Ph.D. DISSERTATION

Dynamic Scene Deblurring: New Models, Algorithms, and Analysis

동적 환경 디블러링을 위한 새로운 모델, 알고리즘, 그리고 해석에 관한 연구

BY

TAE HYUN KIM

August 2016

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

Blurring artifacts are the most common flaws in photographs. To remove these artifacts, many deblurring methods which restore sharp images from blurry ones have been studied considerably in the field of computational photography. However, state-of-the-art deblurring methods are based on a strong assumption that the captured scenes are static, and thus a great many things still remain to be done. In particular, these conventional methods fail to deblur blurry images captured in dynamic environments which have spatially varying blurs caused by various sources such as camera shake including out-of-plane motion, moving objects, depth variation, and so on. Therefore, the deblurring problem becomes more difficult and deeply challenging for dynamic scenes.

Therefore, in this dissertation, addressing the deblurring problem of general dynamic scenes is a goal, and new solutions are introduced, that remove spatially varying blurs in dynamic scenes unlike conventional methods built on the assumption that the captured scenes are static. Three kinds of dynamic scene deblurring methods are proposed to achieve this goal, and they are based on: (1) *segmentation*, (2) sharp *exemplar*, (3) *kernel-parametrization*. The proposed approaches are introduced from segment-wise to pixel-wise approaches, and pixel-wise varying general blurs are handled in the end.

First, the segmentation-based deblurring method estimates the latent image,

multiple different kernels, and associated segments jointly. With the aid of the joint approach, segmentation-based method could achieve accurate blur kernel within a segment, remove segment-wise varying blurs, and reduce artifacts at the motion boundaries which are common in conventional approaches. Next, an *exemplar*-based deblurring method is proposed, which utilizes a sharp exemplar to estimate highly accurate blur kernel and overcomes the limitations of the segmentation-based method that cannot handle small or texture-less segments. Lastly, the deblurring method using kernel-parametrization approximates the locally varying kernel as linear using motion flows. Thus the proposed method based on kernel-parametrization is generally applicable to remove pixel-wise varying blurs, and estimates the latent image and motion flow at the same time.

With the proposed methods, significantly improved deblurring qualities are achieved, and intensive experimental evaluations demonstrate the superiority of the proposed methods in dynamic scene deblurring, in which state-of-the-art methods fail to deblur.

Key words: Blind deblurring, Non-uniform blur, Spatially varying blur, Dynamic Scenes deblurring, Motion segmentation, Kernel-parametrization, Exemplar

Student number: 2011-30224

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During an exposure time, CCD and CMOS sensors integrate photons, and the collected photons are converted to brightness values in digital images. Under low light conditions, integration time can be increased up to several seconds to collect more photons and obtain brighter images. However, motion blurs during the long exposure period severely degrade images. As these blurs are most common artifact in photographs and recent demands for clear images are very high, deblurring has become a popular research topic in computational photography; these research efforts aim to restore sharp photos on memorable days, and to improve the accuracy of numerous computer vision applications.

In particular, blind deblurring is a method used to restore sharp images from blurry ones caused by camera shake or moving object during shutter time. However, the blind deblurring problem is difficult to solve because it is highly ill-posed. To solve this problem, many researchers have studied the deblurring algorithms for the latent image and blur kernel estimation, and recast the deblurring problem as an energy minimization problem with data and regularization terms.

Figure 1.1: A real image with locally varying blurs captured in a dynamic scene.

Specifically, uniform and shift-invariant blur from 2D translational camera shake was handled in earlier works [16, 3, 17, 1, 6]. These were followed by work on the more general case of camera shake in 3D space [5, 4, 18, 2, 6], and thus motion blurs caused by camera shakes have been studied intensively under an assumption that the captured scenes are static. However, blurs in real images captured in dynamic scenes are locally different as shown in Fig. 1.1, since they are caused by co-existing different sources such as camera shake, moving objects, and depth variation. Therefore, blur kernels are also spatially varying, and the deblurring problem becomes more difficult and very challenging. Notably, solution space of the unknown blur kernel becomes very huge when the blur kernel changes spatially.

Therefore, the main goal of this dissertation is to remove spatially varying blurs and achieves good deblurring results in dynamic scenes. To do so, several methods to solve the difficulties in dynamic scene deblurring problem are proposed, and

Proposed methods	Blur kernel applicability
• Deblurring method with segmentation	• Low (segment-wise varying blurs)
• Deblurring method with exemplar	• Mid (segment-wise varying blurs)
• Deblurring methods with kernel-parametrization	• High (pixel-wise varying blurs)

Figure 1.2: Summary of the methods proposed in this dissertation.

the three kinds of proposed methods are based on: (1) *segmentation*, (2) *exemplar*, (3) *kernel-parametrization*. As shown in Fig. 1.2, proposed methods are introduced from segment-wise approach to pixel-wise approach, and handle general pixel-wise varying blurs in the end. In particular, proposed methods with segmentation and exemplar assume that blurs are caused by segment-wise rigid motion, and thus estimate segment-wise varying kernels. Therefore, they can estimate accurate blur kernel within a segment, but cannot handle pixel-wise varying blurs. Meanwhile, proposed deblurring methods using kernel-parametrization assume pixel-wise varying blurs, thus estimate pixel-wise varying blur kernels using a piece-wise linear kernel approximation. The structure of the dissertation, detailed introductions and contributions of the proposed methods are summarized in the followings.

In Chapter 2, a *segmentation*-based deblurring approach is introduced [10]. As moving objects and background have different motion blurs in dynamic scenes, accurate (motion) segmentation is required for removing each distinct blur. Thus, using a segment-wise rigid motion assumption, locally different motion blurs and their

associated weights, and the latent sharp image are jointly estimated with a novel energy model designed with the weighted sum of multiple blur data models. In this framework, the local weights are determined adaptively and get high values when the corresponding data models have high data fidelity. And, the weight information is used for the segmentation of the motion blur. Moreover, non-local regularization of weights are incorporated to produce more reliable segmentation results, and a convex optimization-based method is used for minimizing the proposed energy model.

In Chapter 3, an *exemplar*-based deblurring method is proposed. As the segmentation-based method proposed in Chapter 2, attempts to address dynamic scene deblurring problem by handling segment-wise different blurs, it achieves good deblurring results when segmentation succeeds. However, segmentation itself with a single image is very difficult problem, and thus the segmentation-based method occasionally fails where segments of the moving objects are small or texture-less. In contrast, in this chapter, sharp exemplar image of the scene, which can be obtained from bracket sequence imaging or a nearby video frame, to elevate the deblurring quality for dynamic scenes by overcoming the limitations of the segmentation-based method, is utilized. The key issue in utilizing sharp exemplars is to establish dense correspondences between the sharp and blurred images in the presence of spatially-variant blurs. This problem is addressed with a new blur-aware optical flow estimation method that is based on the notion that pixels which have similar motion will exhibit similar blur. In an iterative process, the proposed method computes flows, aligns the sharp exemplar to the blurred image according to this flow, uses the flow-aligned exemplar image to facilitate the estimation of blur kernels, their associated image segments, and the latent image.

In Chapter 4, a deblurring method using *kernel-parametrization*, which is gen-

erally applicable to handle pixel-wise varying blurs, is proposed to deblur a single blurry image [8]. Because our segmentation-based and exemplar-based methods estimate segment-wise different blur kernel, they cannot handle pixel-wise varying blurs in practice. Thus, in this chapter, a segmentation-free dynamic scene deblurring method, which handles pixel-wise varying blurs unlike other conventional methods, is studied. To be specific, the approximation technique of the local blur kernel as a motion model is employed, and the problem is recast as to infer the latent image and pixel-wise varying motion flows jointly in an energy minimization framework. By doing so, various types of blur can be handled caused by camera shake including out-of-plane motion, moving objects, depth variation, and so on. Moreover, to overcome the limitation of the conventional blur constraint in the joint formulation, which fail to deblur where edges are not significant, motion blur cues from image-statistics are integrated into the proposed energy model. In addition, a higher order regularizer is employed to reduce staircase-like artifacts among neighboring motion flows on slanted surfaces. Moreover, an efficient solvers to minimize the proposed non-convex energy function is presented.

Lastly, in Chapter 5, *kernel-parametrization* technique is naturally extended to remove blurs in videos [19], and it improves deblurring qualities greatly using additional temporal information in videos. To handle spatially and temporally varying blurs in dynamic videos caused by various sources, bidirectional optical flows are inferred to handle motion blurs more accurately. Moreover, Gaussian blur maps are estimated to remove optical blur from defocus in our new blur model. In doing so, a single energy model is proposed, that jointly estimates optical flows, defocus blur maps and latent frames. Moreover, a framework and efficient solvers to minimize the proposed energy model is provided. To evaluate the performance of non-uniform

deblurring methods objectively, a new realistic dataset with ground truths have been constructed. Extensive experimental results on publicly available challenging video data demonstrate that the proposed method produces qualitatively superior performance than the state-of-the-art methods which often fail in either deblurring or optical flow estimation.

Conclusion of the dissertation is given in Chapter 6 with a summary, and suggestions for future works are also provided.

Chapter 2

Image Deblurring with Segmentation

2.1 Introduction and Related Work

Blurring artifacts are among the most common flaws in photographs. Camera shake or moving objects during the exposure time cause these artifacts under low light conditions. To address this problem, single image deblurring methods which restore a sharp image from a blurred image have been researched considerably in the field of computer vision with the recent increased demand for clear images.

In general, the blind deblurring problem that restores the blurry image without knowing the blur kernel is highly ill-posed. Therefore various energy models that are composed of the regularization and data term have been proposed to find the sharp image and the blur kernel jointly, in the form of

$$\mathbf{E}(\mathbf{L}, \mathbf{K}) = \mathbf{E}_{data}(\mathbf{L}, \mathbf{K}, \mathbf{B}) + \mathbf{E}_{reg}(\mathbf{L}, \mathbf{K}), \tag{2.1}$$

where $\mathbf{L}$ and $\mathbf{B}$ denote the vector form of the latent and blurred images, respectively. The matrix $\mathbf{K}$ denotes the blur kernel whose row vector corresponds to the blur kernel placed at each pixel location. The data term $\mathbf{E}_{data}$ measures the data fidelity and the regularization term $\mathbf{E}_{reg}$ enforces the smoothness constraint to the latent image as well as to the blur kernel.

Depending on the type of the blur kernel, blind deblurring approaches can be categorized into two types. One is the uniform kernel approach which assumes shift invariant blur kernel, and the other is the non-uniform approach which assumes spatially varying blur kernel.

When the blur kernel is shift invariant and uniform over the entire image [16, 3], it is possible to restore the latent image quickly with the aid of a fast Fourier transform (FFT) and additional parallel processing [20, 17, 1]. However, the assumption of shift invariant blur kernel does not hold good when there exists a rotational movement of the camera or a moving object in the scene.

To alleviate these limitations of uniform motion blur assumption, several non-uniform kernel based methods are proposed. In particular, recent approaches focus on modeling the rotation of camera as well as translation [5, 4, 2], and they obtained promising results in the deblurring of *static* scene.

However, problems still remain in more general settings where not only camera shake but also moving objects exist. For example, in Fig. 2.1, restoring the image with the uniform blur kernel that has been estimated from the moving bus raises a severe artifacts in the background region (Fig. 2.1(b)). And also the uniform kernel estimated from the background fails deblurring the bus (Fig. 2.1(c)). Note that even the state-of-the-art non-uniform blur kernel method [2] which can deblur rotational camera shake does not restore the moving bus either (Fig. 2.1(d)).

Therefore the dynamic scene deblurring problem is deeply challenging. Thus far, only a limited amount of research has been done on this problem [21, 22, 23]. However, many of these works are still in a nascent stage, and even the hardware-assisted method has difficulties in handling dynamic scene deblurring problem [24].

Levin [21], for the first time, proposed a sequential two-stage approach to solve this problem. She argued that moving objects and background should be handled with different blur kernels to remove the artifacts from deblurring with an inaccurate blur kernel. To begin with, she segmented blur motions by comparing likelihoods with a set of given one dimensional box filters, then applied the Richardson-Lucy deconvolution algorithm to each segmented region with its corresponding box filter. For the first time, she approached this challenging problem with a simple and intuitive way. However, the kernel for the segmentation is limited to the box filters and thus the poor segmentation results could cause undesirable artifacts since the segmentation-stage and the deblurring-stage are separated. Harmeling et al. [22] proposed a method that restores overlapping patches of the blurred image. This approach could handle smoothly varying blur kernels but could not handle the abrupt change of the blur kernel near the boundary of moving objects since they did not segment the motion blurs. More recent work of Ji et al. [23] is based on the interpolation of initially estimated kernels and showed much better results by reducing errors from inaccurate blur kernels, but it also could not overcome the motion boundary problems as in [22].

Figure 2.1: Comparison of deblurring results. The green box illustrates the region used for blur kernel estimation for each image. (a) Blurry image. (b)-(c) Results of Xu et al. [1]. (d) Result of Whyte et al. [2]. (e)-(f) Our segmentation and deblurring results.

In principle, the dynamic scene deblurring problem also requires the segmentation of differently blurred regions. Therefore, in this work, we address the problem of estimating latent image as well as different blur motions and their implicit (soft) segmentations. Note that, to the best of our knowledge, this work [10] is the first dynamic scene deblurring approach that estimates these variables jointly. In our framework, we propose a new energy model including multiple blur kernels and their associated pixel-wise weights. The weight of a kernel takes high value he kernel gives high data fidelity. At the same time, the blur kernels are estimated from the pixels whose associated weights have high values. Therefore, locally varying weight information allow us to segmentation the blur motions. In addition, we add non-local regularization to the weight variables to enforce the smoothness in segmentation.

In this chapter, we introduce a more general and new deblurring framework that can adaptively combine different blur models to estimate the spatially varying blur kernels. Also, as illustrated in Fig. 2.1(e)-(f), we provide the segmentation of the motion blur as well as better deblurring results. Note that since our framework is general in nature, any blur models and optimization method can be incorporated. We demonstrate the effectiveness of our new deblurring framework by the test results on very challenging images on which conventional techniques break down.

2.2 Segmentation-based Dynamic Scene Deblurring Model

In our segmentation-based dynamic scene deblurring model, we assume the existence of various blur motions. Therefore, find both blur kernels and their corresponding blur regions should be estimated jointly. Moreover, we employ both the uniform kernels, which are simple and fast, and the non-uniform kernels which can handle

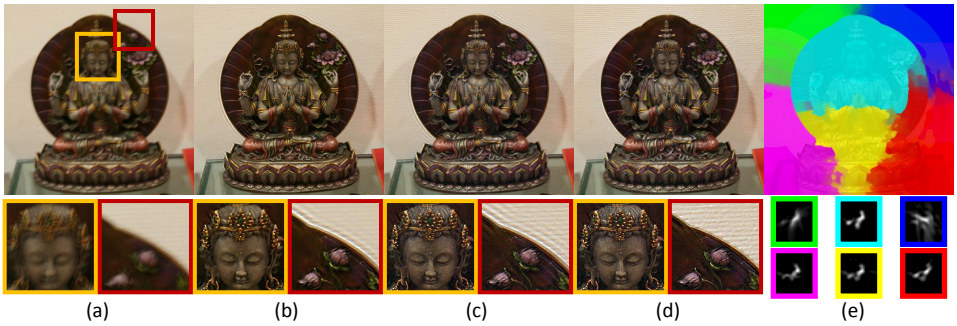


Figure 2.2: Multiple blur kernel models give a much better result without additional process to remove ringing artifact. (a) A blurry image of static scene. (b) Deblurring result of Shan et al. [3]. (c) Our deblurring result with one uniform kernel. (d) Our segmentation-based deblurring result. (e) Illustration of six weight variables. Slightly different blur kernels are estimated.

camera rotation.

