine-grained sampling and grids without increasing computational loads. Moreover, it is not necessary to compute all the K_θ as a large number of K_θ may not be used in the optimization process.

6.4.3 Recovering Latent Image with Deconvolution

In the deconvolution step, we estimate the latent image L by fixing the weights W . Since the weights for the poses of the inactive set are set to be 0, we reconstruct the blur matrix with the active set A , $\tilde{\mathcal{K}} = \sum_{\theta \in A} w_{\theta} K_{\theta}$, and the optimization problem becomes

$$\min_L \|\tilde{\mathcal{K}}L - B\|^2 + \Phi_1(L). \quad (6.10)$$

We use the regularization terms in [49] and a similar iterative approach to solve this problem.

6.5 Experimental Results

In this section, we evaluate the proposed algorithm against several state-of-the-art single image deblurring methods for spatially variant blur. The experiments are carried out on a machine with a 3.40 GHz CPU and 16 GB RAM. Our MATLAB implementation takes about 600 seconds to process a blurry image of 441×611 pixels, which is about 1.5 times faster than the MATLAB implementation of [24], the most efficient single image non-uniform deblurring method to the best of our knowledge¹. Our method with the proposed initialization strategy usually takes 5 to 10 iterations to converge with the criteria that the recovered image does not change above a threshold between two iterations. Meanwhile, our method without the initialization strategy takes more than 20 iterations to converge and the results are not satisfying.

Initialization Test using Synthetic Data: We evaluate the proposed initialization method with synthesized data by manually generating three-dimensional camera motion with a connected path in the motion space and initializing their weights. We then apply the synthesized camera motion on a sharp image to obtain a blurry image, based on geometric model in Eqn. 6.2. The proposed backprojection method is evaluated using either

¹We note that the source code of [24] is not available and the time comparison is based on the reported results.

the ground truth or the estimated kernels as shown in Figure 6.2. First, we project the synthesized motion to several image sites and collect the projected blur kernels as the input for backprojection as described in Section 6.4.1. The results in Figure 6.2(c) show that the backprojected results are close to the synthesized motion with respect to poses and weights. This experiment demonstrates the backprojection method models the mapping between blur kernels and camera motion well. We next evaluate the backprojection method using the estimated blur kernels. Since the kernels of nearby locations are similar due to the spatial continuity, we can assume the blur kernels remain the same within a small region and apply a uniform deblurring algorithm to estimate them. From the blurry image, we select a few image patches from which the blur kernels are estimated using a fast uniform deblurring algorithm [12]. As shown in Figure 6.2(d), the estimated poses cover the synthesized motion (i.e., the blocks in (d) cover all blocks in (b)) which indicates that the selected poses are good enough as the first active set for weight estimation. We note the backprojected poses are sparse with respect to the whole sampled pose space (around 100 poses compared with 3375 discretized poses in this case).

Non-uniform Deblurring: In Figure 6.3, we compare with state-of-the-art single image deblurring algorithms for spatially variant blur [60, 21, 22, 24]. Since not all the source codes are available, we use the reported results for comparisons. Among these methods, three algorithms [60, 21, 24] adopt similar geometric models for camera shake which enforce global constraints for blur kernels while one method [22] estimates kernels locally. In [60], three-dimensional rotations are considered for modeling camera motion while the other methods [21, 24] and our approach focus on estimation of x, y translation and in-plane rotation. We compare mainly with [24] since it achieves the state-of-the-art deblurring results in terms of speed and quality.

For the *Pantheon* image, both the proposed method and [24] recover sharp images without obvious blur while the results from [60] contain ringing effect along the roof. As only three-dimensional rotations are considered in [60], camera translation is not fully removed in this case. In the *Gas station* image, the results from our algorithm contain fewer

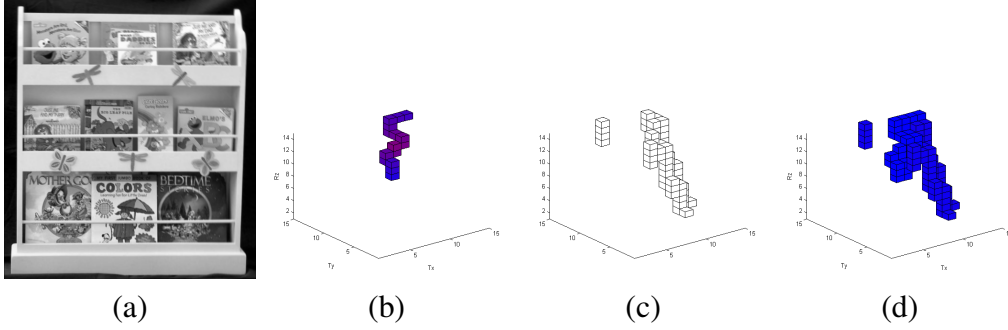


Figure 6.2: Initialization using backprojection with synthetic data. Light green cubes indicate small weight values and dark blue ones indicate large ones. (a) synthesized blurry image. (b) its corresponding camera motion. (c) initial camera motion using backprojection with ground truth blur kernels at several image sites. (d) initial camera motion using backprojection with blur kernels estimated by a uniform deblurring method [12].

artifacts and sharper edges (e.g., letters next to the gas pump). The results show that camera motion for this image is better estimated by our approach. In the *Butcher shop* image, most blur effect can be removed by [22, 24], although some details are smoothed when compared with the results generated by the proposed method. We note that an additional step is used in [24] to reduce artifacts using corrective weighting after the latent images are recovered, while our approach does not have any post-processing. However, as shown in the inset images of *Books* and *Butcher shop*, the results from our algorithm recover more details in general. Compared to the results generated by state-of-the-art algorithms, our approach not only removes non-uniform blur efficiently but also generates comparable or better deblurred images.

The main limitation of our approach is that it may fail or require numerous iterations to converge if the blurs are too large (e.g., z rotation of 10 degrees), which is also the case for other non-uniform deblurring algorithms [21, 24]. The reason is that the blurs resulting from large rotational motion cannot be well estimated from image patches using any uniform deblurring method. Therefore the pose weights are not initialized well, and



Figure 6.3: Comparison with state-of-the-art single image deblurring methods for spatially variant blur. Top to bottom: *Pantheon*, *Gas station*, *Books* and *Butcher shop*.

this likely results in local minimum.

6.6 Conclusion

In this chapter we present a fast single image deblurring approach to remove non-uniform blur. We propose an initialization method of camera motion using backprojection to facilitate the convergence and avoid local minimums. We develop a method to solve pose weights and the latent image iteratively. To estimate the pose weights, we restrict the solution space in a small subset for efficiency and add perturbation to make the system robust. Experimental validation with state-of-the-art methods demonstrate the efficiency and effectiveness of our approach. Our future work will focus on developing a public code library of state-of-the-art algorithms with a benchmark dataset and sound metrics for quantitative evaluations.

Chapter 7

Joint Depth Estimation and Camera Shake Removal from Single Blurry Image

7.1 Introduction

The motion deblurring problem has drawn much attention in recent years with demonstrated success. Most approaches model a blurry image as the integration of intermediate frames captured by the camera along the motion trajectory, typically assuming in-plane translational camera motions. Recent endeavors in nonuniform deblurring further explore a rotational camera motion model [13, 60, 57] and broaden the application range of deblurring approaches. These methods, however, do not consider large scene depth variation, which is ubiquitous in real-world scenes.