As there are multiple blur kernels in a dynamic scene, each blurred pixel should be restored from one of them and each kernel should be estimated from its related pixels. For this, we introduce pixel-wise weight variables. A pixel-wise weight variable is associated with a blur kernel and it gains high values on the pixels related with the blur kernel. Therefore, the weight variables imply the segmentation of motion blur, and the proposed energy model is given by,

$$\mathbf{E} = \mathbf{E}_{data}(\mathbf{L}, \mathbf{W}, \mathbf{A}, \mathbf{B}) + \mathbf{E}_{reg}(\mathbf{L}, \mathbf{W}, \mathbf{A}). \tag{2.2}$$

The set $\mathbf{A} = \{\mathbf{K}_n\}$ denotes a set of N blur kernels, and the set $\mathbf{W} = \{\mathbf{W}_n\}$ means a set of N weight variables where $n = 1, 2, \ldots, N$. Weighting matrix $\mathbf{W}_n$ is associated with the corresponding blur kernel $\mathbf{K}_n$.

Compared with the conventional model in (2.1), our new energy model involves additional weight variables and multiple blur kernels, so it becomes a more complex and challenging model. In particular, the model in (2.1) is a special case of the proposed model in (2.2), ours can also render reliable results even for static scenes.

Note that even in the case of a static scene with only translational camera shake, the captured real image may contain various blur motions because depth variation or radial distortion may generate unexpected blur effects. Since the proposed model employs multiple blur kernels, it could handle this problem complementary and produce much better results than the conventional methods. In Fig. 2.2, similar but slightly different six uniform blur kernels and their associated segments are jointly estimated and a much clearer image with less ringing artifacts is obtained by our method.

2.2.1 Adaptive blur model selection

To handle locally varying blurs, a data term that adaptively selects and fuses proper blur models among candidate models is proposed. In doing so, we adopt a strategy that chooses the locally (pixel-wise) best suited model by measuring the data fidelities and gives a high value to the associated weight variable. At the same time, to obtain correct blur kernels, it is required that each blur kernel is estimated from pixels whose associated weight variable shows high values. In this way, the set $\mathbf{W}$ segments the motion blurs by selecting the locally best suited data model. The data term of the proposed new energy model is formulated by a weighted sum of the multiple data models with some constraints as follows, and minimizing it is equal to select locally best data model.

$$E_{data}(\mathbf{L},\mathbf{W},\mathbf{A},\mathbf{B}) = \lambda_s \sum_{n=1}^{N} \sum_{\partial_*} \|\mathbf{W}_n^{\frac{1}{2}} \odot (\partial_* \mathbf{K}_n \mathbf{L} - \partial_* \mathbf{B})\|^2 \tag{2.3}$$

where N is the number of maximal blur models in the scene and λ_s is the parameter adjusting the scale of our data term and the continuous weight vector is constrained

to be (pixel-wise) $\mathbf{W}_n \succeq \boldsymbol{0}$ and $\sum_{n=1}^{N} \mathbf{W}_n = \boldsymbol{1}$. The operator $\odot$ means the element-wise (Hadamard) product of two vectors and the operator $\partial_* \in \{\partial_x, \partial_y\}$ denotes the partial derivative in horizontal and vertical directions [17]. To reduce ringing artifacts, we also use gradient maps, but we do not use brightness map or second order gradient maps unlike conventional methods [3, 17, 1]. Despite this, we can obtain satisfying results by means of multiple blur models and reduce the computational cost.

2.2.2 Regularization

As dynamic scene deblurring is a highly ill-posed problem, regularization enforcing the smoothness of variables is necessary to obtain a reliable solution. In our energy model, three primal variables are the latent image $\mathbf{L}$, the set of blur kernel matrices $\mathbf{A}$ and the set of the weight variables $\mathbf{W}$, and each has different kinds of regularization as

$$\mathbf{E}_{reg}(\mathbf{L}, \mathbf{W}, \mathbf{A}) = \mathbf{E}_{reg}(\mathbf{L}) + \mathbf{E}_{reg}(\mathbf{W}) + \mathbf{E}_{reg}(\mathbf{A}), \tag{2.4}$$

and the details of which are described in the following sections.

2.2.2.1 Regularization for the latent image

We design the latent image to be sharp in edge regions and smooth in flat regions to suppress noise. For this purpose many researchers have studied various priors of the latent image and it is known that l_p norm on the gradient map with $0.7 \leq p \leq 1$ could capture the statistics of natural images with heavy tailed distribution [25, 26, 5]. However, conventional optimization algorithms with sparse norm less than $p < 1$ are hard to optimize and require additional computational efforts. Thus, our model

adopts the total variation model used in [1] as the prior of the latent image, as follows:

$$\mathbf{E}_{reg}(\mathbf{L}) = |\nabla \mathbf{L}|. \tag{2.5}$$

2.2.2.2 Regularization for the weight variables

We assumed that a blurry object can be restored by one of the various blur models, and the motion blur does not change abruptly except on the boundary of a moving object. So, the locally varying weight variable should be segmented and we adopt non local regularization for this purpose.

Non-local regularization is widely used in computer vision and have come into the spotlight lately [27, 28]. The formulation incorporated in our deblurring model is,

$$\mathbf{E}_{reg}(\mathbf{W}) = \sum_{n=1}^{N} \sum_{\mathbf{x},\mathbf{y} \in \mathcal{N}(\mathbf{x})} g_s(\mathbf{x},\mathbf{y}) \cdot |\mathbf{W}_n(\mathbf{x}) - \mathbf{W}_n(\mathbf{y})|, \tag{2.6}$$

where $\mathcal{N}(\mathbf{x})$ denotes neighboring pixels of $\mathbf{x}$ and the function $g_s(\mathbf{x},\mathbf{y})$ is a non-local similarity map which is used to define the mutual support between the pixels at positions $\mathbf{x}$ and $\mathbf{y}$. Similar to the work in [27] the non-local similarity map between two neighboring pixels is defined as

$$g_s(\mathbf{x},\mathbf{y}) = e^{-(\frac{\|\mathbf{x}-\mathbf{y}\|}{\sigma_D})^2} \cdot e^{-(\frac{\mathbf{L}_0(\mathbf{x})-\mathbf{L}_0(\mathbf{y})}{\sigma_I})^2}, \tag{2.7}$$

where the parameters σ_D and σ_I are used to adjust the slope of the non-local similarity map and the given latent image $\mathbf{L}_0$ can be obtained from the initial of each level in the coarse to fine approach or previous result in the iterative optimization procedure. To reflect the properties of the weight variable $\mathbf{W}_n$, with similar values

between neighboring pixels but discontinuity on the boundary of moving objects, it is necessary to use a model that could give sparsity on the difference of weights between neighboring pixels. For this reason, our regularization of weight variables is also based on the total variation. Note that, in contrast to the result from weight regularization with only four neighbors, the result from regularization with dozens of neighbors is much better in both motion blur segmentation and deblurring as shown in Fig. 2.3.

2.2.2.3 Regularization for the blur kernels

As we use both uniform and non-uniform kernels in our blur models, two different regularization models are required.

First, if the blur kernel matrix consists of uniform blur kernel, we use Tikhonov regularization which is typically used in other methods of uniform blur kernel regularization due to its simplicity [20, 17, 1]. By using this regularization, we can have a smooth kernel. The energy function for regularization on an uniform kernel is formulated by

$$\mathbf{E}_{reg}(\mathbf{K}_n) = \beta_s \|\mathbf{k}_n\|^2, \tag{2.8}$$

where $\mathbf{k}_n$ is the uniform blur kernel corresponding to the kernel matrix $\mathbf{K}_n$ and the parameter β_s controls the influence of the regularization.

Secondly, for a non-uniform blur kernel $\mathbf{K}_n$, we also use Tikhonov regularization but in a different manner from the case of uniform blur kernel because a non-uniform kernel is estimated in a different way. To be specific, similar to [5, 4], the non-uniform kernel matrix $\mathbf{K}_n$ is made by restricting it to linear combinations of the several basis

kernels as

$$\mathbf{K}_n = \sum_{m=1}^{M} \mu_{m,n} \mathbf{b}_m, \tag{2.9}$$

where the vector $\mathbf{b}_m$ is the m_{th} basis kernel induced by a possible camera shake and M denotes the total number of basis kernels. $\mu_{m,n}$ is the coefficient of basis kernel $\mathbf{b}_m$ which satisfies $\mu_{m,n} \geq 0$ and $\sum_{m=1}^{M} \mu_{m,n} = 1$. Since the non-uniform kernel is determined by $\boldsymbol{\mu}_n = [\mu_{1,n}, \mu_{2,n}, \cdots, \mu_{M,n}]^T$, we regularize $\boldsymbol{\mu}_n$ instead of regularizing the kernel matrix $\mathbf{K}_n$ itself. Then the energy function for regularization of the non-uniform kernel is formulated by

$$E_{reg}(\mathbf{K}_n) = \gamma_s \|\boldsymbol{\mu}_n\|^2, \tag{2.10}$$

where the parameter γ_s adjusts the scale of regularization on $\boldsymbol{\mu}_n$.

2.3 Optimization

The proposed dynamic scene deblurring model introduced in the previous section and the final objective function is as follows:

$$\begin{aligned}\min_{\mathbf{L},\mathbf{W},\mathbf{A}} \; & |\nabla \mathbf{L}| + \beta_s \sum_{\substack{n=1 \\ (uniform)}}^{N} \|\mathbf{k}_n\|^2 + \gamma_s \sum_{\substack{n=1 \\ (non-uniform)}}^{N} \|\boldsymbol{\mu}_n\|^2 + \\ & \lambda_s \sum_{n=1}^{N} \sum_{\partial_*} \|\mathbf{W}_n^{\frac{1}{2}} \odot (\partial_* \mathbf{K}_n \mathbf{L} - \partial_* \mathbf{B})\|^2 + \sum_{n=1}^{N} \sum_{\mathbf{x},\mathbf{y} \in \mathcal{N}(\mathbf{x})} g_s(\mathbf{x},\mathbf{y}) |\mathbf{W}_n(\mathbf{x}) - \mathbf{W}_n(\mathbf{y})|,\end{aligned} \tag{2.11}$$

where $\mathbf{W}_n(\mathbf{x}) \geq 0$ and $\sum_{n=1}^{N} \mathbf{W}_n(\mathbf{x}) = 1$. Although our final objective model is not a jointly convex function, three sub-problems with respect to $\mathbf{L}$, $\mathbf{W}$ and $\mathbf{A}$ are

convex. Therefore, instead of using complex optimization such as sampling based technique for global optimum, we propose an iterative optimization method similar to [17, 1, 3] for easier inference. By alternatively optimizing each subproblem in an iterative process, we can efficiently estimate the $\mathbf{L}$, $\mathbf{W}$ and $\mathbf{A}$ and can obtain successful results. Each proposed subproblem can be modeled as a convex function, and we adopt the first-order primal-dual algorithm [29] to solve each problem.

2.3.1 Sharp image restoration

Sharp image restoration methods are widely researched in both non-blind and blind deblurring methods and some fast solutions are available with the aid of FFT. The update procedure of $\mathbf{L}$ by the first-order primal-dual algorithm is

$$\begin{cases} \mathbf{q}^{m+1} = \frac{\mathbf{q}^m + \sigma_L \mathbf{S}\mathbf{L}^m}{\max(\mathbf{1}, \mathbf{q}^m + \sigma_L \mathbf{S}\mathbf{L}^m)} \\ \mathbf{L}^{m+1} = \arg\min_{\mathbf{L}} \frac{(\mathbf{L} - (\mathbf{L}^m - \tau_L \mathbf{S}^T \mathbf{q}^{m+1}))^2}{2\tau_L} + \lambda_s \sum_{n=1}^{N} \sum_{\partial_*} |\mathbf{W}_n^{\frac{1}{2}} \odot (\partial_* \mathbf{K}_n \mathbf{L} - \partial_* \mathbf{B})\|^2, \end{cases} \tag{2.12}$$

where $m \geq 0$ means iteration number, $\mathbf{q}$ denotes the dual variable of $\mathbf{L}$ defined on the vector space and $\mathbf{S}$ is a continuous linear operator that calculates the difference between two pixels. The update steps σ_L and τ_L control the convergence rate as defined in [29]. Initially, $\mathbf{q}^0 = \mathbf{0}$ and $\mathbf{L}^0 = \mathbf{B}$. In particular, since the primal update for $\mathbf{L}$ in (2.12) is a quadratic form, we have adopted the Landweber method [30] to solve with FFT similar to [20].

2.3.2 Weight estimation

The use of non-local regularization and constraints on weight variables makes it hard to infer, but with the aid of convexity, we also adopt the first-order primal-dual algorithm and the update step is given by,

$$\begin{cases} \mathbf{r}_{n,\mathbf{y}}^{m+1}(\mathbf{x}) = \min(g_s(\mathbf{x},\mathbf{y}), \max(-g_s(\mathbf{x},\mathbf{y}), \mathbf{r}_{n,\mathbf{y}}^{m}(\mathbf{x}) + \sigma_{\mathbf{w}}(\mathbf{Z}_{\mathbf{y}}\mathbf{W}_n^m)(\mathbf{x})) \\ \mathbf{W}_n^{m+1} = \mathbf{W}_n^m - \tau_{\mathbf{w}}(\sum_{\mathbf{y}} \mathbf{Z}_{\mathbf{y}}^T \mathbf{r}_{n,\mathbf{y}}^{m+1} + \lambda_s \sum_{\partial_*} (\partial_* \mathbf{K}_n \mathbf{L} - \partial_* \mathbf{B}) \odot (\partial_* \mathbf{K}_n \mathbf{L} - \partial_* \mathbf{B})) \\ \mathbf{W}^{m+1} = \Pi_{\mathbf{W}}(\mathbf{W}^{m+1}), \end{cases} \tag{2.13}$$

where $\mathbf{r}_{n,\mathbf{y}}$ is a dual variable defined on the vector space, and update steps $\sigma_{\mathbf{w}}$ and $\tau_{\mathbf{w}}$ control the convergence rate as defined in [29]. $\mathbf{Z}_{\mathbf{y}}$ is a continuous linear operator that calculates the difference between two neighboring pixels at $\mathbf{x}$ and $\mathbf{y}$. Since $\mathbf{W}$ has some constraints, $\mathbf{W}_n(\mathbf{x}) \geq 0$ and $\sum_{n=1}^{N} \mathbf{W}_n(\mathbf{x}) = 1$, the orthogonal projection $\Pi_{\mathbf{W}}$ projects $\mathbf{W}$ onto the unit simplex [31]. This projection converges with N iterations at most and the detail is in Algorithm 1.

Algorithm 1 The algorithm of projection onto unit simplex

1: $T = \{1, ..., N\}$
2: $\mathbf{W}_n(\mathbf{x}) \leftarrow \mathbf{W}_n(\mathbf{x}) - (\sum_n \mathbf{W}_n(\mathbf{x}) - 1)/|T|,$ if $n \in T$
3: $T \leftarrow T - \{n\},$ if $\mathbf{W}_n(\mathbf{x}) < 0$
4: $\mathbf{W}_n(\mathbf{x}) \leftarrow 0,$ if $n \notin T$
5: Repeat steps 2-4 until $\sum_{n=1}^{N} \mathbf{W}_n(\mathbf{x}) = 1$, for all $\mathbf{x}$.

For weight estimation, it is important to set proper initial value for $\mathbf{W}$. The initial weight map is designed so that at least one of the segments for uniform kernels can cover the moving object. However since we don't know the position and the size

of it, we use many overlapping segments to cover it initially. If the moving object occupies large part or has strong edges in an initial segment, then the blur kernel of the moving object can be roughly estimated from that segment. Then, by iterations, both the accuracies of the blur kernel and segment increase. An example of using 6 uniform blur kernels and 1 non-uniform blur kernel and their corresponding initial weight maps is illustrated in Fig. 2.4. We observed empirically that different settings of initial weight map do not change the results significantly. Thus, we used the same initial segmentations as in Fig. 2.4 and set the initial values of 1/3 for each segment for all experiments.

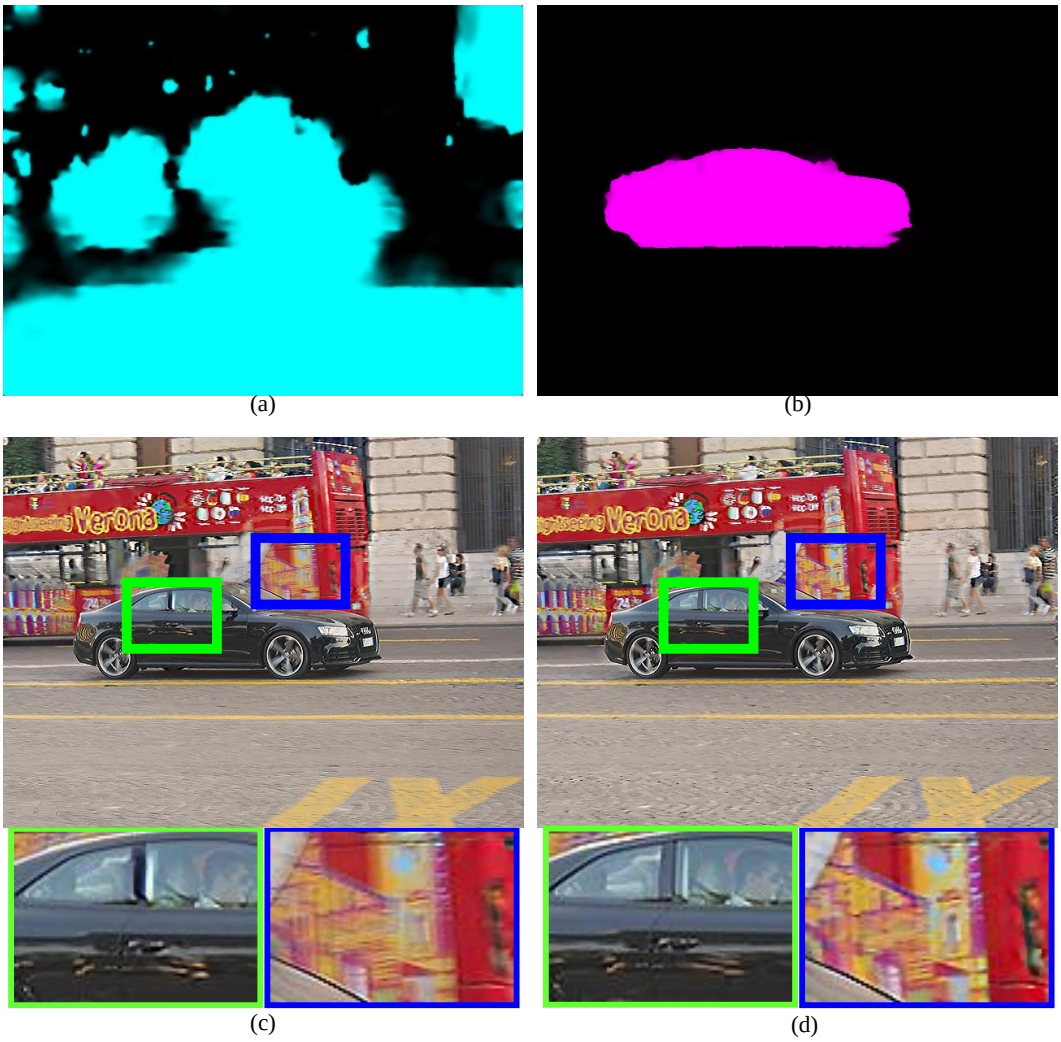


Figure 2.3: Comparison between conventional TV and non-local TV regularizations. (a) TV regularization model. (b) Non-local TV (80 neighboring pixels) regularization model. (c)-(d) Deblurring results with weight variables in (a) and (b) respectively.

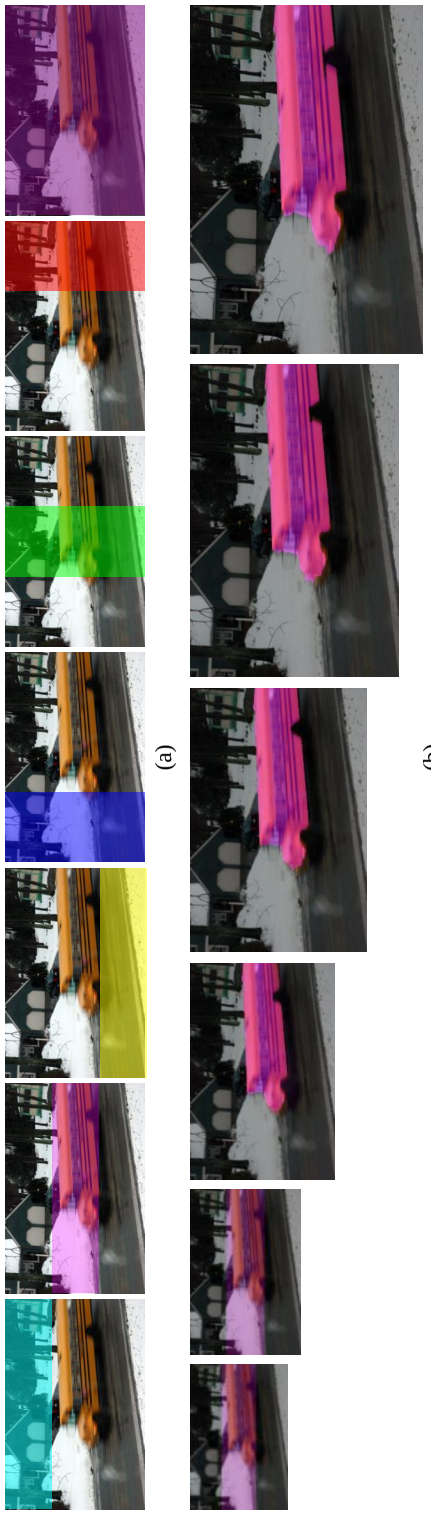


Figure 2.4: Weight initialization. (a) An example of the initial set-up of weight variables. The six columns on the left illustrate the initial weight variables corresponding to six uniform models, and the right most column shows the initial weight variable corresponding to a non-uniform model. (b) Change of a weight variable from coarse to fine level. The distribution of a weight variable gradually changes and finally fits on the moving bus.

2.3.3 Kernel estimation

The proposed method includes multiple blur models and a blur model could be either uniform and non-uniform kernel. Therefore, we have to estimate both uniform and non-uniform kernels. The blur kernel estimation methods for both approaches have been widely studied in blind deblurring methods, but ours is somewhat different because the proposed model includes additional weight variables. Since proper initial value for blur kernel is also important, the method guiding the latent image using prediction step [17, 1] is widely used. So, we adopt the predicted gradient maps $\{\mathbf{p}_x, \mathbf{p}_y\}$ defined in [17], instead of using latent image itself for accurate kernel estimation.

2.3.3.1 Uniform kernel estimation

For $\mathbf{L}$ and $\mathbf{W}$ being fixed, our energy model for uniform kernel $\mathbf{k}_n$ is quadratic and the solution can be easily obtained. The quadratic objective function with some constraints on the uniform kernel is given by

$$\begin{aligned}\min_{\mathbf{K}_n} \lambda_s((\mathbf{P}_x\mathbf{K}_n - \partial_x\mathbf{B})^T diag(\mathbf{W}_n)(\mathbf{P}_x\mathbf{K}_n - \partial_x\mathbf{B}) + \\ (\mathbf{P}_y\mathbf{K}_n - \partial_y\mathbf{B})^T diag(\mathbf{W}_n)(\mathbf{P}_y\mathbf{K}_n - \partial_y\mathbf{B})) + \beta_s\|\mathbf{k}_n\|^2.\end{aligned} \tag{2.14}$$

Note that matrices $\mathbf{P}_x$ and $\mathbf{P}_y$ consist of $\mathbf{p}_x$ and $\mathbf{p}_y$, respectively. To represent the formulation as a quadratic form we introduce $diag(\mathbf{W}_n)$ which is a diagonal matrix whose diagonal entries are diagonalized elements of $\mathbf{W}_n$, and non-diagonal entires are zeros. As this problem is convex, we can use any quadratic programming methods to solve it, and we have adopted the Landweber method [30] to iteratively minimize with FFT for reducing computations.

2.3.3.2 Non-uniform kernel estimation

Since non-uniform kernel $\mathbf{K}_n$ is a weighted sum of M basis kernels and the blurry image $\mathbf{B}$ is equal to $\mathbf{K}_n\mathbf{L}$, we can derive an equation,

$$\mathbf{K}_n\mathbf{L} = [\mathbf{b}_1\mathbf{L}, \mathbf{b}_2\mathbf{L}, \cdots, \mathbf{b}_M\mathbf{L}] \cdot \boldsymbol{\mu}_n. \tag{2.15}$$

Therefore, the minimization on the coefficient vector $\boldsymbol{\mu}_n$ for non-uniform kernel is given by

$$\min_{\boldsymbol{\mu}_n} \ \lambda_s(\|\mathbf{W}_n^{\frac{1}{2}} \odot (\mathbf{H}_x\boldsymbol{\mu}_n - \partial_x\mathbf{B})\|^2 + \|\mathbf{W}_n^{\frac{1}{2}} \odot (\mathbf{H}_y\boldsymbol{\mu}_n - \partial_y\mathbf{B})\|^2) + \gamma_s\|\boldsymbol{\mu}_n\|^2, \tag{2.16}$$

where $\mathbf{H}_x$ and $\mathbf{H}_y$ are derived from $\mathbf{K}_n\mathbf{p}_x = \mathbf{H}_x\boldsymbol{\mu}_n$ and $\mathbf{K}_n\mathbf{p}_y = \mathbf{H}_y\boldsymbol{\mu}_n$, respectively. Since this energy function can also be represented as the quadratic form, we can find an optimal $\mathbf{u}_i$ by quadratic programming. To be specific, the quadratic programming is formulated as

$$\min_{\boldsymbol{\mu}_n} \frac{1}{2}\boldsymbol{\mu}_n^T\mathbf{H}\boldsymbol{\mu}_n + \mathbf{f}^T\boldsymbol{\mu}_n, \tag{2.17}$$

where

$$\begin{cases} \mathbf{H} = \mathbf{H}_x^T diag(\mathbf{W}_n)\mathbf{H}_x + \mathbf{H}_y^T diag(\mathbf{W}_n)\mathbf{H}_y + \frac{\gamma_s}{\lambda_s}\mathbf{I}, \\ \mathbf{f} = \mathbf{H}_x^T diag(\mathbf{W}_n)\mathbf{B}_x + \mathbf{H}_y^T diag(\mathbf{W}_n)\mathbf{B}_y. \end{cases} \tag{2.18}$$

The minimization is performed by the interior point method and we can obtain the non-uniform blur kernel matrix as $\mathbf{K}_n = \sum_{m=1}^{M} \boldsymbol{\mu}_n(m)\mathbf{b}_m$ from (2.9).

2.3.4 Overall procedure

In the previous sections, we introduce the efficient minimization methods for the latent image, weights, and multiple blur kernels. However, there exist many unknown variables in our model and the traditional iterative optimization is prone to be stuck in local minimum. To alleviate this problem, we adopt coarse to fine approach like most recent blind deconvolution algorithms [1, 17, 32], and the overall procedure of our dynamic scene deblurring is in Algorithm 2.

Algorithm 2 The overall procedure of the proposed algorithm

Input: A blurry image $\mathbf{B}$

Output: $\mathbf{L}$, $\mathbf{W} = \{\mathbf{W}_n\}$ and $\mathbf{A} = \{\mathbf{K}_n\}$

1: Build an image pyramid, which has 5 levels, with a scale factor of 0.5
2: **for** $t = 1$ to 3 **do**
3: Update blur kernels with the predicted gradient maps $\{\mathbf{p}_x, \mathbf{p}_y\}$. (Sec. 2.3.3)
4: **for** $n = 1$ to 30 **do**
5: Continuous optimization of the latent image. (Sec. 2.3.1)
6: Continuous optimization of the weights. (Sec. 2.3.2)
7: **end for**
8: **end for**
9: Propagate variables to the next pyramid level if exists.
10: Repeat steps 2-9 from coarse to fine pyramid level.

2.4 Experiments

Although many parameters are used for our segmentation-based deblurring method, most of them are reliable and less sensitive to various blurry images except for the parameter λ_s. Since we do not estimate the noise level and the blur strength of

the input image, λ_s that adjusts the influence of data term should be tuned from the statistics of the input image. It ranges from 50 to 500 and it has a low value when the noise level is high or the blur is severe. The other parameters are fixed and we use six uniform kernel models and one non-uniform kernel model, so $N = 7$ in all experiments. By setting N as large as possible, we can handle various kinds of blur motions but it raises costs and thus we determined the number of models empirically and fixed it. Note, however, that the numbers of segmented regions in the final results in Fig. 2.5 are less than 7 and adaptive to each image. This is due to our sparsity priors to the weight variables. The initial values of the seven weight variables are set as illustrated in Fig. 2.4(a) and the value of each area is set $\frac{1}{3}$. We use 80 neighbors of a pixel in a 9×9 patch for non-local regularization and the parameters are $\sigma_D = 40$, $\sigma_I = \frac{25}{255}$, $\beta_s = 10\lambda_s$, $\gamma_s = 1000\lambda_s$.

The framework of our method in Algorithm 2 is designed for gray image restoration. However, the estimated set of blur kernels from a gray image is also used for deblurring the corresponding color image by applying the sharp image restoration step introduced in Sec. 2.3.1 for each color channel.

In Fig. 2.5, the motion blur segmentation and deblurring results of real dynamic scenes are shown. We observe that substantial improvements are achieved in the hair of the running bull and the letter on the bus. Also we compared our results with dynamic scenes to conventional methods. As shown in Fig. 2.6, there are serious artifacts near the boundaries of moving objects in the results of other methods, while our method gives relatively clean results with the aid of various blur models and motion blur segmentations.

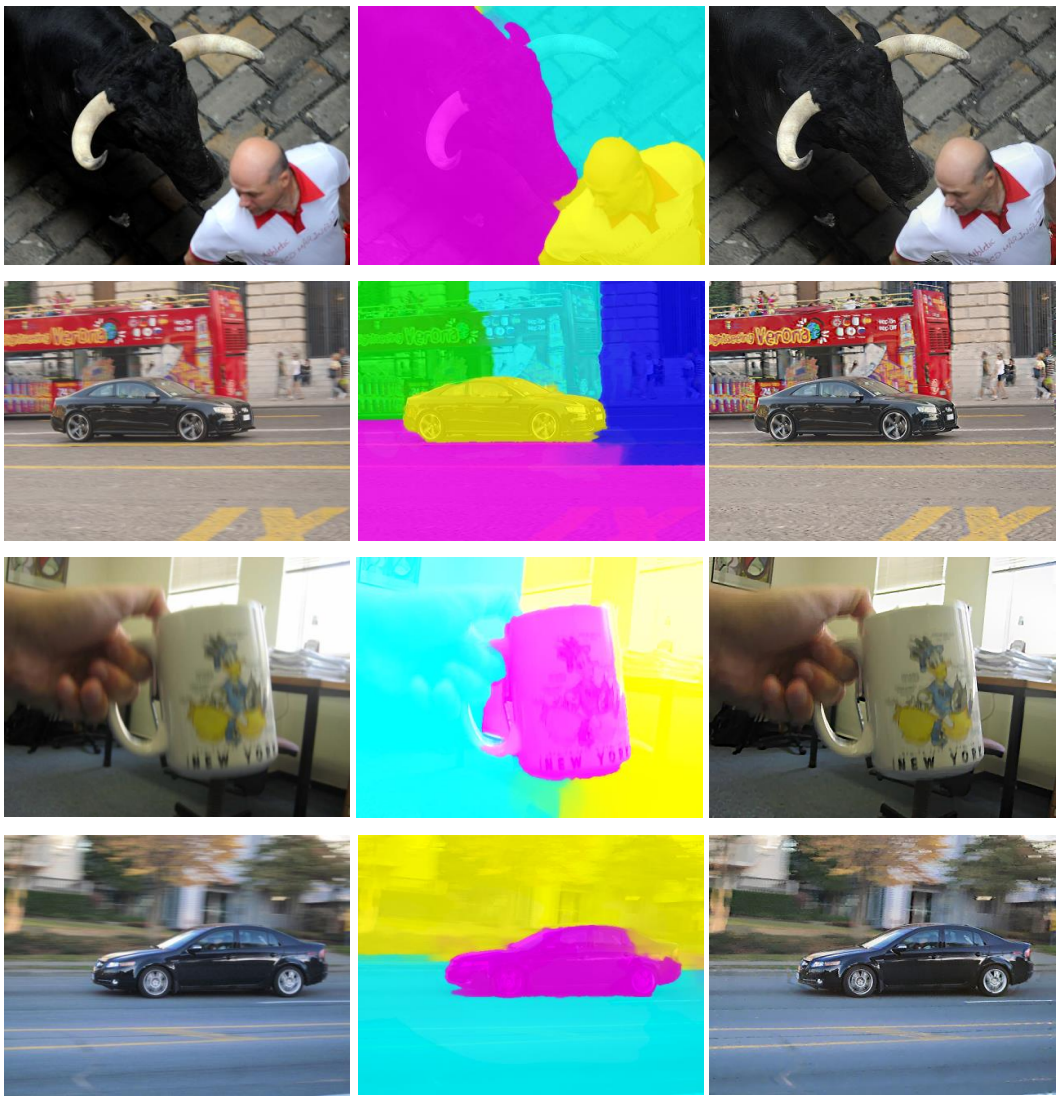
We also compared our deblurring results on static scenes with conventional methods in Fig. 2.7 and Fig. 2.8. The blurry input Picasso image in Fig. 2.7 is degraded

by an uniform kernel. Although it is possible to obtain a sharp image with methods based on uniform kernel, our model shows much better result in reducing ringing artifacts and restoring details well. In addition, the blurry input used in Fig. 2.8 has non-uniform blur motion which is generated by rotational camera shake. In this case, one of seven blur models which corresponds to the non-uniform blur kernel gained almost all weights and still showed competitive result compared to the state-of-the-art non-uniform kernel based methods.