Figure 7.1 shows one outdoor image taken by a hand-held camera. Almost all existing deblurring methods do not perform well in this seemingly simple example due to the non-ignorable depth variation. While some algorithms on depth-aware motion deblur have been recently proposed [64, 45], they nevertheless leverage additional observations for

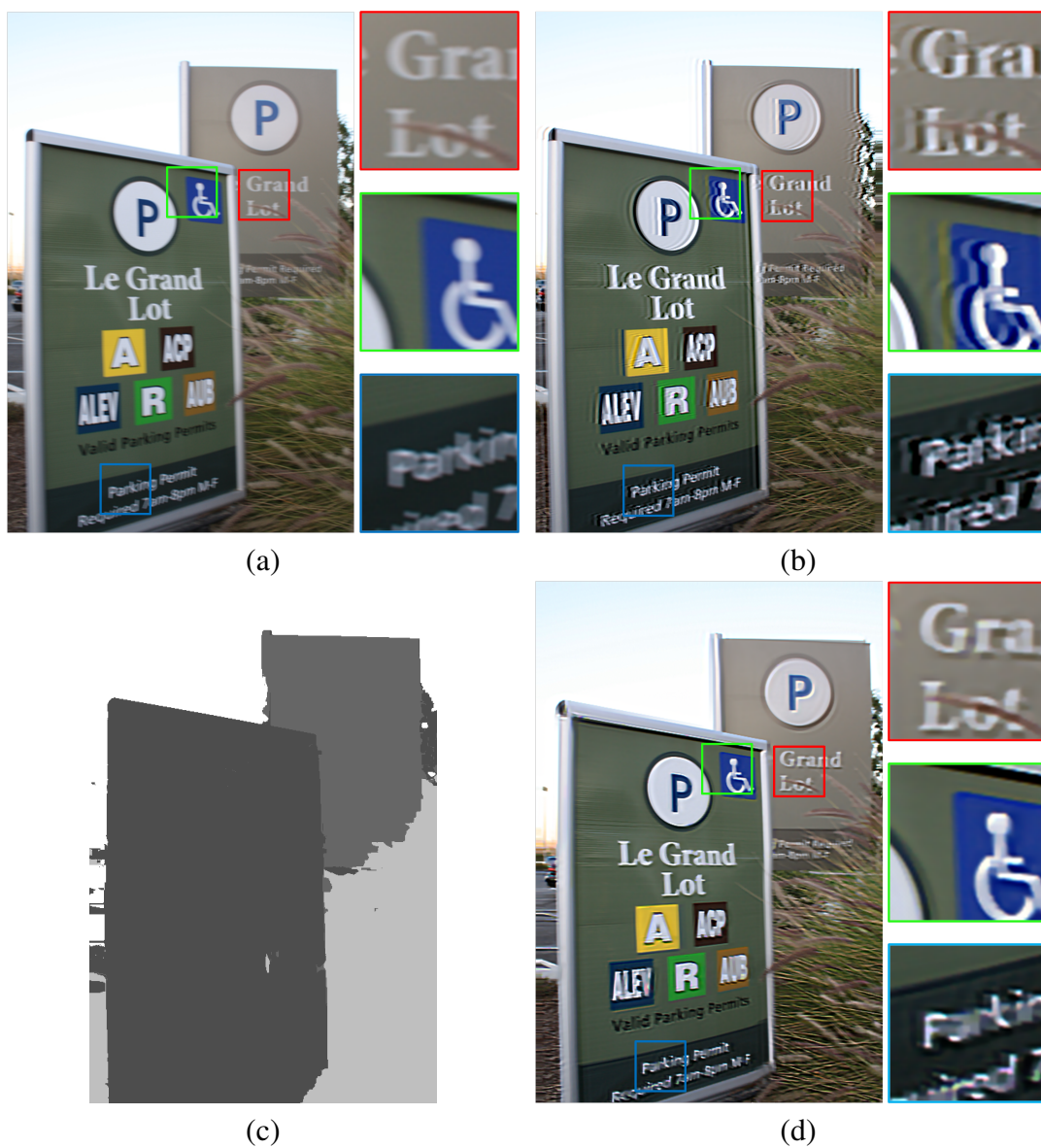


Figure 7.1: Depth-aware non-uniform deblurring. (a) blurry image; (b) restored image by the method [24] assuming constant scene depth; (c) estimated depth map by our method; (d) restored image by our method using the estimated depth map.

this challenging task. The inherent limitation of the existing deblurring methods to handle depth variation is readily explained. On the one hand, even with known scene depth, the solution space of the deblurring problem with depth consideration is typically very large, making it not suitable for practical applications. On the other hand, depth information is not usually available and depth inference from a single image is extremely difficult.

In this chapter, we propose a unified framework for the joint restoration of scene depth and the latent clear image. While depth information is critical in blur removal, the blur image provides an additional cue for depth information. In addition, rather than inferring depth for isolated pixels [39], we extract depth layers and model the layer-wise blur respectively. This makes the problem better constrained and is the major difference to the previous work. We develop an expectation-maximization (EM) scheme to solve scene depth and the latent image alternately by exploiting their underlying geometric relationships, from a single blurry image. We demonstrate that the proposed model effectively removes non-uniform blur from images with depth variation.

7.2 Related Work

The problem of removing spatially invariant blur has been studied extensively and numerous algorithms have been proposed. Since blind deconvolution is an ill-posed problem, prior knowledge or additional information are often required for effective solutions. In the image deblurring literature, two types of additional information are often used: natural image priors [18, 38, 49, 40, 12, 15] and additional image observations [3, 46, 69, 56, 42].

Recently, the problem of removing spatially variant blur caused by camera motion has attracted much attention due to its wide range of applications [2, 13, 50, 56, 31, 60, 21, 24, 57, 59, 26, 29]. Early work splits an image into several regions where each is modeled with a constant PSF and recovered by a uniform deblur algorithm [2, 13]. However, these methods are sensitive to region size which is required to be sufficiently small for the assumption of uniform PSFs to hold. Hybrid camera systems have also been used to

capture a local blur kernel at each pixel location based on optical flow [56, 31]. The methods with additional inputs are demonstrated to be effective on image deblurring, but such approaches are not applicable to single image non-uniform deblurring.

In [60, 57], geometric models are proposed to model the observed blurry image as the integration of all the intermediate images captured by the camera along the motion trajectory. The intermediate images are modeled as the homographic transformations of the latent image. Based on this model, the spatially-varying deblurring problem can be solved by estimating a general camera motion function and then the latent image. A similar model has been used to model three degrees of camera motion [21] (with in-plane translations and rotation) instead of camera rotations in [60]. To reduce the computational load, fast non-uniform deblur methods have been developed to speed up the optimization step [24, 26]. In [33], a dataset for motion deblurring is constructed by recording and playing back the camera shake with a robot system. The blurry images are obtained by taking photos of sharp pictures which assume constant depth. Notwithstanding the demonstrated success, the above-mentioned non-uniform deblurring methods do not consider the effect of scene depth on camera motion blur or they assume constant scene depth for simplicity.

To exploit depth information from the blurry input, recent methods [13, 50, 8] partition an image into foreground/background and remove the blur caused by object motion. A coded aperture method has been proposed for deblurring and the decoded images are used to infer pixel-wise depth in a Markov random field [39]. Depth effects on translational blur have also been discussed in [52] where the blur is estimated from the user specified region with approximately constant depth to infer depth. In [64], two blurry images of the same scene are used to infer depth with stereo techniques and further improve the blur removal. Similarly, a non-uniform motion deblurring method has been proposed to exploit bilayer-scene constraints from two blurry observations [45]. In this work, we consider single-image non-uniform blur caused by camera motion and scene depth variation. Depth estimation from one single image itself is a difficult problem. A depth estimation method has been proposed based on the visual depth cues, texture and color, in [48]. However,

the problem becomes more challenging when a blurry input is given as the depth cues are destroyed in the presence of motion blur. We address this issue by exploiting the underlying geometric properties of a blurry image.