In Fig. 2.9, our deblurring results for real images including defocus blurs are shown. Since the propose method can handle not only motion blurs but also (Gaussian or disc) defocus blurs, ours removes defocus blurs in the background clearly.

2.5 Summary

We proposed a novel single image deblurring framework that can handle multiple moving objects in the scene as well as camera shake. By introducing multiple blur models and their locally varying weight variables which favor the blur models giving better data fidelity, we could also obtain the segmented blur region as well as restored image. We demonstrated the superiority of our method over conventional methods in dynamic scene deblurring as well as in static scene cases. The future challenges and remaining problems are determining the number of moving objects via non-parametric methods. Since the number of moving objects is unknown in a real image, we should set a large number of blur models. Another problem is the run-time of our method. Due to multiple blur models and additional weight variables, computational costs increase. Thus, our future works will include developing an efficient optimization method and parallel implementation using GPGPU.



(a) (b) (c)

Figure 2.5: Deblurring results of real dynamic scenes. (a) Blurry images. (b)-(c) Results from our segmentation-based approach.

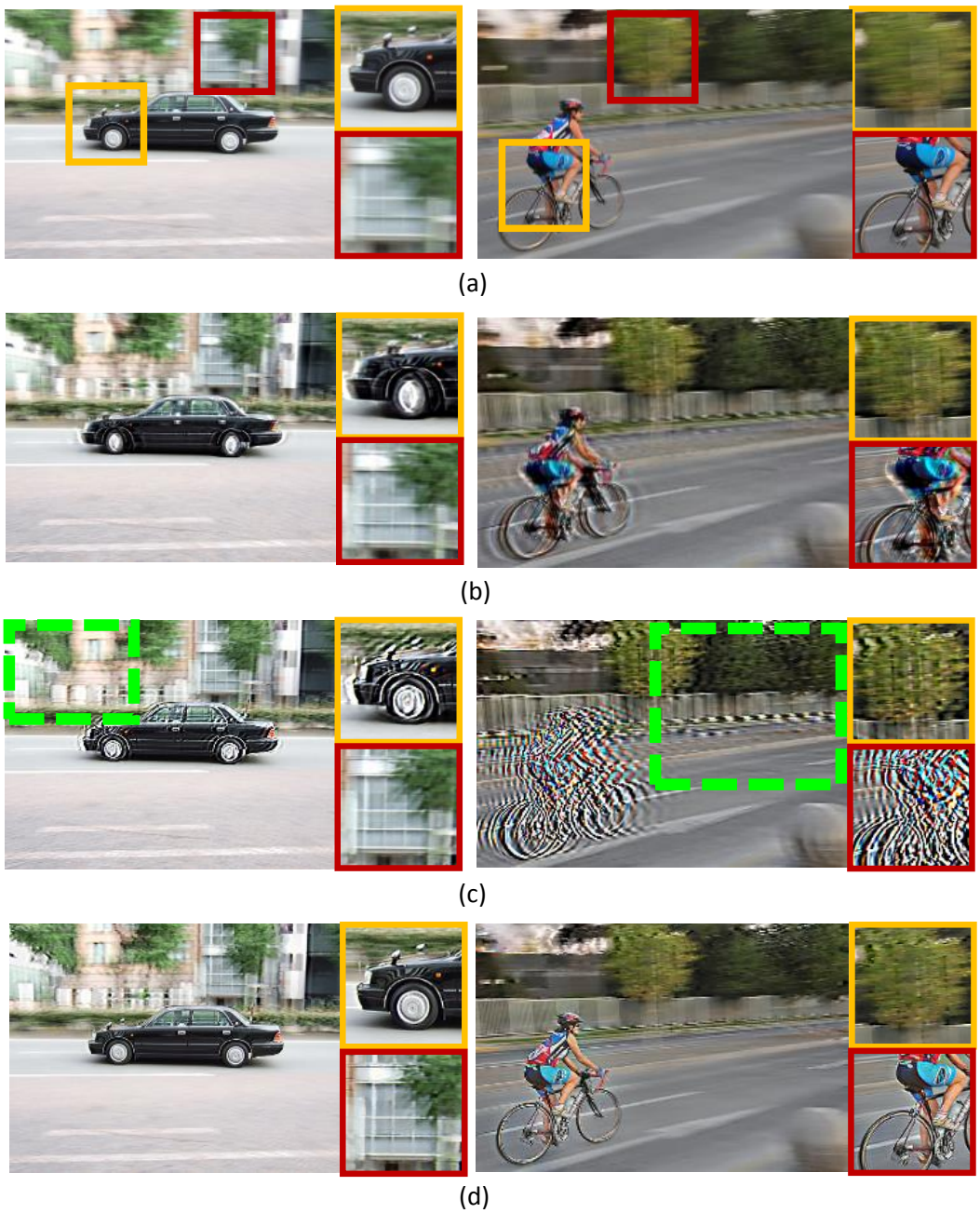


Figure 2.6: Comparison of dynamic scene deblurring results. (a) Blurry images of real dynamic scenes. (b) Deblurring results of Whyte et al. [2]. (c) Deblurring results of Xu et al. [1]. Dashed green boxes in the figures denote the regions used for estimating uniform blur kernels and used for restoring the background regions. (d) Our segmentation-based deblurring results.

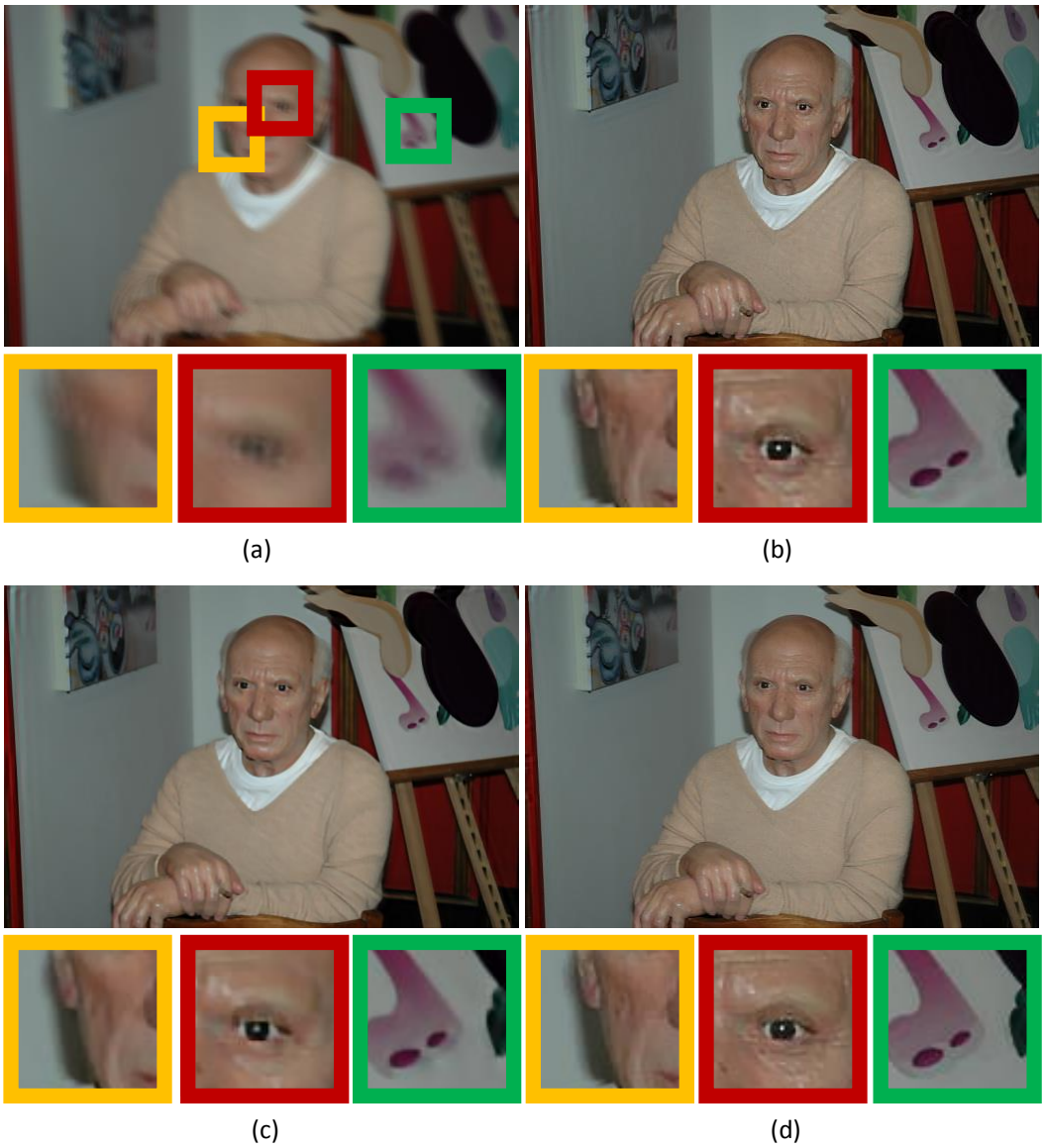


Figure 2.7: Comparison of static scene deblurring. (a) Blurry Picasso image. (b) Result of Shan et al. [3]. (c) Result of Xu et al. [1]. (d) Our segmentation-based deblurring result.

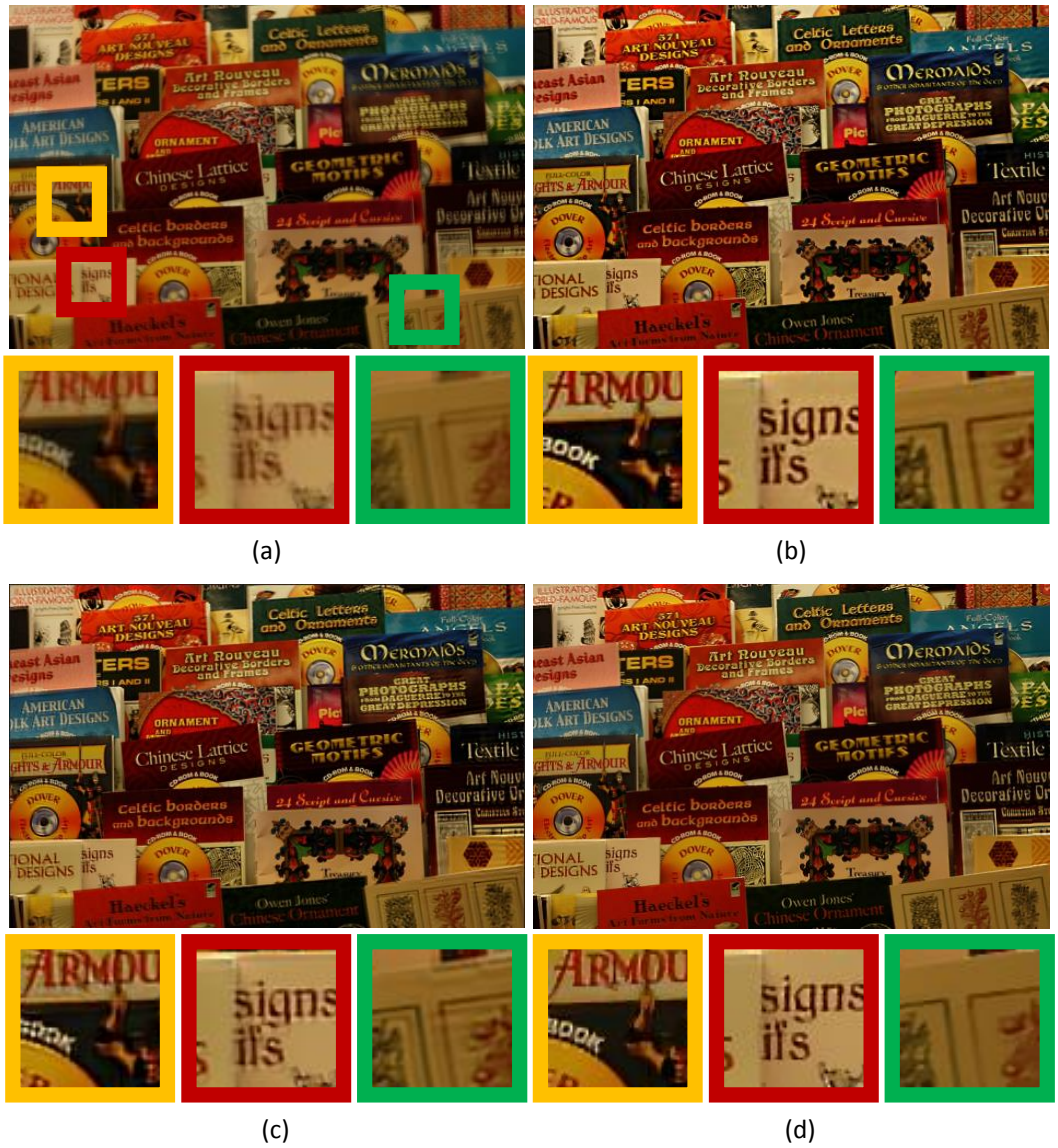


Figure 2.8: Comparison of static scene deblurring. Magazine image is blurred by rotational camera shake and requires non-uniform blur kernel to be restored. (a) Blurry Magazine image. (b) Result of Hirsch et al. [4]. (c) Result of Gupta et al. [5]. (d) Our segmentation-based deblurring result.

Figure 2.9: Other deblurring results. Top to bottom: Real blurry images including defocus blurs and our deblurring results.

Chapter 3

Image Deblurring with Exemplar

In the previous chapter, we introduced a segmentation-based dynamic scene deblurring method that jointly carries out deblurring and segmentation. Although the proposed method achieves good results, the segmentation-based method requires large and texture-rich segments to estimate accurate blur kernels. However, motion blurs in real-world images, especially those captured with a long exposure time or containing fast moving objects, are often large, non-linear and may vary abruptly as indicated by the light streaks in Fig. 3.1. This makes it difficult to estimate accurate blur kernels from a small and texture-less segment, and other state-of-the-art dynamic scene deblurring methods may perform poorly because of poor kernel approximation by a linear blur model or due to inaccurate segmentation.