7.3 Depth-Aware Deblurring

In this section, we present the depth-aware geometric model of non-uniform blur caused by camera motion, and propose an EM formulation for joint depth estimation and latent image recovery.

The commonly used geometric model of camera motion considers an observed blurry image y as the weighted sum of the transformed results based on the sharp image x [60, 21, 57],

$$y = \sum_{j=1}^t w_j (K(\theta_j)x) + n, \quad (7.1)$$

where $\{\theta_j\}_{j=1}^t$ denotes the sampled camera poses, the weights $W = (w_1, \dots, w_t)^\top$ indicate the exposure time the camera stays at each pose, $K(\theta_j)$ is the matrix that warps the latent image x to the transformed result at a sampled pose θ_j and n as the observation noise. In this formulation, $K(\theta_j)$ can also be viewed as a blur kernel basis of pose θ_j where the r -th row is the weight vector representing how the r -th pixel of the latent image x contributes to the blurry image. In [21], the matrix $K(\theta_j)$ is induced from the homography P_j of the latent image assuming constant depth d ,

$$P_j = C(R_j + \frac{1}{d}T_j[0, 0, 1])C^{-1}, \quad (7.2)$$

where R_j and T_j denote the rotational and translational components of pose θ_j , and C is the intrinsic matrix of the camera. This equation also shows that depth variation affects only translational blur. While this model assumes fronto-parallel scenes, we consider more generic cases with varying depth in this work. Considering depth variation, the r -th row of the warping matrix varies with the depth at the r -th pixel. Thus, we denote $K(\theta_j, D)$ as the matrix induced by pose θ_j and depth map D .

For practical applications, recent work simplifies the camera motion from the six-dimensional space to a three-dimensional subspace. In [60], only three rotations are used to approximate the camera motion, while in-plane translations and rotation are considered in [21, 24, 26]. It has been shown that these two models achieve similar performance under some assumptions [33]. In this work, we consider in-plane translations and rotation of camera motion, since out-of-plane rotations can be approximated with in-plane translations at a certain depth in the proposed depth-aware model.

To make the problem tractable, we first split the image into disjoint regions, $x = \sum_{i=1}^N x_i$ for the latent image and $y = \sum_{i=1}^N y_i$ for the blurry image, and assign a constant depth d_i to each region x_i . Here we use x_i to indicate the i -th region, with the intensity of a pixel the same as the one in the original image x if it belongs to i -th region and 0 otherwise. We rewrite (7.1) as

$$y = \sum_{i=1}^N y_i = \sum_{i=1}^N \sum_{j=1}^t w_j (K(\theta_j, d_i) x_i) + n. \quad (7.3)$$

Given only the blurry image, there are numerous unknown variables in (7.3), which makes it difficult to solve.

7.3.1 EM Formulation

Since there are numerous unknown variables in (7.3) and the equation cannot be solved directly, we propose an EM formulation to solve the problem. Given the observed image y , the goal is to estimate the latent image and the weights of camera motion $\gamma = (x, W)$ in terms of the unknown depth map D . The maximum likelihood estimate (MLE) of this problem is determined by the marginal likelihood of the observed image,

$$L(\gamma; y) = p(y|\gamma) = \sum_D p(y, D|\gamma). \quad (7.4)$$

It can be solved with the EM algorithm by iteratively applying the following steps:

E-step: Compute the expected value of the log likelihood function under the current estimate of $\gamma^{(l)}$,

$$Q(\gamma|\gamma^{(l)}) = E_{p(D|y, \gamma^{(l)})}[\log p(y, D|\gamma)], \quad (7.5)$$

M-step: Determine $\gamma^{(l+1)} = \arg \max_{\gamma} Q(\gamma|\gamma^{(l)})$.

We explain these steps in the following sections.

E-step

To express $Q(\gamma|\gamma^{(l)})$, we first consider the likelihood

$$p(y|x, W, D) = \prod_{i=1}^N p(y_i|x_i, W, d_i), \quad (7.6)$$

in terms of disjoint regions. We assume a discrete sampled set of the potential depth values, $\{v_k\}_{k=1}^s$, and introduce an auxiliary variable $\alpha_{i,k}$ indicating the value of depth d_i as

$$\alpha_{i,k} = \begin{cases} 1, & \text{if } d_i = v_k, \\ 0, & \text{otherwise,} \end{cases} \quad (7.7)$$

and $\sum_{k=1}^s \alpha_{i,k} = 1$ since only one depth value is selected for each region. The likelihood $p(y_i|x_i, W, d_i)$ for each region is then expressed as the data fitting term and the auxiliary variable,

$$\begin{aligned} & p(y_i|x_i, W, d_i) \\ &= \exp\left(-\frac{1}{2\eta^2} \left\| y_i - \sum_{j=1}^t w_j K(\theta_j, d_i) x_i \right\|^2\right) \\ &= \exp\left(-\frac{1}{2\eta^2} \sum_{k=1}^s \alpha_{i,k} \left\| y_i - \sum_{j=1}^t w_j K(\theta_j, v_k) x_i \right\|^2\right), \end{aligned} \quad (7.8)$$

where we assume a zero mean Gaussian prior for signal noise $n \sim N(0, \eta^2)$. Here the warping matrix $K(\theta_j, v_k)$ can be pre-computed once the sampling of camera poses and depth values is determined.

Assuming uniform distribution on $p(D)$, we write the log-likelihood $\log p(y, D|\gamma)$ using (7.6) and (7.8),

$$\begin{aligned} \log p(y, D|x, W) &= \log p(y|x, W, D)p(D|x, W) \\ &\propto -\frac{1}{2\eta^2} \sum_{i=1}^N \sum_{k=1}^s \alpha_{i,k} \|y_i - \sum_{j=1}^t w_j K(\theta_j, v_k) x_i\|^2. \end{aligned} \quad (7.9)$$

and (7.5) can be written as

$$\begin{aligned} &E_{p(D|y, (x, W)^{(l)})}[\log p(y, D|x, W)] \\ &\propto -\frac{1}{2\eta^2} \sum_{i=1}^N \sum_{k=1}^s E[\alpha_{i,k}^{(l)}] (\|y_i - \sum_{j=1}^t w_j K(\theta_j, v_k) x_i\|^2). \end{aligned} \quad (7.10)$$

Here $E[\alpha_{i,k}^{(l)}]$ denotes the expectation of $\alpha_{i,k}$ in the l -th iteration and it can be computed with the pseudo-likelihood

$$\begin{aligned} \bar{\alpha}_{i,k}^{(l)} &= E[\alpha_{i,k}^{(l)}] = p(\alpha_{i,k}^{(l)} = 1 | y_i, x_i^{(l)}, W^{(l)}) \\ &= \frac{\exp(-\frac{1}{2\eta^2} \|y_i - \sum_{j=1}^t w_j^{(l)} K(\theta_j, v_k) x_i^{(l)}\|^2)}{\sum_{k'=1}^s \exp(-\frac{1}{2\eta^2} \|y_i - \sum_{j=1}^t w_j^{(l)} K(\theta_j, v_{k'}) x_i^{(l)}\|^2)}, \end{aligned} \quad (7.11)$$

which also satisfies $\sum_{k=1}^s \bar{\alpha}_{i,k}^{(l)} = 1$.

M-step

Once all the $\bar{\alpha}_{i,k}^{(l)}$ are determined, we solve γ in the M-step with the optimization problem,

$$\gamma^{(l+1)} = \arg \max_{(x, W)} \sum_{i=1}^N \sum_{k=1}^s \bar{\alpha}_{i,k}^{(l)} (\|y_i - \sum_{j=1}^t w_j K(\theta_j, v_k) x_i\|^2). \quad (7.12)$$

This optimization problem is similar to the existing non-uniform methods based on the geometric model, except that the image is partitioned into multiple depth layers. We solve x and W alternatively in a way similar to the other non-uniform deblurring algorithms [60, 21, 24, 26] and explain the detailed steps in Section 7.3.4.

7.3.2 Image Partition

In this work, each blurry image is partitioned into multiple depth layers. There are two main issues for the blurry-image partition as shown in Figure 7.2. One is the ringing artifacts along edges which makes it difficult to segment blurry image. Another problem is the ambiguity of the region boundary, where is likely to be caused by the combination of objects appearing in different layers. To address these issues, we first process a blurry image with a smoothing filter (e.g., Gaussian) to suppress the artifacts. We then apply a matting algorithm [10] on the filtered image to obtain multi-layer segmentation results, with the scribble for each layer from the user. The pixel with matting value less than one is likely to lie on the region boundary and we assign a weight based on the matting value for the data fitting term. We present an example in Figure 7.2 to show how a blurry image is partitioned into multiple layers.

7.3.3 Depth Estimation

In Section 7.3.1, we discussed how to compute the expectation of $\alpha_{i,k}$ and use it in the M-step. Considering the adverse effect of the small edges, we apply the L_0 smoothing filter [65] to the estimated latent image x before computing $\bar{\alpha}_{i,k}$. To determine the depth value of each layer, we discard small values of $\bar{\alpha}_{i,k}$ and only keep the highest one $\hat{k}_i = \arg \max_{k=1}^s \bar{\alpha}_{i,k}$. The depth value corresponding to the highest value $\bar{\alpha}_{i,k}$ is assigned to the region x_i and y_i , $d_i = v_{\hat{k}_i}$. One can also determine the depth by combining several top $\bar{\alpha}_{i,k}$ values.

7.3.4 Constant-Depth Non-Uniform Deblurring

In this section, we explain how to solve for x and W by fixing the depth D (we abuse the notation to indicate $\{\bar{\alpha}_{i,k}\}_{i,k}$ in (7.12)). We adopt the same formulation as existing non-uniform deblurring methods in an alternating way, i.e., solving one by fixing the others.

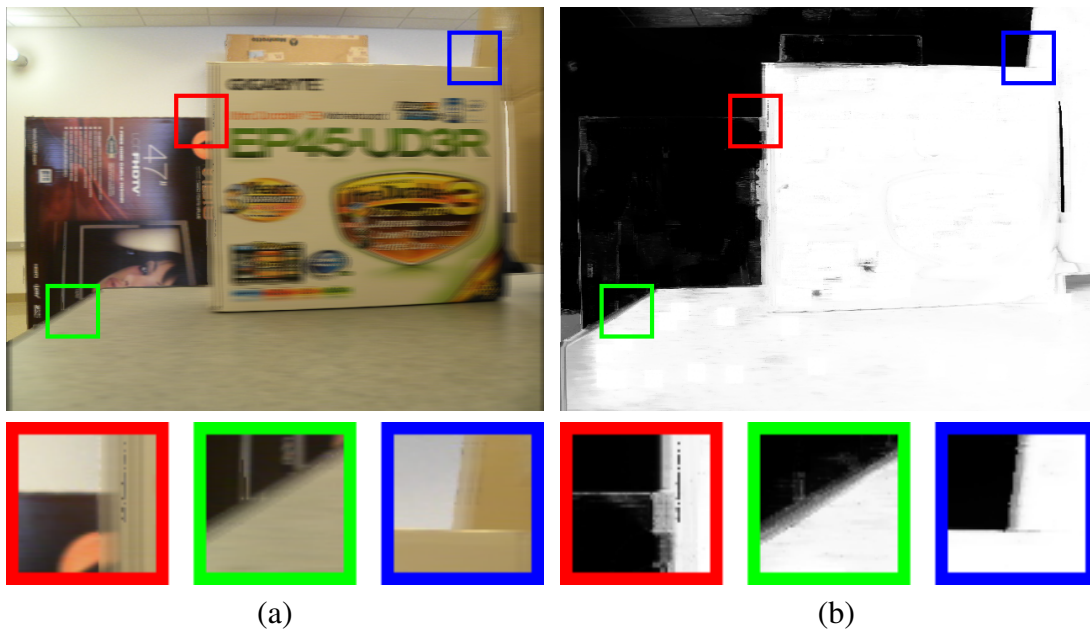


Figure 7.2: Image partition using matting. (a) blurry image and patches that are difficult for image partitioning, (b) one layer using matting and the corresponding matting patches.

At the first iteration, we initialize the weights W using backprojection as [26] to facilitate the convergence.

Image Prediction: Before the alternating optimization, we predict the approximate latent image using a bilateral filter and a shock filter similar to [12]. The bilateral filter is first applied to reduce noise and remove small edges, and then the shock filter is used to enhance strong edges. By doing so, the predicted image $\tilde{x} = \sum_{i=1}^N \tilde{x}_i$ contains less artifacts and sharp edges, which leads to more accurate estimate of pose weights W as the predicted image resembles the latent image.

Weight Estimation: Image derivatives have been shown to be effective for suppressing ringing artifacts in deblurring problems [49, 12]. Thus, we replace the image intensity with derivatives in the data fitting term and express the weight estimation as

$$\begin{aligned}\bar{W} &= \arg \max_W \sum_{i=1}^N \sum_{k=1}^s \bar{\alpha}_{i,k} (||\partial y_i - \sum_{j=1}^t w_j K(\theta_j, v_k) \partial \tilde{x}_i||^2) \\ &= \arg \max_W \sum_{i=1}^N (||\partial y_i - \sum_{j=1}^t w_j K(\theta_j, v_{\hat{k}_i}) \partial \tilde{x}_i||^2).\end{aligned}\tag{7.13}$$

Since W represents the weights of camera motion trajectory, we add a smoothness term to enforce continuity and write (7.13) in a matrix form,

$$\bar{W} = \arg \max_W ||AW - b||^2 + \lambda ||\Gamma W||^2,\tag{7.14}$$

where A is the matrix with $K(\theta_j, v_{\hat{k}_i}) \partial \tilde{x}_i$ at index (i, j) and $b = (\partial y_1, \dots, \partial y_N)^\top$. Here we denote Γ as the matrix form of the partial operator ∂ . To address computational issues, we use a similar formulation as [26] by solving it with a subset of the camera pose space. The subset is determined by perturbing the one from the last iteration. We use conjugate gradient to minimize (7.14) and normalize W with unit sum.

Latent Image Estimation: To recover the latent image, we solve the following optimization problem,

$$\begin{aligned}\bar{x} &= \arg \max_x \sum_{i=1}^N \sum_{k=1}^s \bar{\alpha}_{i,k} (||\partial y_i - \sum_{j=1}^t w_j K(\theta_j, v_k) \partial x_i||^2) \\ &= \arg \max_x \sum_{i=1}^N (||\partial y_i - \sum_{j=1}^t w_j K(\theta_j, v_{\hat{k}_i}) \partial x_i||^2).\end{aligned}\tag{7.15}$$

Since $\{x_i\}_{i=1}^N$ are disjoint regions, we solve x_i independently with constraints,

$$\bar{x}_i = \arg \max_{x_i} ||\partial y_i - \sum_{j=1}^t w_j K(\theta_j, v_{\hat{k}_i}) \partial x_i||^2 + \mu ||\partial x_i||,\tag{7.16}$$

where we use a simple regularization to favor smooth gradients. For fast optimization, we adopt the strategy of splitting the image into grids and performing uniform deconvolution on each region, similar to [24]. It is also possible to use other constraints or deconvolution schemes [39, 49, 34, 35] to solve (7.15). For the final deconvolution, we use the method with sparse constraints [34] to recover the image.