Therefore, to facilitate more accurate kernel estimation for dynamic scenes, we present a new method that takes as input a sharp exemplar image in addition to the blurred photograph and overcome the limitations of our previous segmentation-

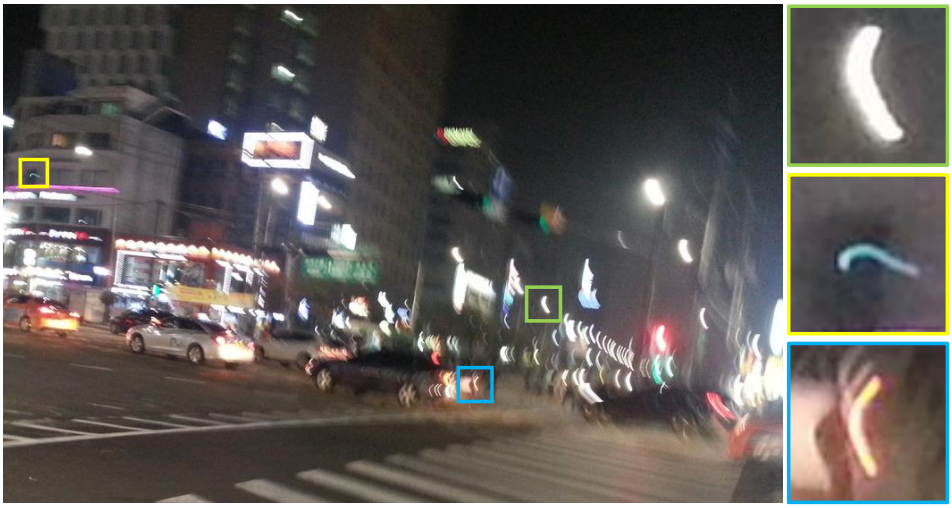


Figure 3.1: A blurry image captured in low light conditions. The light streaks indicate the variety of blur kernel shapes that may co-exist in a dynamic scene.

based deblurring method in this chapter. In practice, such image pairs are readily obtainable from bracket sequences captured by common digital cameras, multi-view image sets, image collections of a given object, and videos that contain lucky sharp frames [12]. The core of our method is a proposed blur-aware flow estimation technique for establishing dense correspondences between the two images. Based on the notion that pixels with similar motion exhibit similar blur, our method identifies and iteratively refines groups of pixels that share a common motion and blur. Initially these groups are determined by clustering of optical flows. Warping the sharp image by these flows produces an exemplar that is used to assist in the estimation of blur kernels, their associated image segments, and the latent image. In estimating these quantities, the pixel groups are adjusted to better fit the estimated blurs, and the resulting latent image is used to refine the optical flow estimates. In this way, the estimated flow determines a sharp exemplar for enhancing deblurring results, while the deblurring leads to refinements in the optical flow. This iterative process results

in significant improvements for spatially-variant motion deblurring.

In contrast to Yuan et al.'s method for a uniformly blurred and non-blurred image pair [20], as well as dynamic scene techniques that assume linear motion blur [33, 9], our method is able to accommodate non-linear and spatially-varying motion blurs. Moreover, the use of an exemplar leads to less sensitivity to segmentation quality in comparison to conventional segmentation-based methods [34, 10], and also improves latent image restoration in small segments that contain less edge information. Similar to the work of HaCohen et al. [35], our method estimates dense correspondences within the deblurring process. However, a major difference is that HaCohen et al.'s method can handle only smooth and slowly varying blurs, while the proposed method can deal with abruptly changing blurs in dynamic scenes through the segmentation of differently blurred regions. In extensive experiments, we show that the proposed method compares favorably to state-of-the-art deblurring methods on challenging dynamic scenes.

3.1 Introduction and Related Work

Photographs captured under low-light conditions are commonly degraded by motion blurs. These blurs may arise from camera shake during the exposure period, and from objects moving in the scene. Even when the lighting conditions are good, there may still be multiple moving objects that produce undesirable blurring effects within an image. This problem has motivated considerable work on image deblurring.

Earlier work focused on removing relatively simple blurs that are uniform and shift-invariant from 2D translational camera shake [16, 3, 17, 1].

In a different approach, a few methods utilize two or more blurry images to

further constrain the deblurring problem [36, 37, 38]. Chen et al. [36] utilized two uniformly blurred images of the same scene to restore a latent frame with the aid of a robust cost function and kernel priors such as sparseness and continuity. Cai et al. [37] proposed a multi-frame deblurring method based on sparseness of both the blur kernels and the latent image. In contrast to these methods for uniform blur, the work of Li et al. [38] handles spatially-varying blur produced from rotational camera movement by estimating a homography between two consecutive images.

Also using additional images are methods for exemplar-based deblurring, which utilize a sharp reference image to improve deblurring results. This was first done by Yuan et al. [20], who restored a blurry image using a sharp but noisy reference image taken with a short exposure. A manual procedure is employed to align the two images. HaCohen et al. [35] later proposed a method that automatically estimates dense correspondences between shared image components in the deblurring process. Recent work by Pan et al. [13] utilized external sharp face exemplars to restore blurry face images. They search for a sharp exemplar among the training samples that corresponds to the given blurry photo, and restore the face image by utilizing salient facial structures obtained from the exemplar.

These previous methods are not designed to handle motion blurs in dynamic scenes, which may contain camera shake, moving objects, and depth variations all in the same image. To deal with dynamic scenes, several methods specifically address spatially-varying blurs from moving objects. Cho et al. [33] segmented an image into multiple regions of homogeneous motions and estimated their corresponding 1D Gaussian blur kernels using multiple images. Moreover, Sun et al. [9] estimated linearly approximated blur kernels using a convolutional neural network (CNN) and recovered the latent image by applying a conventional non-blind deblurring method

with the estimated linear kernels.

3.2 Method Overview

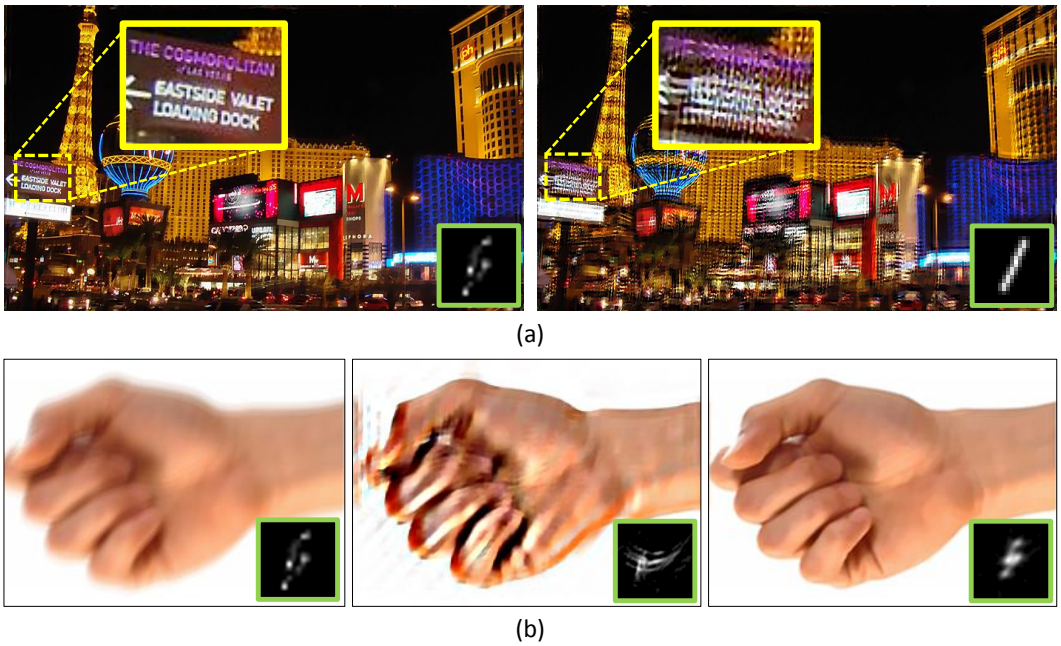


Figure 3.2: Effects of exemplar-based deblurring. (a) From left to right: non-blindly deblurred image with the true non-linear kernel and with an approximated linear kernel. (b) From left to right: blurry image, result of Xu et al. [6], and the result of our exemplar-based deblurring.

The linear kernel approximation used in conventional dynamic scene deblurring methods [34, 9] are efficient ways to handle pixel-wise varying blur kernels, but their kernel approximation errors lead to low-quality results when the kernels are complex and highly non-linear as shown in Fig. 3.2 (a). To render a high-quality image, our deblurring method estimates accurate segment-wise kernels without linear approximation. However, segmentation itself is a difficult problem to solve, and thus we mitigate segmentation errors by utilizing a non-blurred image in this chapter. By

using an exemplar obtained from a non-blurred image, we can estimate blur kernels and the latent image with greater accuracy, even when the image is less textured and lacks strong edges for inferring kernels, as shown in Fig. 3.2 (b).

For exemplar-based deblurring to be effective, dense correspondences between the blurred and non-blurred images are necessary. However, the previously proposed blur-aware optical flow constraint in [15] is not valid when the motions are not translational and the blur varies abruptly as reported in Chapter 4. Therefore, we propose a new optical flow estimation method that handles locally varying blurs to obtain dense correspondences and a useful exemplar. This exemplar is used to improve estimation of spatially-variant blur kernels, their associated uniformly blurred segments, and the latent image.

Estimating these highly related unknowns is difficult to accomplish simultaneously, so we propose an efficient multi-stage approach that solves for the unknowns sequentially as described in the following sections. After identifying a sharp image, the first stage deals with optical flow estimation between the blurred and non-blurred image, and generates an aligned exemplar from the non-blurred image. In the next stage, we deblur the blurry image with the help of the exemplar. Finally, the optical flow is refined using the deblurring estimates. We repeat these steps several times until convergence.

3.3 Stage I: Exemplar Acquisition

3.3.1 Sharp image acquisition and preprocessing

We commonly observe sharp images in bracket sequences and video. For the case of a bracket sequence, a sharp short-exposure image is manually chosen. Although

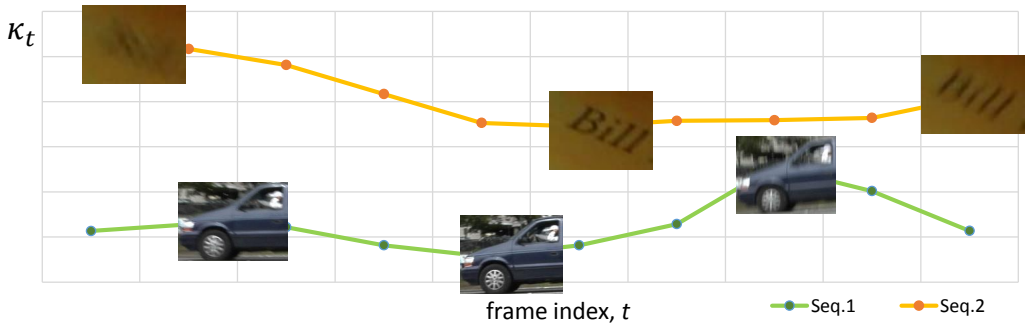


Figure 3.3: Magnitude of motion flows for measuring sharpness.

a short-exposure image is sharp, it may contain high noise and have different color statistics compared to the long-exposure image. Therefore, we apply bilateral filtering with fixed parameters to reduce noise in the sharp reference image as suggested in [28]. Then, we transform the colors of the sharp image to be similar between corresponding points using the color transfer method of Hu et al. [39]. As it is known that applying a non-linear color transfer function to a blurred image causes the blur kernel to vary spatially [40], we adjust the colors of the non-blurred image only.

For a video, manual selection of a sharp frame is less practical. We could choose a lucky sharp image by considering the magnitude of motion [12]. However, that approach is intended for camera shake and may not be as effective for video of a dynamic scene. In Chapter 4, bi-directional optical flows are used to determine the blur kernel in video. Based on this, within a local temporal window (ten frames in our implementation) we choose a sharp frame that has the smallest motion flows according to the following metric:

$$\kappa_t = \sum_{\mathbf{x}} \|\mathbf{u}_{t\to t+1}(\mathbf{x})\| + \|\mathbf{u}_{t\to t-1}(\mathbf{x})\|, \tag{3.1}$$

where t is the frame index, and $\mathbf{u}_{t\to t+1}$ and $\mathbf{u}_{t\to t-1}$ denote forward and backward

optical flows respectively. Since it is difficult to estimate accurate optical flows among blurry frames, we downsize the images to suppress blur and estimate optical flows using a fast method [41]. The effectiveness of this metric in measuring sharpness is shown for two examples in Fig. 3.3.

3.3.2 Exemplar from blur-aware optical flow estimation

To acquire an accurately aligned exemplar from the given non-blurred image in Sec. 3.3.1, we need to estimate dense correspondences. In the work of Portz et al. [15], a method for optical flow estimation between blurry images was proposed based on a commutative law of shift-invariant kernels, where the brightness of corresponding points becomes constant after convolving the blur kernel of each image to the other image. However, the commutative law does not hold when the motion is not translational and the blur varies abruptly. So we present a new method that estimates flows for the case of spatially-varying kernels. In our flow estimation, the key idea is that pixels which have similar motions are assumed to be blurred similarly. Based on this, we estimate locally rigid transformations (homographies) of uniformly blurred segments. Specifically, the proposed method locally (pixel-wise) chooses a homography/kernel pair among multiple candidates. This is solved as a discrete labeling problem.

To obtain candidate homography/kernel pairs, we need several initialization steps. First, we calculate the initial optical flow by applying a conventional method [7] that can handle large displacements. We cluster the initial flow into sets of similar translational motion by applying the k-means algorithm, with k=50 and with small clusters less than 1000 pixels in size removed. For the resulting corresponding segments, we estimate an initial homography using RANSAC, and also a kernel using

the method of [17]. The proposed discrete labeling model using these candidates is formulated as:

$$E(\mathbf{W^S}) = \sum_n \sum_{\mathbf{x}} \mathbf{W}_n^{\mathbf{S}}(\mathbf{x}) \cdot \min(|\mathbf{K}_n^{\mathbf{S}}\mathbf{S}(\mathbf{x}) - \mathbf{B}(\mathbf{x} + \mathbf{u}_n(\mathbf{x}))|, \tau) + \alpha_e \sum_n |\nabla \mathbf{W}_n^{\mathbf{S}}| + \beta_e \sum_{(n,m)} \sum_{(\mathbf{x},\mathbf{y})} G(\mathbf{x}) \mathbf{W}_n^{\mathbf{S}}(\mathbf{x}) \mathbf{W}_m^{\mathbf{S}}(\mathbf{y}) |\mathbf{u}_n(\mathbf{x}) - \mathbf{u}_m(\mathbf{y})|, \tag{3.2}$$

where $\mathbf{x}$ denotes the pixel location, and $\mathbf{y}$ indexes the four neighboring pixels of $\mathbf{x}$. Non-blurred and blurred images are $\mathbf{S}$ and $\mathbf{B}$ respectively. In addition, $\mathbf{W^S} = \{\mathbf{W}_1^{\mathbf{S}}, \ldots, \mathbf{W}_N^{\mathbf{S}}\}$ denotes a set of N binary indicators with pixel-wise constraints such that $\sum_{n=1}^{N} \mathbf{W}_n^{\mathbf{S}}(\mathbf{x}) = 1$ and $\mathbf{W}_n^{\mathbf{S}}(\mathbf{x}) \in \{0, 1\}$. The candidate kernel matrix is $\mathbf{K}_n^{\mathbf{S}}$, and motion $\mathbf{u}_n$ is induced by the associated n^{th} candidate homography. An edge map is given by $G(\mathbf{x}) = \exp(-\frac{(\nabla \mathbf{S}(\mathbf{x}))^2}{\sigma_G})$, where $\sigma_G = 15$ is a fixed parameter. Parameters α_e and β_e control the weights of the associated terms, and τ represents a threshold for robustness against outliers.

In the proposed energy model, the first term denotes a new data term based on an optical flow constraint that assumes constant brightness between the generated blurry image and real blurry image. The homography/kernel pair that gives a lower data cost is favored in the proposed data term. As these are selected locally, spatially-varying blurs can be handled. The second term represents the Potts model for enforcing smoothness, in this case, similar labels among neighboring pixels. The last term enforces optical flow to have similar values among neighboring pixels, and is coupled with an edge map to allow discontinuities.