7.3.5 Initialization and Implementation Details

In this work, we set the parameters $\lambda = 0.5$ and $\mu = 0.1$ in all experiments.

In the beginning, we initialize the depth of each layer to be the same value. Thus, the first iteration can be viewed as the non-uniform deblurring without depth variation. After applying the L_0 smoothing filter [65], the estimated latent image results in a good depth estimation. Figure 7.3 shows synthetic example of the depth estimation in the first iteration. The proposed method provides a decent depth estimation except for the sky region, which does not have much texture and therefore is less important for image deblurring. With proper depth estimation in the first iteration, the proposed method converges to a solution efficiently.

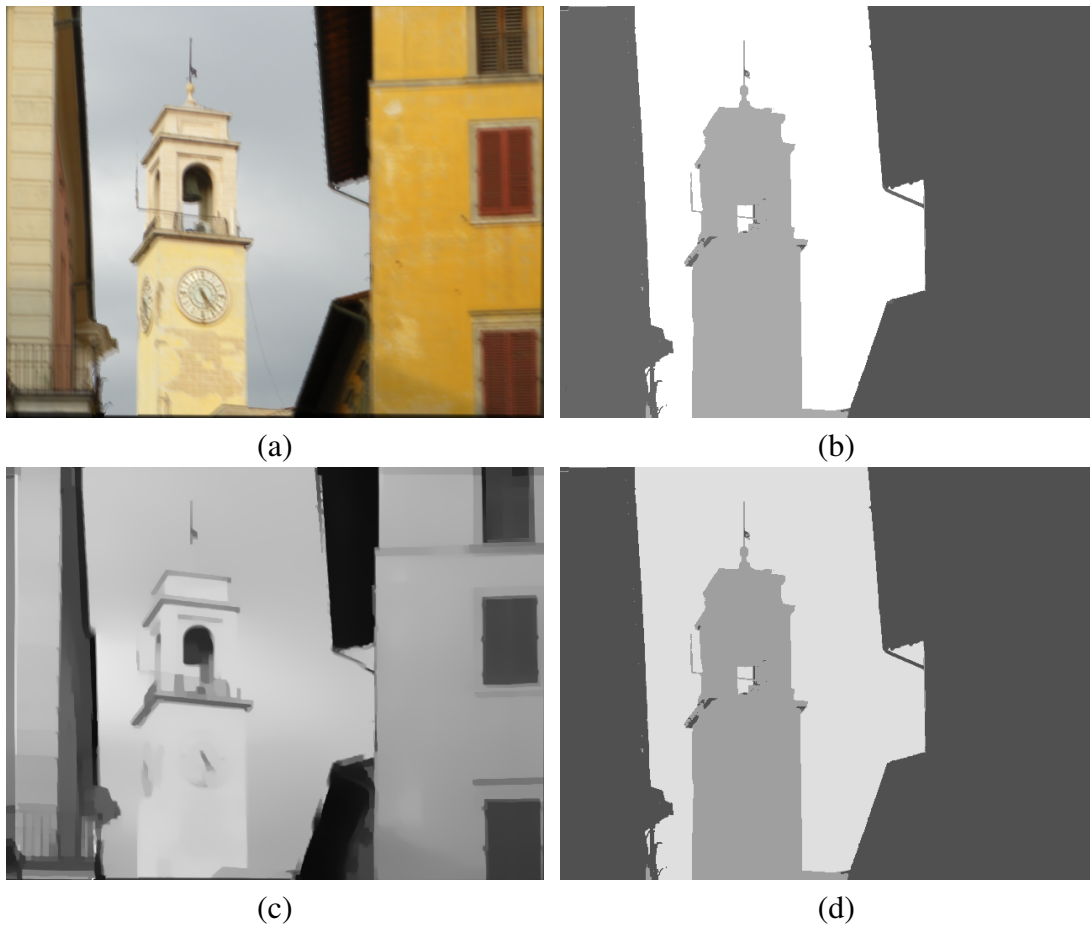


Figure 7.3: Synthetic example of the depth estimation in the first iteration. Dark region indicates small depth. (a) synthetic blurry image; (b) ground truth depth; (c) estimated latent image after smoothing filter; (d) our estimated depth using (c).

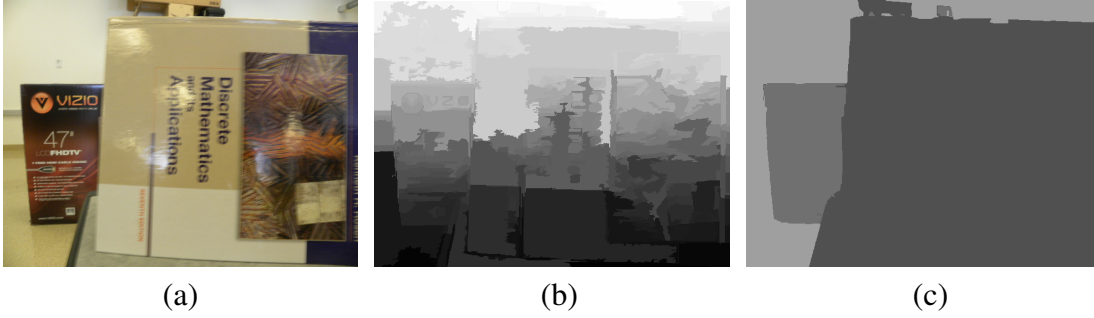


Figure 7.4: Comparison with depth-from-single-image method [48] on a real image. (a) blurry image; (b) estimated depth by [48]; (c) our estimated depth.

7.4 Experimental Results

In this section, we first evaluate depth estimation results of the proposed algorithm and then compare deblurring performance against several state-of-the-art methods for spatially variant blur. Implemented in MATLAB, it takes about 2.5 hours to process a blurry image of 480×640 pixels on a 3.40 GHz machine with 16 GB RAM.

7.4.1 Depth Estimation

Depth estimation from a single image is a difficult task. The recent approach [48] proposes an MRF-based method to infer scene depth from visual cues, e.g., texture and color. In Figure 7.4, we present an example of a natural blurry image and the depth estimation by [48] and our approach. Although our method does not capture subtle depth variation, it estimates the rough depth well based on the geometry information contained in the blurry image. In contrast, the method [48] is not effective in estimating depth from blurry images where region texture and edge contrast are missing as a result of motion blur.

We evaluate our depth estimation algorithm on synthetic images with spatially variant blurs. First, we collect a set of sharp images and their depth maps (i.e., indoor images with Kinect sensors and outdoor scenes with laser scanners). We partition the image into