We solve the discrete energy minimization problem in (3.2) using the graph-cut algorithm [42]. After discrete optimization, we can determine optical flow by assigning the locally chosen motion model as $\mathbf{u}(\mathbf{x}) = \sum_n \mathbf{W}_n^{\mathbf{S}}(\mathbf{x}) \mathbf{u}_n(\mathbf{x})$, and obtain the

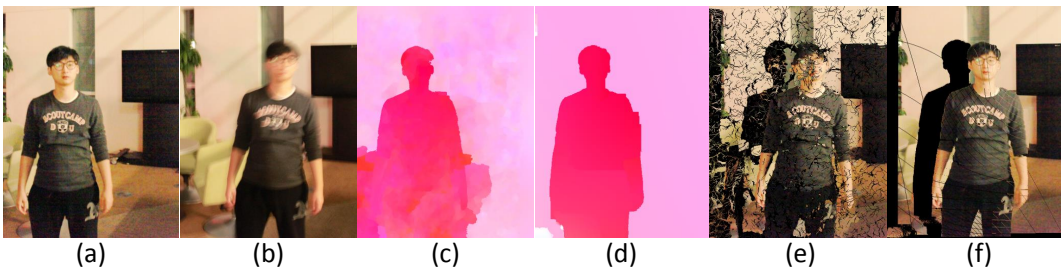


Figure 3.4: Comparison between intermediate and final results. (a) Input non-blurred image after color correction. (b) Input blurry image. (c) Color coded initial flow from Xu et al. [7]. (d) Our color coded blur-aware flow. (e) Exemplar from initial flow $\mathbf{u}$. (f) Exemplar from blur-aware flow $\mathbf{u}$.

sharp exemplar $\bar{\mathbf{S}}$ by warping the non-blurred image $\mathbf{S}$ backward with the obtained flow $\mathbf{u}$. In Fig. 3.4, we demonstrate that our blur-aware technique estimates accurate optical flows in the presence of spatially-varying blurs, and preserves flow discontinuities at the boundaries. As a result, the exemplar obtained from our blur-aware flow is well aligned with the input blurry image as shown in Fig. 3.4 (e)-(f).

3.4 Stage II: Exemplar-based Deblurring

In the second stage, the other unknowns are estimated with the help of the sharp exemplar and motion flow obtained in the previous stage. As in [10], we estimate on the blurry image $\mathbf{B}$ a set of uniform blur kernels $\mathbf{K}^{\mathbf{B}}$, their associated segments $\mathbf{W}^{\mathbf{B}}$, and the latent image $\mathbf{L}$. Specifically, the blur kernels are represented by sparse kernel matrices $\mathbf{K}^{\mathbf{B}} = \{\mathbf{K}^{\mathbf{B}}_1, \ldots, \mathbf{K}^{\mathbf{B}}_N\}$, and $\mathbf{W}^{\mathbf{B}} = \{\mathbf{W}^{\mathbf{B}}_1, \ldots, \mathbf{W}^{\mathbf{B}}_N\}$ denotes a set of binary maps where $\sum_n \mathbf{W}^{\mathbf{B}}_n(\mathbf{x}) = 1$ and $\mathbf{W}^{\mathbf{B}}_n(\mathbf{x}) \in \{0, 1\}$. In addition, to handle occluded or invalid pixels in our model, we define an occlusion state indicator $\mathbf{o}(\mathbf{x}) \in \{0, 1\}$, and use it in the proposed model. If the corresponding point of a pixel at $\mathbf{x}$ is occluded [43] or saturated [39], $\mathbf{o}(\mathbf{x}) = 0$. Otherwise, $\mathbf{o}(\mathbf{x}) = 1$.

The energy models for $\mathbf{L}$, $\mathbf{W}^{\mathbf{B}}$ and $\mathbf{K}^{\mathbf{B}}$ are detailed in the following subsections.

3.4.1 Exemplar-based latent image restoration

Our image restoration step is similar to the technique proposed in [10] except for the incorporation of an exemplar term. With the use of an exemplar $\bar{\mathbf{S}}$, several problems such as ringing artifacts and saturation due to inaccurately estimated kernels can be addressed. The formulation of this step, with kernel matrices $\mathbf{K}^{\mathbf{B}}$ and segments $\mathbf{W}^{\mathbf{B}}$ fixed, is given by

$$E(\mathbf{L}) = \sum_{\mathbf{x}} |\nabla \mathbf{L}(\mathbf{x})| + \lambda_e \sum_{n} \sum_{\partial *} \sum_{\mathbf{x}} \mathbf{W}_n^{\mathbf{B}}(\mathbf{x})(\partial_* \mathbf{K}_n^{\mathbf{B}} \mathbf{L}(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x}))^2 + \zeta_e \sum_{\mathbf{x}} \mathbf{o}(\mathbf{x})(\mathbf{L}(\mathbf{x}) - \bar{\mathbf{S}}(\mathbf{x}))^2, \tag{3.3}$$

where $\partial_* \in \{\partial_o, \partial_x, \partial_y\}$ denotes linear operators (identity matrix, horizontal derivative, and vertical derivative). Parameters λ_e and ζ_e control the weights of the coupled terms.

The first term represents a prior for the latent image. We adopt a total variation (TV) regularization that not only enforces sparsity on the latent image but also allows for simple computation. The second term is an adaptive blur model based on locally chosen blur kernels. The last term is the key term in the proposed exemplar-based deblurring method. It favors copying the intensities and structures of the reference image onto the corresponding pixels of the latent image. In areas of the blurry image that are occluded or saturated in the reference image, this last term drops out and the deblurring becomes a standard deconvolution with priors on the latent image and kernel.

Computationally, the cost of minimizing (3.3) is higher when the number of labels N is large. Thus, a large, sparse kernel matrix $\mathbf{K}$ where the m^{th} row corresponds to

the chosen kernel at the m^{th} pixel is used in approximating (3.3):

$$E(\mathbf{L}) \approx \sum_{\mathbf{x}} |\nabla \mathbf{L}(\mathbf{x})| + \lambda_e \sum_{\partial *} \|\partial_* \mathbf{K}\mathbf{L} - \partial_* \mathbf{B}\|^2 + \zeta_e \sum_{\mathbf{x}} \mathbf{o}(\mathbf{x})(\mathbf{L}(\mathbf{x}) - \bar{\mathbf{S}}(\mathbf{x}))^2. \quad (3.4)$$

This energy model with respect to latent image $\mathbf{L}$ is convex, and thus any conventional convex optimization technique can be used. We adopt the first-order primal-dual algorithm [29] to estimate the latent image $\mathbf{L}$, since it converges quickly.

3.4.2 Motion-aware segmentation

In the proposed method, multiple uniform blur kernels are estimated to restore the latent image; thus, uniformly blurred regions should be identified while the latent image $\mathbf{L}$ and kernel matrices $\mathbf{K}^{\mathbf{B}}$ are fixed. We adopted a non-local smoothness term in Chapter 2, but we found it to be inefficient when the size of the segment is small or when the intensities at the boundaries are not distinct.

To address this problem, we utilize parametric motion (homography) information extracted during the first stage by assuming that pixels with similar motions are blurred similarly. Our motion-aware segmentation model is given by

$$\begin{aligned} E(\mathbf{W}^{\mathbf{B}}) =& \lambda_e \sum_n \sum_{\partial *} \sum_{\mathbf{x}} \mathbf{W}_n^{\mathbf{B}}(\mathbf{x})(\partial_* \mathbf{K}_n^{\mathbf{B}} \mathbf{L}(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x}))^2 + \\ & \lambda_e \sum_n \sum_{\partial *} \sum_{\mathbf{x}} \mathbf{W}_n^{\mathbf{B}}(\mathbf{x})\mathbf{o}(\mathbf{x})(\partial_* \mathbf{K}_n^{\mathbf{B}} \bar{\mathbf{S}}_n(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x}))^2 + \\ & \mu_e \sum_n \sum_{\mathbf{x}} |\nabla \mathbf{W}_n^{\mathbf{B}}(\mathbf{x})| + \eta_e \sum_{(n,m)} \sum_{(\mathbf{x},\mathbf{y})} T(\mathbf{x}) \mathbf{W}_n^{\mathbf{B}}(\mathbf{x}) \mathbf{W}_m^{\mathbf{B}}(\mathbf{y}) |\bar{\mathbf{u}}_n(\mathbf{x}) - \bar{\mathbf{u}}_m(\mathbf{y})|, \end{aligned} \quad (3.5)$$

where $\bar{\mathbf{u}}_n$ is induced by the inverse of the n^{th} homography model from the first stage,

and the warped non-blurred image by $\bar{\mathbf{u}}_n$ is $\bar{\mathbf{S}}_n(\mathbf{x}) = \mathbf{S}(\mathbf{x}+\bar{\mathbf{u}}_n(\mathbf{x}))$. Parameters μ_e and η_e control the weights of the coupled terms. An edge map to allow discontinuities near edges is expressed as $T(\mathbf{x}) = \exp(-\frac{(\nabla \mathbf{L}_0(\mathbf{x}))^2}{\sigma_T})$, where $\sigma_T = 15$ is a fixed parameter and $\mathbf{L}_0$ denotes an initial latent image in the iterative optimization framework.

In the first and second terms, our energy model chooses the locally (pixel-wise) best-fit model by measuring the data fidelities, and $\mathbf{W}_n^{\mathbf{B}}(\mathbf{x}) = 1$ is assigned when the cost of the associated blur model is low. Thus, $\mathbf{W}_n^{\mathbf{B}}$ represents the proper blur model from among the candidate models. In the second term, the parametric motion model is associated with the segmentation task; this term helps to extract segments that are not only uniformly blurred but also moving rigidly. The third and last terms are pairwise terms. The third term is the Potts model, and the last term enforces motions to have similar values. As motion information is utilized in the second and last terms, the proposed segmentation model is motion-aware. To minimize (3.5), we use the graph-cut algorithm [42].

3.4.3 Robust kernel estimation

In conventional blind deblurring methods, robust kernel estimation requires strong edge information in the latent image [17, 1, 6] to accurately estimate blur kernels in less-textured and mid-frequency regions. Unlike conventional single-image deblurring methods, a sharp image is employed in our formulation, which can compensate for a lack of strong edges. The latent image $\mathbf{L}$ and segments $\mathbf{W}^{\mathbf{B}}$ are fixed to estimate

blur kernels:

$$E(\mathbf{K}^{\mathbf{B}}) = \gamma_e \sum_n \|\mathbf{K}_n^{\mathbf{B}}\|^2 + \lambda_e \sum_n \sum_{\partial *} \sum_{\mathbf{x}} \mathbf{W}_n^{\mathbf{B}}(\mathbf{x})(\partial_* \mathbf{K}_n^{\mathbf{B}} \mathbf{L}(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x}))^2 + \lambda_e \sum_n \sum_{\partial *} \sum_{\mathbf{x}} \mathbf{W}_n^{\mathbf{B}}(\mathbf{x})\mathbf{o}(\mathbf{x})(\partial_* \mathbf{K}_n^{\mathbf{B}} \bar{\mathbf{S}}_n(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x}))^2, \tag{3.6}$$

where $\mathbf{K}_n^{\mathbf{B}}$ is a vector representation of the n^{th} uniform kernel and can be represented as a sparse blur kernel matrix $\mathbf{K}_n^{\mathbf{B}}$. Parameters γ_e and λ_e control the weights of their associated terms. Note that warped non-blurred image $\bar{\mathbf{S}}_n$ using the n^{th} homography model is defined in Sec. 3.4.2.

The first term denotes the Tikhonov regularization used in [17, 20, 10]. It regularizes the kernel and allows for fast optimization. The second term is the adaptive blur model proposed in Chapter 2, which helps to estimate $\mathbf{K}_n^{\mathbf{B}}$ within a segment where $\mathbf{W}_n^{\mathbf{B}}(\mathbf{x}) = 1$. The last term is a new term that utilizes the non-blurred image in kernel estimation. Notably, our model estimates the kernels more robustly even when $\mathbf{L}$ has less edge information, by taking advantage of the sharp image in the last term as shown in Fig. 3.2 (b).

Since the proposed kernel model is convex with respect to $\mathbf{K}_n^{\mathbf{B}}$, we can estimate the segment-wise kernels rapidly by applying a slight variant of the fast kernel estimation method used in [17]. In practice, we use an edge-enhanced and noise-reduced latent image $\mathbf{L}$, also known as a predicted gradient map [17], to facilitate accurate kernel estimation and fast convergence.

3.4.4 Unified energy model and optimization

The aforementioned energy models can be integrated into a single unified objective function, given by

$$E(\mathbf{L},\mathbf{K^B},\mathbf{W^B}) = \sum_{\mathbf{x}} |\nabla \mathbf{L}(\mathbf{x})| + \zeta_e \sum_{\mathbf{x}} \mathbf{o}(\mathbf{x}) \|\mathbf{L}(\mathbf{x}) - \bar{\mathbf{S}}(\mathbf{x})\|^2 + \gamma_e \sum_{n} \|\mathbf{K}_n^{\mathbf{B}}\|^2 +$$
$$\mu_e \sum_{n} \sum_{\mathbf{x}} |\nabla \mathbf{W}_n^{\mathbf{B}}(\mathbf{x})| + \eta_e \sum_{(n,m)} \sum_{(\mathbf{x},\mathbf{y})} T(\mathbf{x}) \mathbf{W}_n^{\mathbf{B}}(\mathbf{x}) \mathbf{W}_m^{\mathbf{B}}(\mathbf{y}) |\bar{\mathbf{u}}_n(\mathbf{x}) - \bar{\mathbf{u}}_m(\mathbf{y})| +$$
$$\lambda_e \sum_{n} \sum_{\partial *} \sum_{\mathbf{x}} \mathbf{W}_n^{\mathbf{B}}(\mathbf{x}) ((\partial_* \mathbf{K}_n^{\mathbf{B}} \mathbf{L}(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x}))^2 + \mathbf{o}(\mathbf{x}) (\partial_* \mathbf{K}_n^{\mathbf{B}} \bar{\mathbf{S}}_n(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x}))^2). \tag{3.7}$$

As this energy function is non-convex, the proposed energy model is minimized through an iterative alternating optimization by solving for one of the variables while fixing the others. Specifically, we minimize (3.4), (3.5), and (3.6) in an iterative manner as in [10]. Although the alternating optimization method cannot guarantee a globally optimal solution, each sub-problem can converge to a minimum.

3.5 Stage III: Post-processing and Refinement

Since the optical flow initially estimated with a conventional method and the exemplar computed from it likely contain significant errors, they are refined using the deblurring estimates from the previous stage. In our method, the optical flow is updated at the original image scale, with the current flow used for initialization. The updating model is based on total generalized variation regularization [44] to avoid a piecewise constant solution that induces fronto-parallel artifacts and yields piecewise

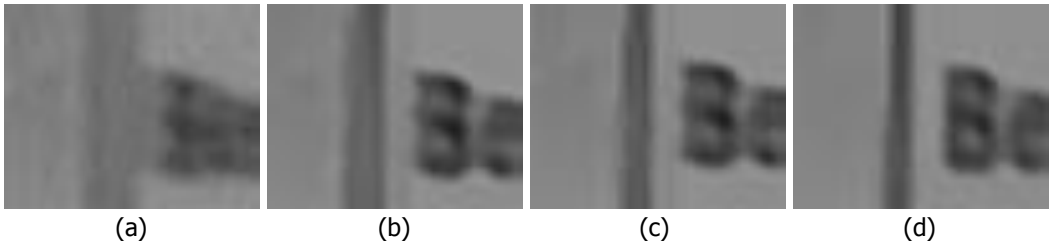


Figure 3.5: Effect of iterations on deblurring. (a) A patch of the blurry input image. After (b) 1st iteration, (c) 2nd iteration, and (d) 3rd iteration.

affine motions. The formulation of this stage is given by

$$\min_{\mathbf{u},\mathbf{v}} \sum_{\mathbf{x}} G(\mathbf{x}) \cdot (|\nabla \mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})| + \omega_1 |\nabla \mathbf{v}(\mathbf{x})|) + \omega_2 \cdot \psi(\mathbf{S}(\mathbf{x}), \mathbf{L}(\mathbf{x} + \mathbf{u})), \tag{3.8}$$

where the auxiliary vector $\mathbf{v}$ leads the second-order derivative of the flow $\mathbf{u}$ to be constant. Parameters $\omega_1 = 5$ and $\omega_2 = 2$ control the weights of the second and last terms. The data fidelity function ψ initially computes the 5x5 Ternary Census transforms of two patches centered at $\mathbf{S}(\mathbf{x})$ and $\mathbf{L}(\mathbf{x} + \mathbf{u}(\mathbf{x}))$, and subsequently measures the Hamming distance between the two patches. As the data function is non-convex, the optimization technique used in [45], which transforms the original non-convex function into a piecewise convex function, is adopted. The updated occlusion state $\mathbf{o}(\mathbf{x})$ and optical flow are then used to refine the exemplar and parametric motion models (homographies) with RANSAC, and the latent image is restored gradually by iterating the deblurring and refinement processes until convergence, as shown in Fig. 3.5.