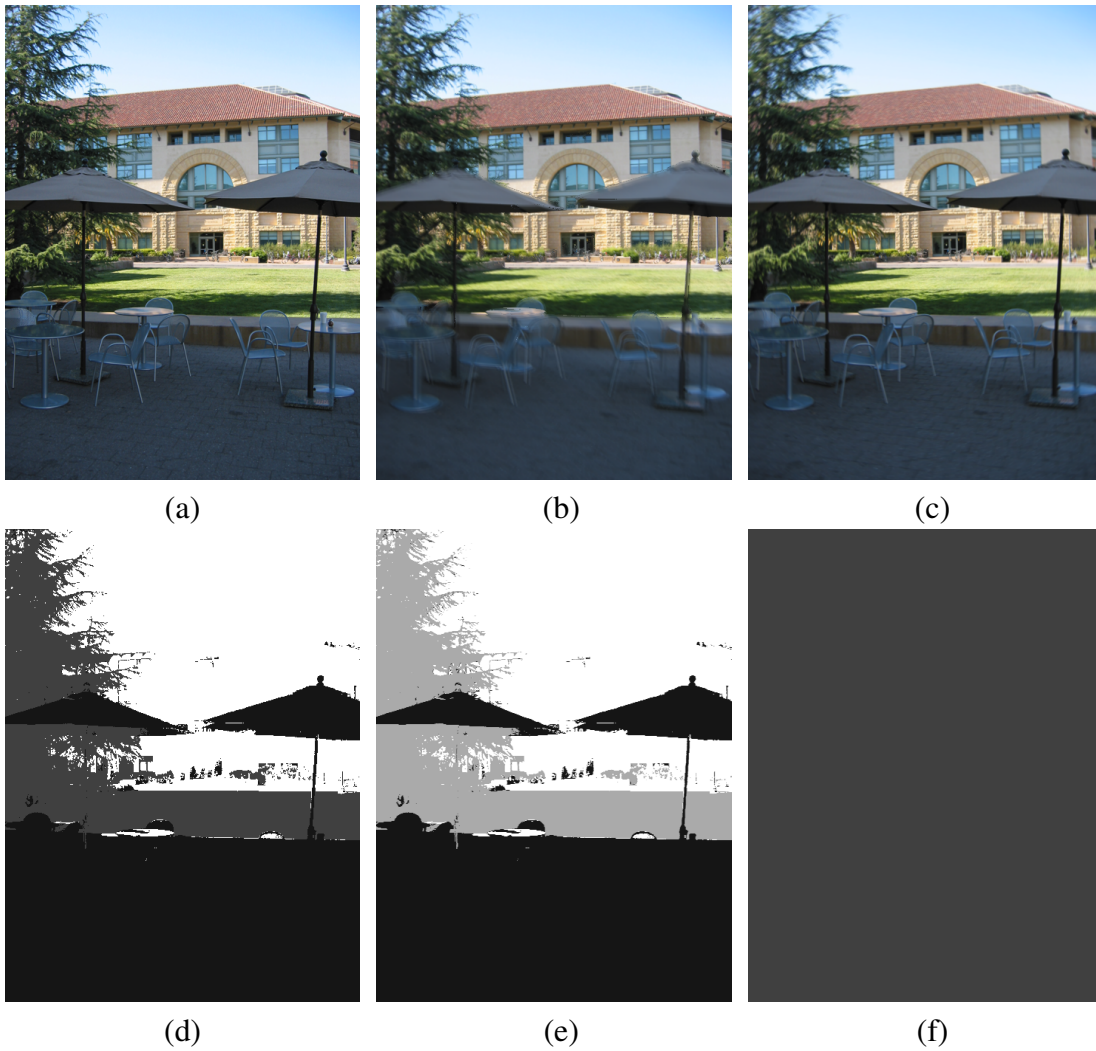


Figure 7.5: Depth estimation results on synthetic images. (a) sharp image; (b) synthesized blurry image caused by both translational and rotational blur; (c) synthesized blurry image caused by only rotational blur; (d) ground-truth depth; (e) our estimated depth from (b); (f) our estimated depth from (c).

Table 7.1: Average error with respect to the minimal number of translated pixels in the camera motion.

# of pixels	1	3	5	7	9	11
error (meter)	2.02	1.67	1.47	1.35	1.10	1.03

different layers based on the depth map and assign the average depth to each layer. Synthetic camera motion trajectories and corresponding weights are then applied to generate blurry images. For experiments on depth estimation, 420 pairs of blurry and sharp images generated from 6 sharp images (3 indoor and 3 outdoor images from [48]), and 70 randomly synthesized camera motions are collected. Each sharp image has the depth variation from 2 meters to 20 meters with at least three depth layers. The average error per pixel is 2 meters and most errors are from the cases with small camera translations. As demonstrated in Table 7.1, our depth estimation becomes more accurate when there exists more translational camera motion. Figure 7.5 shows two depth estimation results by the proposed algorithm, from one blurry image caused by both translational and rotational camera motion and another caused only by rotational blur. With only rotational blur, our method estimates a uniform depth value for each layer as shown in Figure 7.5(f). The results, although not accurate with respect to depth estimation, satisfy our model that depth variation does not affect rotational motion blur as mentioned in Section 7.3. This further demonstrates that our model is able to infer the geometric information contained in motion blur.

7.4.2 Non-Uniform Deblurring

In this section, we compare our methods with several state-of-the-art single image deblurring methods [21, 24, 64]. The experiments on the synthetic blurry images and real blurry images are carried out to validate our model and EM algorithm. We also show that our method can be used to remove non-uniform blur caused by camera motion for a scene

with small depth variation. Due to space limitations, only a few results are presented in this section and more results can be found in the supplementary material. We note that the dataset [33] is constructed by taking blurry photos on sharp pictures assuming constant depth, and thus it is not suitable for evaluating our depth-aware method.

Synthetic Blurry Images: We evaluate deblurring algorithms on the synthetic images obtained using the same process as described in Section 7.4.1. As shown in Figure 7.6(a), there are more than two objects at different depths in the synthetic image, and the objects in the back are less blurry than those in the front. The state-of-the-art non-uniform deblurring method based on constant depth [24] does not work well (Figure 7.6(b)) since the data-fitting terms of different objects penalize each other in the weight estimation of the camera motion. However, this problem can be alleviated by considering depth factor as formulated in this work. In Figure 7.6(c) and (d), we demonstrate that the proposed EM formulation generates reasonable depth estimation and improves the deblurred results. We select the text regions to better visualize the difference.

If the image contains small depth variation, our method naturally falls back to constant-depth non-uniform deblurring.

Real Blurry Images: In these experiments, we evaluate the proposed method on the real-world images blurred by hand-shake. Figure 7.7 shows some results with comparisons to the state-of-the-art methods for spatially variant blurs [24, 64, 26]. In the *Books* example, several books and a cookie box appear in the scene with small depth variation. In the *Boxes* example, there are a dozen objects which appear in a cluttered scene with large depth variation. The uniform algorithm in [12] and the method in [24] assume constant depth and do not perform well in this case, while the proposed method recovers sharper images. Our method is not able to detect subtle depth change, and instead infers mainly two different depth layers in this image. However, by separating the image into two layers, the proposed method obtains more accurate weight estimation of the camera motion compared to the one using constant depth, which also demonstrates the effectiveness of the depth-aware model.

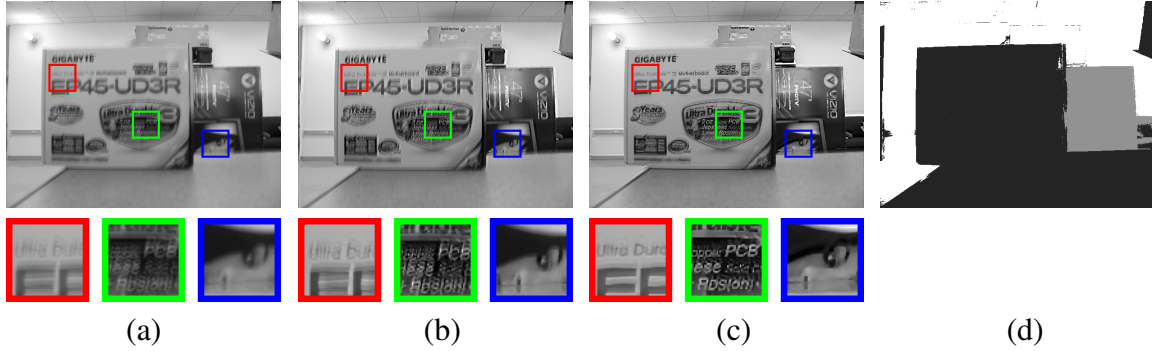


Figure 7.6: Depth-aware non-uniform deblurring on synthetic images. (a) synthetic blurry image; (b) deblurred result of [24]; (c) recovered image using our method; (d) our estimated depth (best viewed on a high-resolution display).