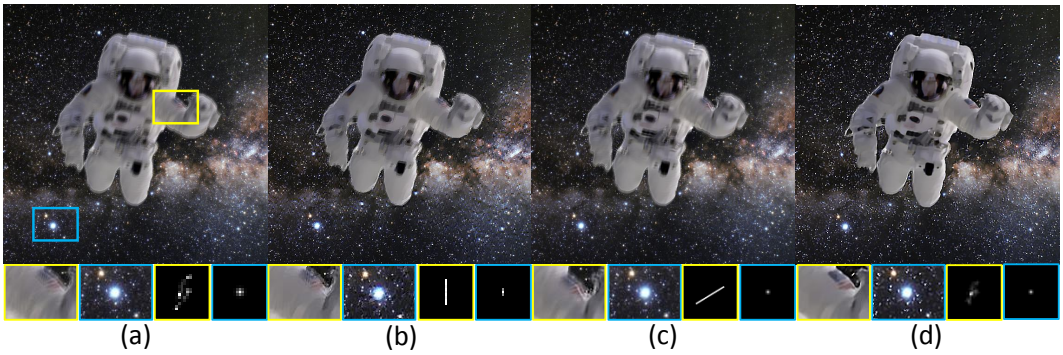


Figure 3.6: Comparison with other dynamic scene deblurring methods. (a) A synthetically blurred image and true kernels. (b) Result by [8]. (c) Result by [9]. (d) Our exemplar-based deblurred image and kernels.

3.6 Experiments

Our method was implemented with empirically set parameters of $\alpha_e = 30$, $\beta_e = 15$, $\lambda_e = 100$, $\zeta_e = 0.2 * \lambda_e$, $\gamma_e = 50/\lambda_e$, $\mu_e = 10$, and $\eta_e = 10$ for most of our experiments. When the noise of a blurred image is high, we choose smaller λ_e to reduce noise in the latent image. The exemplar-based deblurring of Sec. 3.4 and the refinement step of Sec. 3.5 are iterated three times with grayscale inputs. Finally, we restore the latent color image by minimizing (3.4) for each color channel. The exemplar-based deblurring of Sec. 3.4 is conducted in a coarse-to-fine manner to handle large blurs and promote fast convergence.

First, we compare our method with the segmentation-based deblurring method in Chapter 2. For evaluation, several blurred and non-blurred image pairs from synthetic datasets generated by the method in 5.5 are used, and PSNR values are compared in Table. 3.1. As expected, the exemplar-based method outperforms segmentation-based method.

Moreover, we evaluate on five blurred and non-blurred image pairs from the

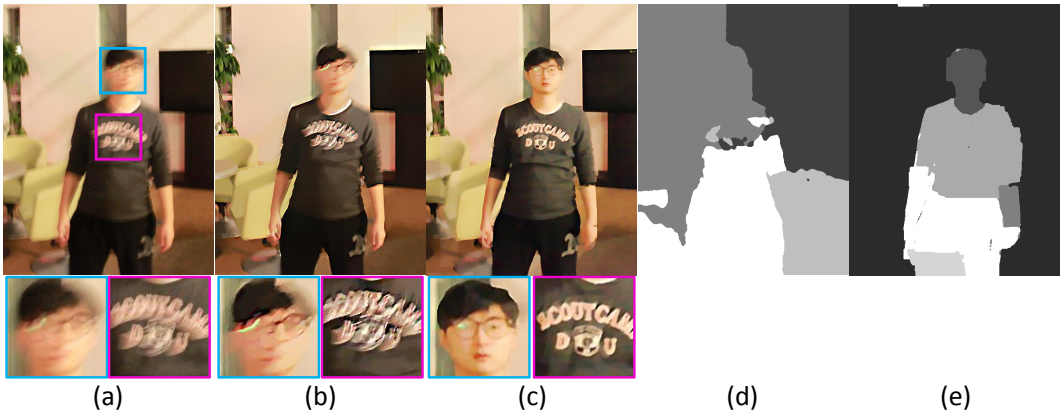


Figure 3.7: Comparison with previous segmentation-based method. (a) A real blurry image. (b) Segmentation-based deblurring result [10]. (c) Our exemplar-based deblurring result. (d) Segmentation from our previous segmentation-based method [10]. (e) Our exemplar-based segmentation result.

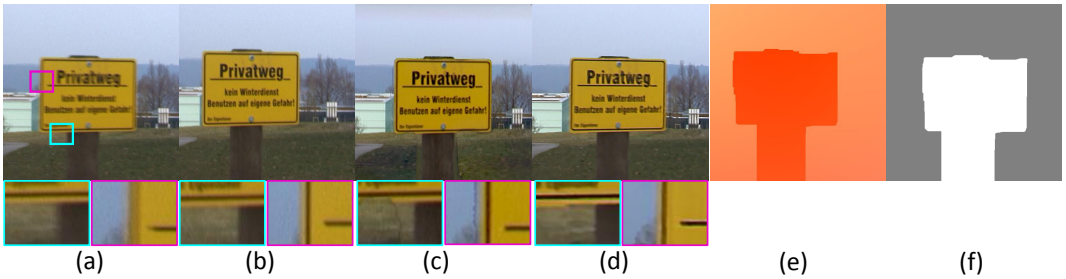


Figure 3.8: Comparison with segmentation-based video deblurring method. (a) A blurry image. (b) An automatically chosen sharp image. (c) Deblurring result from [11]. (d) Our exemplar-based deblurring result. (e) Our color-coded flow, $\mathbf{u}$. (f) Our segmentation result.

SINTEL dataset [46] to measure flow accuracy. As shown in Table. 3.2, our exemplar-based method achieves about a 9 percent gain in average end point error (EPE) over the initial flow [7], and significantly outperforms the blur-aware flow method [15]. Notably, atmospheric effects (e.g. fog) are added in the SINTEL blurry images, we do not provide deblurring results with SINTEL dataset.

Next, we qualitatively compare our method with approximation-based non-uniform deblurring methods [8, 9] in Fig. 3.6. Since local kernels are modeled as linear in [8, 9],

Seq.	Segmentation-based	Exemplar-based
#1	20.25	26.48
#2	23.23	28.09
#3	22.46	26.22
#4	22.47	25.69
#5	22.20	23.26
#6	21.56	24.68
#7	23.54	25.73
#8	20.69	25.30
#9	20.35	25.03
#10	20.84	24.02
Avg.	21.76	**25.45**

Table 3.1: Quantitative comparisons with the previous segmentation-based method. PSNR values are used for evaluating deblurring performance.

Xu et al. [7]	Portz et al. [15]	Exemplar-based
23.65	33.73	**21.61**

Table 3.2: Quantitative comparisons with the methods of Xu et al. [7] and Portz et al. [15]. EPE values are used for evaluating optical flow accuracies.

they are unable to recover non-linear blurs, leading to severe artifacts. By contrast, our method estimates different non-linear kernels accurately and restores the latent image well.

In Fig. 3.7, we compare the proposed exemplar-based deblurring method with our previous segmentation-based dynamic scene deblurring method in [10]. The real blurry image contains spatially-varying blur from camera shake and a moving person. In particular, the person moves rotationally, so the head, upper body, and lower body move differently. Therefore the size of each uniformly blurred region is rela-

tively small in the body. Since the method of [10] requires large segments to infer accurate kernels, it fails to extract uniformly blurred small segments. Unlike [10], our method utilizes an additional sharp reference image, and the proposed motion-aware segmentation makes it possible to segment uniformly blurred small segments. Our method restores the moving person much better as a result, as seen from the person's face and the letters on the shirt.

We also compare with the state-of-the-art segmentation-based video deblurring method [11] in Fig. 3.8. For this comparison, we use a sharp and blurry image pair chosen automatically from the dataset used in [11]. Though the chosen sharp image, which is the sharpest frame available, is not especially clear, our method nevertheless successfully restores the latent image. Unlike our method, the technique in [11] uses more than ten frames to restore the same latent image, so our deblurring result is a bit less clear than that of [11]. On the other hand, the method of [11] fails to infer accurate segments at motion boundaries, and thus generates severe artifacts in comparison to our result. Notice that our segmentation result in Fig. 3.8 (c) is very similar to the color-coded optical flow in Fig. 3.8 (b), since our segmentation relies on motion information in (3.5).

Next in Fig. 3.9, we compare with a real blurry image whose blurs are caused by multiple moving objects as well as camera shake. Our technique outperforms single-image deblurring methods designed to handle spatially varying blur, whose results are in Fig. 3.9 (b)-(d). Since the source code of [47], which uses image pairs, is not available, we compare in Fig. 3.9 (e) with the exemplar-based video deblurring method of [12], which uses more than 50 consecutive images. Even though we use only one image pair, our method restores the moving objects in Fig. 3.9 better, and exhibits competitive deblurring results on the static background. Notably, our

method produces better results in mid-frequency regions like the leaves, because the proposed method is based on deconvolution, unlike the interpolation from multiple frames in [12] which leads to smoothed results.

In Fig. 3.10, we show that promising results are obtained by applying our deblurring method to challenging blurry images. In general, moving people generate spatially-varying blurs, and restoration of people is an important problem. The results demonstrate successful restoration of the moving people and removal of spatially varying blurs in the images.

3.7 Summary

In this study, we introduced a novel exemplar-based dynamic scene deblurring method that effectively handles non-uniform blurs with the use of a blurred and non-blurred image pair. By estimating dense correspondences between the input images, our method obtains accurately aligned exemplars which help in removing spatially-varying blurs. We provided an efficient multi-stage framework to estimate multiple unknowns and achieved significant deblurring improvements in comparison to the state-of-the-art methods on challenging dynamic scenes.

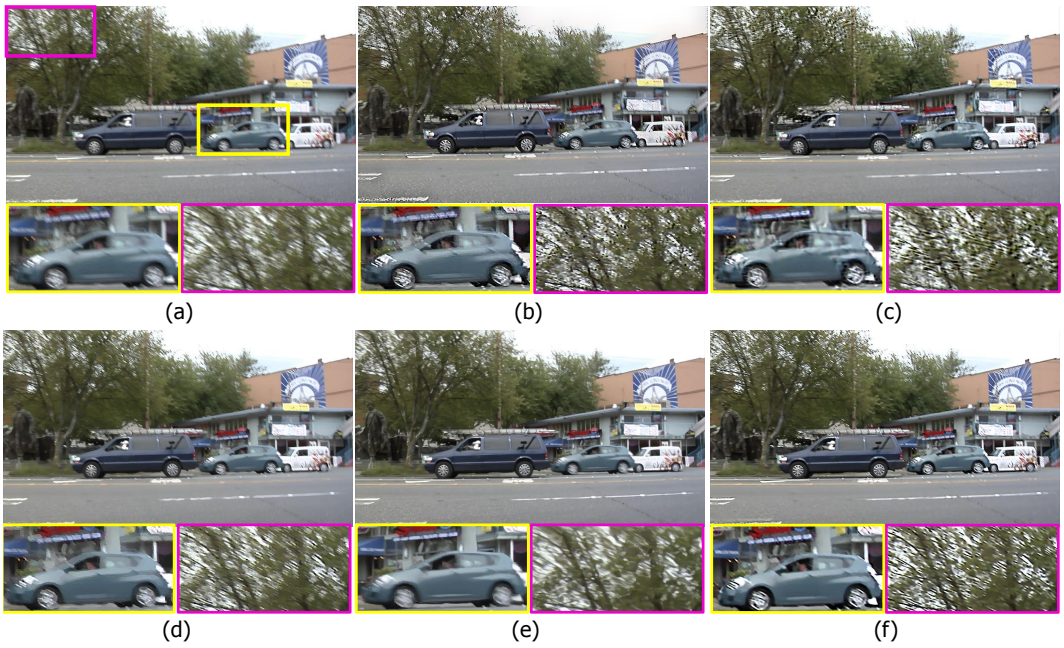


Figure 3.9: Qualitative comparisons. (a) A real blurry image containing moving objects. (b) Result from the segmentation-based method[10]. (c) Result from [9]. (d) Result from [6]. (e) Result from [12]. (f) Our exemplar-based deblurring result.

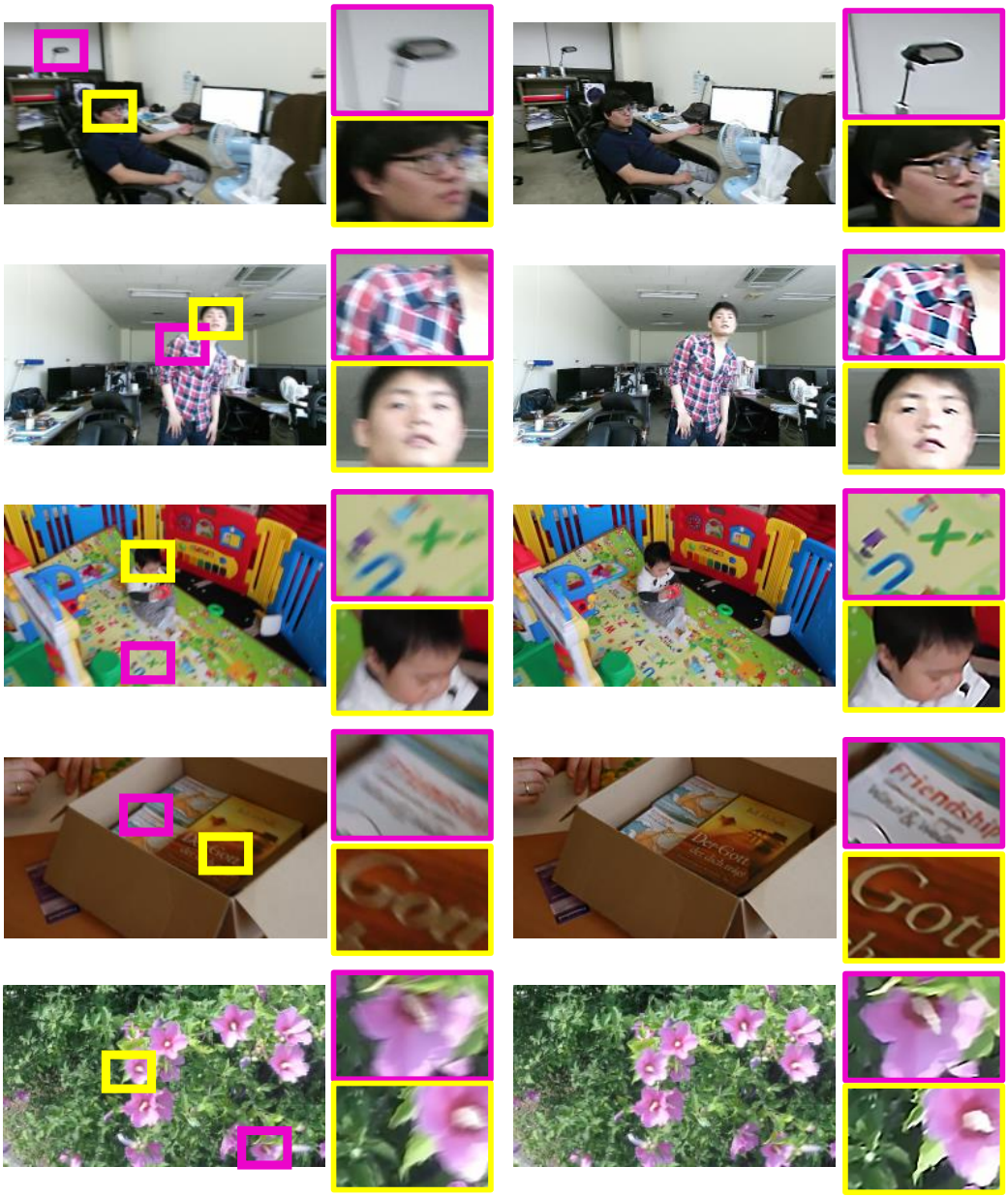


Figure 3.10: Other deblurring results on challenging blurry images. Left to right: Blurry images and deblurring results.