We note that the proposed method on real images does not estimate depth as accurate as in the cases of synthetic images. One reason is that there are many unknown variables to be estimated and the error surface is complex with alternating optimization in the M-step. Thus it is difficult to guarantee convergence to a good local minimum. Some other reasons include the simplified camera model assumption. Since our model is based on the pinhole model, lens error and defocus blur due to depth variation may affect the accuracy of depth estimation for real blurry images.

7.5 Conclusion

In this chapter, we discuss the depth influence of depth on the geometric model of non-uniform blur caused by camera motion. We present a unified layer-based model for depth-aware deblurring and propose an EM formulation to recover depth and the latent image jointly given only a blurry image. We demonstrate the effectiveness of our depth estimation scheme when there exists certain translations, and the estimated depth information helps improve the performance of latent image reconstruction over the state-of-the-art

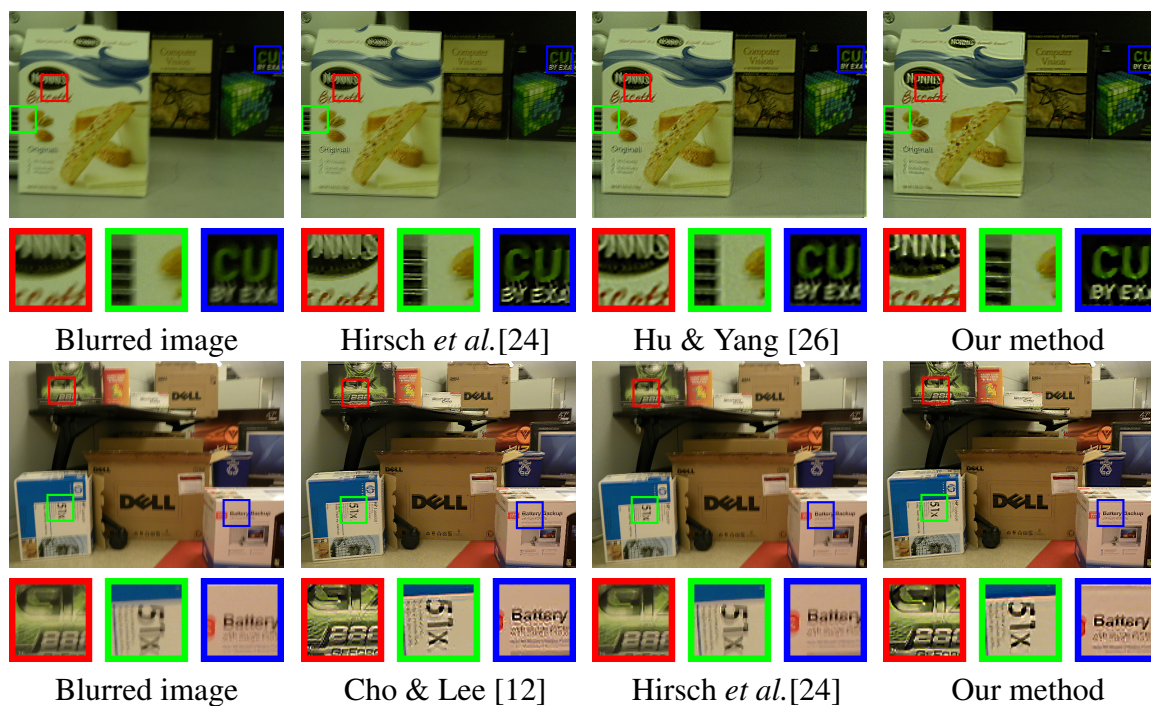


Figure 7.7: Depth-aware non-uniform deblurring on real images. Top to bottom: *Books*, and *Boxes* examples (best viewed on high-resolution displays).

models that assume constant depth. When the depth variation of the scene is limited, the proposed algorithm performs similarly to the non-uniform deblur algorithms that assume constant depth.

Chapter 8

Conclusion and Future Work

This chapter concludes the thesis and presents possible future research topics in the future.

8.1 Summary

This thesis investigates new algorithms to remove motion blur from single input image. To address the ill-posed problem, additional information, such as image priors and additional observations, is required to better constrain the solution. The problem becomes more difficult if there is camera rotation involving, which result in spatially varying blur. To explore generic image priors, Chapter 3 introduces sparse constraints on the image representation for deblurring. The same coefficients for both the latent image and blurry image are enforced, with respect to the corresponding sharp and blurry dictionaries. Sparsity has been shown to be successful as a generic prior for natural images, however, the information contained in image content has not been well explored. To address this concern, we propose a technique to exploit the motion trajectory information in each blurry image in Chapter 4. The light streaks, which depict the blurred light sources and blur trajectory, are extracted automatically by using the spectrum property between the blurry

image and the blur kernel. With the help of the detected light streaks, strong constraints for each image are enforced to estimate the blur kernel. We show that the proposed technique can be used to address deblurring problems on challenging scenarios of low-light images. We also explore the influence of image structure on deblurring algorithms in Chapter 5. Towards this, we propose a learning-based approach for selecting good features and good patches for blur kernel estimation. The results show that the proposed algorithm selects good regions automatically for deblurring with efficient and favorable results, thereby relieving users from tedious trials for selecting image patches.

The uniform blur model is sufficient if the camera shake is caused only by in-plane translation without any out-of-plane rotations, but this does not hold in general. Recent methods adopt a geometric model to formulate the non-uniform blur problem, but suffer from large computational load due to dense sampling of camera poses. Chapter 6 presents a fast image deblurring approach to address this issue. The proposed initialization method of camera motion employs backprojection to facilitate the convergence and avoid local minima. Instead of estimating all the pose weights, we restrict the solution space in a small subset for efficiency. In addition to efficient algorithms for spatially varying blur, we also explore a general model for this problem in Chapter 7. The usually ignored depth influence on the geometric model was discussed and a unified lay-based model was proposed for depth-involved deblurring problem. We propose an EM formulation to estimate the scene depth as well as the latent image jointly given only one input. The results demonstrate that solving depth from a single blurry input is possible and that it helps to perform deblurring.

8.2 Future work

There are three interesting directions to explore future research on motion deblurring. The following section describes these ideas in detail.

8.2.1 General Non-Uniform Model

In Chapter 7, we presented a model that takes scene depth into consideration. Since our depth-aware blur model is based on multiple layers with constant depth in each, we mainly handle scenes with multiple objects in frontal views. It may fail for the scenes that contain elongated objects with large depth variation. It would be interesting to exploit more general models to resolve more complex scenes of depth variation. Moreover, similar to the existing methods, the computational load of our non-uniform deblurring algorithm is high. Thus, how to efficiently solve the complex system and reduce the computational load is an interesting research topic.

8.2.2 Mobile Deblurring System

Unlike priors commonly used for latent images (e.g., those derived from natural image statistics), the distribution of camera motions is difficult to model due to its large diversity and inherent randomness. To gain additional information on such motion, inertial sensors such as gyroscopes and accelerometers have recently been exploited, with blur kernels estimated directly from the recorded camera motions and then used together with non-blind deconvolution techniques to recover the sharp latent image. These sensors have become standard build-in features in current smartphones, and it would be very useful to design mobile deblurring systems based on the sensors. However, the inertial sensors from current smartphones present some challenging issues. First, the time delay between sensor and image measurements and the rotational center of camera motion can vary with each photograph. Second, the noise in inertial sensor readings cannot be completely eliminated by offline calibration. Additionally, inaccurate accelerometer readings can accumulate into substantial errors in measured camera translation. We would like to propose a mobile deblurring system to address these issues in our future work, including how to dynamically recover the time delay between the inertial and camera sensors, how to calibrate for the rotational center of camera motion online, and how to refine the blur kernels initially

reconstructed from noisy sensor readings.

8.2.3 Joint Object Segmentation and Motion Deblurring

Object motion blur is challenging as the moving objects and the background in the scene undergo different blur. To remove the motion blur, a good segmentation of the moving objects is essential. Most of the recent object motion deblurring methods address the problem by segmenting the input image into multiple object layers and applying blind deconvolution to each segmented region. However, segmenting the objects itself is a difficult problem due to the ambiguous boundaries between objects. We would like to develop a unified model to jointly estimate the segmentation and the motion of each object in our future work.

Bibliography

- [1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing over-complete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
- [2] J. Bardsley, S. Jefferies, J. Nagy, and R. Plemmons. A computational method for the restoration of images with an unknown, spatially-varying blur. *Optics Express*, 14(5):1767–1782, 2006.
- [3] M. Ben-Ezra and S. Nayar. Motion deblurring using hybrid imaging. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–664, 2003.
- [4] M. Ben-Ezra and S. Nayar. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–698, June 2004.
- [5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- [6] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 104–111, 2009.
- [7] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Framelet-based blind motion deblurring from a single image. *IEEE Transactions on Signal Processing*, 21(2):562–572, 2012.
- [8] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2512–2519, 2010.
- [9] J. Chen, L. Yuan, C.-K. Tang, and L. Quan. Robust dual motion deblurring. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [10] Q. Chen, D. Li, and C.-K. Tang. KNN matting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–876, 2012.

- [11] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.
- [12] S. Cho and S. Lee. Fast motion deblurring. In *Proceedings of ACM SIGGRAPH, Asia*, 2009.
- [13] S. Cho., Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *Proceedings of IEEE International Conference on Computer Vision*, 2007.
- [14] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. In *Proceedings of IEEE International Conference on Computer Vision*, 2011.
- [15] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman. A content-aware image prior. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 169–176, 2010.
- [16] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman. Blur kernel estimation using the radon transform. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 241–248, 2011.
- [17] D. L. Donoho. For most large underdetermined systems of linear equations the minimal l_1 -norm solution is also the sparsest solutions. *Communications on Pure and Applied Mathematics*, pages 797–829, 2004.
- [18] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *Proceedings of ACM SIGGRAPH*, pages 787–794, 2006.
- [19] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *Proceedings of European Conference on Computer Vision*, pages 622–635, 2012.
- [20] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2008.
- [21] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proceedings of European Conference on Computer Vision*, pages 171–184, 2010.
- [22] S. Harmeling, M. Hirsch, and B. Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Advances in Neural Information Processing Systems*, 2010.
- [23] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf. Multiframe blind deconvolution, super-resolution, and saturation correction via incremental EM. In *Proceedings of IEEE International Conference on Image Processing*, 2010.
- [24] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *Proceedings of IEEE International Conference on Computer Vision*, pages 463–470, 2011.

- [25] W. Hu, J. Xue, and N. Zheng. PSF estimation via gradient domain correlation. *IEEE Transactions on Image Processing*, 21(1):386–392, 2012.
- [26] Z. Hu and M.-H. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In *Proceedings of British Machine Vision Conference*, 2012.
- [27] Z. Hu and M.-H. Yang. Good regions to deblur. In *Proceedings of European Conference on Computer Vision*, pages 59–72, 2012.
- [28] B.-S. Hua and K.-L. Low. Interactive motion deblurring using light streaks. In *Proceedings of IEEE International Conference on Image Processing*, 2011.
- [29] H. Ji and K. Wang. A two-stage approach to blind spatially-varying motion deblurring. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 2012.
- [30] J. Jia. Single image motion deblurring using transparency. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [31] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *Proceedings of ACM SIGGRAPH*, page 30, 2010.
- [32] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [33] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *Proceedings of European Conference on Computer Vision*, pages 27–40, 2012.
- [34] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, 2009.
- [35] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 233–240, 2011.
- [36] S. Kumar and M. Hebert. Discriminative random field. *International Journal of Computer Vision*, 68(2):179–201, 2006.
- [37] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, pages 282–289, 2001.
- [38] A. Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*, pages 841–848, 2006.

- [39] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *Proceedings of ACM SIGGRAPH*, 2007.
- [40] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2009.
- [41] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2657–2664, 2011.
- [42] F. Li, J. Yu, and J. Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [43] Y. Lou, A. Bertozzi, and S. Soatto. Direct sparse deblurring. *Journal of Mathematical Imaging and Vision*, 39(1):1–12, 2011.
- [44] L. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):745–754, 1974.
- [45] C. Paramanand and A. N. Rajagopalan. Non-uniform motion deblurring for bilayer scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [46] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. In *Proceedings of ACM SIGGRAPH*, pages 795–804, 2006.
- [47] W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972.
- [48] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009.
- [49] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *Proceedings of ACM SIGGRAPH*, 2008.
- [50] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *Proceedings of IEEE International Conference on Computer Vision*, 2007.
- [51] O. Šindelář and F. Šroubek. Image deblurring in smartphone devices using built-in inertial measurement sensors. *Journal of Electronic Imaging*, 22(1):011003–011003, 2013.

- [52] M. Sorel and J. Flusser. Space-variant restoration of images degraded by camera motion blur. *IEEE Transactions on Signal Processing*, 17(2):105–116, 2008.
- [53] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *Proceedings of European Conference on Computer Vision*, pages 582–595, 2008.
- [54] Y.-W. Tai, X. Chen, S. Kim, S. J. Kim, F. Li, J. Yang, J. Yu, Y. Matsushita, and M. S. Brown. Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2498–2512, 2013.
- [55] Y.-W. Tai, H. Du, M. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1012–1028, June 2010.
- [56] Y. W. Tai, H. Du, M. S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [57] Y. W. Tai, P. Tan, and M. S. Brown. Richardson-Lucy deblurring for scenes under projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1603–1618, 2011.
- [58] B. Taskar, V. Chatalbashev, and D. Koller. Learning associative Markov networks. In *Proceedings of International Conference on Machine Learning*, page 102, 2004.
- [59] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. *International Journal of Computer Vision*, 110(2):185–201, 2014.
- [60] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 491–498, 2010.
- [61] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2):168–186, 2012.
- [62] N. Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. MIT Press, 1964.
- [63] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of European Conference on Computer Vision*, pages 157–170, 2010.
- [64] L. Xu and J. Jia. Depth-aware motion deblurring. In *Proceedings of IEEE International Conference on Computational Photography*, 2012.
- [65] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l_0 gradient minimization. *Proceedings of ACM SIGGRAPH*, 30(6):174, 2011.

- [66] L. Xu, S. Zheng, and J. Jia. Unnatural l_0 sparse representation for natural image deblurring. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013.
- [67] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, pages 236–239. Morgan Kaufmann Publishers Inc., 2003.
- [68] Y. Yitzhaky, I. Mor, A. Lantzman, and N. Kopeika. Direct method for restoration of motion-blurred images. *Journal of the Optical Society of America*, 15(6):1512–1519, 1998.
- [69] L. Yuan, J. Sun, L. Quan, and H. Shum. Image deblurring with blurred/noisy image pairs. In *Proceedings of ACM SIGGRAPH*, 2007.
- [70] H. Zhang, D. Wipf, and Y. Zhang. Multi-observation blind deconvolution with an adaptive sparse prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1628–1643, 2014.