Chapter 4

Image Deblurring with Kernel-Parametrization

In the previous chapters, segmentation-based and exemplar-based dynamic scene deblurring approaches were introduced. Both approaches assume that blurs are caused by segment-wise rigid motions, and thus estimate segment-wise different kernels. Although the previous approaches can handle segment-wise varying blurs caused by segment-wise rigid motions, but they restrict the types of blur kernels which can be removed. Therefore these approaches cannot remove pixel-wise blurs by translational motion within a segment, but cannot handle pixel-wise varying blurs caused by the non-rigid motion of moving objects.

To sum up, our previous dynamic scene deblurring methods require accurate motion segmentation for specific types of motion blur. To mitigate this restriction, in this chapter, we propose a deblurring method using kernel-parametrization to deblur dynamic scenes without segmentation and restricting the types of motion blur, when the locally varying blur kernels can be approximated to 2D motion vectors.

Although this restriction excludes non-linear motion, numerous types of motions can be linearized in practical situations [34, 48, 21]. We observed that this assumption holds for many real blurry images and that this approach works even when small rotational camera shake exists, as shown in our experiments.

In the previous work of Dai et al. [48], the authors estimated motion flow via the alpha channel [49] of the blurry image. However, they used a constraint that is different from ours and applied implicit segmentation based on RANSAC. In addition, the result depends on the accuracy of the given alpha channel. By contrast, in this chapter, we propose a method that jointly estimates the latent image and spatially varying motion (kernel) based on the conventional blur constraint that the brightness of the blurred version of the latent image and that of the observed blurry image are same. Using pixel-wise kernel estimation, we could handle spatially varying blurs without accurate motion segmentation, and it was reported as the first approach in [8] that solved the joint problem of motion (kernel) estimation and latent image restoration for a single (non-uniformly) blurred image. Although this approach could handle spatially varying blurs and estimate accurate motion flows at the motion boundaries without using alpha mattes and segmentation results, it could not handle blurs in texture-less regions since it relies on the simple conventional blur constraint which is underconstrained. Therefore, strong and salient edges are needed everywhere to estimate the spatially varying kernel, and it can fail in motion estimation where the edges are weak and less-textured regions exist. Therefore, we further utilize motion blur cues from image-statistics [21, 50] to enable accurate motion (kernel) estimation even where strong edges are not localized, . Specifically, image-statistics of differently blurred images are distinct, and thus it is possible to calculate the likelihoods of candidate (linear) kernels. Therefore, we obtain motion

blur cues by comparing likelihoods of specific kernels. We adaptively integrate these blur cues into a new energy model to facilitate accurate motion estimation where edges are not strong. Moreover, we employ a higher order regularization scheme [44] which is well suited to estimate spatially varying motion flows and reduces the staircasing artifacts. Accordingly, we can achieve significantly improved qualities in both estimated motion (kernel), and the latent image. In addition, the proposed method is embedded into the traditional coarse-to-fine framework to handle large blur. Our finding is that small structures with distinct motion blur give rise to serious artifacts in the coarse-to-fine framework. Thus, we also propose a novel method to re-initialize the motion flow and reduce the error propagated in the coarse to fine framework.

In our optimization framework, in which motion (kernel) estimation and the latent image restoration are carried out simultaneously, we provide a practical and efficient solution that iteratively minimizes the proposed non-convex energy model. By optimizing the proposed energy function, we achieve significant improvements in deblurring quality. Quantitative experimental results with synthetic datasets demonstrate the significantly enhanced performance of the proposed method, and also experimental results on real challenging blurry images exhibit the superiority of the proposed method compared with current state-of-the-art methods.

4.1 Introduction and Related Work

As blurs in dynamic scene vary locally, many state-of-the-art uniform and non-uniform deblurring methods that handle blurs caused by global camera motion cannot be applied to remove general and spatially varying blurs [16, 17, 1, 5, 2, 6].

To address this problem, some researches have focused on deblurring dynamic scenes and established approaches that commonly require accurate motion segmentation. Couzinie-Devy et al. extended the work of Levin in [34] by casting the problem as a multi-label segmentation problem and estimating locally varying blur. However, the method could not handle large blur because of the exponential increases in the the number of candidate labels. This condition restricts the blur kernels to small 2D Gaussian or linear. In addition, we introduced a method that estimates the latent image, multiple blur kernels and associated segments simultaneously. Similarly, Schelten and Roth [54] also solved a joint problem of segmentation and deblurring using a Bayesian model of localized image blur, which assigns segment-wise different blur kernels.

In contrast, deblurring methods using kernel-parametrization can handle spatially varying and abruptly changing blurs by approximating the pixel-wise varying blur kernel. Dai and Wu [48] assumed locally linear blur kernel and parametrized the linear kernel using the motion flow. They proposed a new blur constraint using alpha channel image and estimated linear blur kernel. However, this approach requires computing accurate alpha mattes beforehand.

4.2 Preliminary

To handle spatially varying motion blurs, pixel-wise different kernel estimation is required. However, the solution space of kernel is extremely huge when blur kernel varies spatially, thus we need kernel approximation and parametrization to reduce the solution problem. In this section, we give a brief introduction to the kernel parametrization with motion flows to facilitate spatially varying kernel estimation

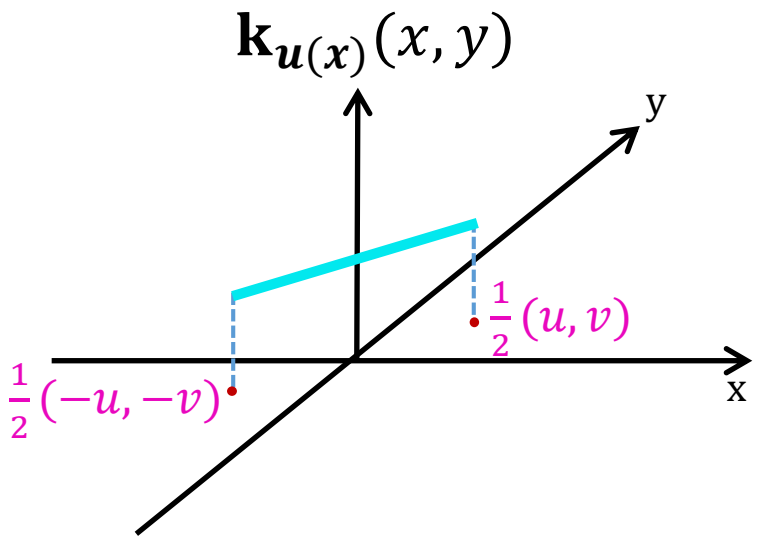


Figure 4.1: Parametrized linear blur kernel $\mathbf{k}_{\mathbf{u}(\mathbf{x})}$ corresponding to a motion vector $\mathbf{u}(\mathbf{x}) = (u, v)^T$.

practically.

According to previous works [21, 34, 48, 8], it is verified that numerous types of locally varying blur kernels can be parametrized with 2-dimensional motion vectors. Although this restriction excludes non-linear motion, many types of motions can be linearized in real situations. We observe that this assumption holds for many real blurry images and that this approach works even when small rotational camera shake exists, as shown in the experimental section. The blur kernel $\mathbf{k}_{\mathbf{u}(\mathbf{x})}$ at pixel location $\mathbf{x}$ using 2-dimensional motion vector $\mathbf{u}(\mathbf{x}) = (u, v)^T$ can be written by,

$$\mathbf{k}_{\mathbf{u}(\mathbf{x})}(x,y) = \begin{cases} \frac{1}{\|\mathbf{u}(\mathbf{x})\|}\delta(vx - uy), & \text{if } x \leq \frac{\|u\|}{2}, y \leq \frac{\|v\|}{2} \\ 0, & \text{otherwise} \end{cases}, \tag{4.1}$$

where δ denotes Kronecker delta and the point (x, y) denotes the location in 2-dimensional kernel space. An illustration of our linear blur kernel which corresponds to a linear motion vector $\mathbf{u}(\mathbf{x})$ is illustrated in Fig. 4.1.

4.3 Proposed Method

To solve the dynamic scene deblurring problem, our non-uniform deblurring method estimates motion flow to parametrize spatially varying blur kernel, and thus can restore the sharp edges at the motion boundaries. However, it is difficult to estimate the reliable kernel where the strong edges do not exist such as sand, grass, and leaves. On the other hand, several statistics-based approaches [21, 50, 55] can provide blur (motion) cues even where edges are not strong. However, these cues are unreliable at the motion boundaries since these approaches assume locally uniform kernel.

Hence, we propose a novel variational deblurring model with robust regularization that *adaptively* fuses reliable blur cues from image-statistics in order to estimate accurate blur kernel and restore the latent image even where the edge information is not sufficient. Extracting motion cues from image statistics, the new dynamic scene deblurring energy model, and details of efficient optimization techniques are described in the following subsections.

4.3.1 Image-statistics-guided motion

As we assume locally linear motion blur and the linear kernel can be parametrized with motion [48, 8], we present a model that estimates spatially varying motion using statistics-based information [50, 55]. Specifically, we propose to choose a pixel-wise different motion from multiple motion candidates as in a typical discrete labeling problem. The formulation is expressed as follows:

$$\min_{\mathbf{W}} \sum_i \sum_{\mathbf{x}} \mathbf{W}_i(\mathbf{x})\xi(\mathbf{x},\mathbf{u}_i) + \eta \sum_i \sum_{\mathbf{x}} |\nabla \mathbf{W}_i(\mathbf{x})| + \nu \sum_{(i,j)} \sum_{(\mathbf{x},\mathbf{y})} \mathbf{W}_i(\mathbf{x})\mathbf{W}_j(\mathbf{y}) \cdot |\mathbf{u}_i - \mathbf{u}_j|, \tag{4.2}$$

where $\mathbf{y}$ indexes the four neighboring pixels of pixel at $\mathbf{x}$. In addition, $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_N\}$ denotes a set of binary maps with pixel-wise constraints such that $\mathbf{W}_i(\mathbf{x}) \in \{0, 1\}$, and $\sum_{i=1}^{N} \mathbf{W}_i(\mathbf{x}) = 1$ where N denotes the number of candidate motions. The i_{th} candidate motion $\mathbf{u}_i$ is determined by changing the motion length between $5 \leq l \leq 29$ with an interval of two and the orientation between $-90^\circ \leq \theta < 90^\circ$ with an interval of 15°. Parameters η and ν control weights of the associated terms. With a specific kernel $\mathbf{k}_{\mathbf{u}_i}$ parameterized by $\mathbf{u}_i$, the data function in the first term using image-statistics is defined as follows:

$$\xi(\mathbf{x}, \mathbf{u}_i) = \exp(-p(\{y_m^{\triangledown}(\mathbf{x})\} \mid \mathbf{k}_{\mathbf{u}_i})), \tag{4.3}$$

where a set $\{y_m^{\triangledown}\} = \{f_m * y^{\triangledown}\}$ represents filter responses of the image derivative $y^{\triangledown}$, and $\{f_m\}$ represents a set of orthogonal sine filters multiplied by a symmetric window function. Notably, the directions of image derivative and propagation of sine waves are equal to the direction of $\mathbf{u}_i$. We calculate the likelihood in the data function 4.3 as suggested in the work [50], and the formulation is given as

$$p(\{y_m^{\triangledown}(\mathbf{x})\} \mid \mathbf{k}_{\mathbf{u}_i}) = \Pi_m \mathcal{N}(y_m^{\triangledown}(\mathbf{x}) \mid 0, s\sigma_{km}^2 + \sigma_{zm}^2), \tag{4.4}$$

where $\mathcal{N}$ denotes a normal distribution, and s controls the scale of the Gaussian image prior. The blur spectrum σ_{km}^2, and noise spectrum σ_{zm}^2 are defined as:

$$\begin{cases} \sigma_{km}^2 = \sum_{\mathbf{x}} |(\mathbf{k}_{\mathbf{u}_i} * f_m)(\mathbf{x})|^2 \\ \sigma_{zm}^2 = \sigma_n^2 \sum_{\mathbf{x}} |(\nabla * f_m)(\mathbf{x})|^2, \end{cases} \tag{4.5}$$

where σ_n^2 denotes the variance of Gaussian noise in the blurry image, and ∇ is a derivative filter in a particular direction (i.e. horizontal and vertical directions).

In the proposed energy model in (4.2), the first term denotes a unary term based on the image-statistics and favors a motion whose data cost is low. As these are selected locally, spatially-varying blurs can be handled. The second term represents the Potts model to enforce smoothness, in this case, same labels among neighboring pixels. The last term favors similar motion values among neighboring pixels.

In practice, to shorten the running time during the discrete optimization step, we reduce the number of labels by choosing the two dominant motion lengths at each orientation. Notably, we can choose pixel-wise best motion candidate at each orientation by simply comparing unary values, and the dominant motion length means the length of the most frequently chosen motion candidate over the entire image. In addition, we add zero motion (i.e. $l = 0$ and $\theta = 0$) as a candidate motion to handle unblurred region. The labeling function in (4.2) is a sub-modular, and we solve the problem using the TRW-S algorithm [56]. After the discrete optimization, we can determine image-statistics-guided motion by assigning the locally chosen motion model as follows:

$$\mathbf{u}_0(\mathbf{x}) = \sum_{i}^{N} \mathbf{W}_i(\mathbf{x})\mathbf{u}_i. \tag{4.6}$$

4.3.2 Adaptive variational deblurring model

The image-statistics-guided motion $\mathbf{u}_0$ is determined under the assumption that the kernel is locally uniform; thus, $\mathbf{u}_0$ is inaccurate at the motion boundaries where blur kernel varies abruptly. For example, image-statistics-guided motion at the boundaries of the moving car are inaccurate as shown in Fig. 4.3 (d). Moreover, as the number of candidate kernels is limited, the image-statistics-based approach would render artifacts during the deblurring process.

By contrast, a variational deblurring model [8] could estimate more accurate

motions at the motion boundaries than the statistics-based approach while still rendering poor results where the edges are not strong. Hence, it is natural to combine both approaches, and we propose a variational deblurring model which adaptively integrates the motion cue $\mathbf{u}_0$ from image statistics with a conventional deblurring framework.

Furthermore, to solve the highly ill-posed dynamic scene deblurring problem, we adopt total generalized variation (TGV) for higher order regularization of motion to enforce smoothness and reduce the staircasing phenomenon among neighboring pixels on a slanted surface in the proposed model.

4.3.2.1 Adaptive Blur Model

To overcome the limitations of the conventional blur model that requires strong and significant edges everywhere, we propose a robust blur model that adaptively fuses the conventional blur constraint and image-statistics-based motion $\mathbf{u}_0$ with a confidence map. The formulation is given as

$$\rho(\mathbf{L}, \mathbf{u}) = \lambda_p \sum_{\partial_*} \|\partial_* \mathbf{K}(\mathbf{u})\mathbf{L} - \partial_* \mathbf{B}\| + \sum_{\mathbf{x}} c(\mathbf{x}) \cdot \min(\tau, |\mathbf{u}(\mathbf{x}) - \mathbf{u}_0(\mathbf{x})|), \tag{4.7}$$

where $\mathbf{u}$, $\mathbf{K}$, $\mathbf{B}$, and $\mathbf{L}$ denote the motion, non-uniform blur kernel matrix, blurry image, and the latent image, respectively. Notably, blur kernel $\mathbf{K}$ is a function of motion $\mathbf{u}$ in the model, and a row vector of $\mathbf{K}$ corresponding to the pixel at $\mathbf{x}$ is the discretized vector form of local kernel $\mathbf{k}_{\mathbf{u}(\mathbf{x})}$ in Fig. 4.1. The linear operators $\partial_* \in \{\partial_o, \partial_x, \partial_y\}$ denote linear filters such as identity matrix, and partial derivative filters in the horizontal and vertical directions [17]. The parameter λ_p determines the weight of the first term, and $\tau = 1$ is a threshold. The locally adaptive confidence

map is represented as c.

In the proposed model, the first term represents a data term based on a conventional blur constraint (i.e. constancy between the blurry image and the convolved latent image with the blur kernel). To enable abrupt changes in the blur kernel near motion boundary, we propose a robust model in the data term that allows discontinuities in the flow field. Thus, the proposed data term is based on L_1 model for facilitating abrupt changes in the blur kernel near motion boundary while preserving discontinuities in the flow field. Although this robust L_1 model was proposed before to estimate the latent image in a non-blind deblurring process [1], this model has not been used in the estimation of blur kernel, because of its computational difficulties.

The second term adaptively integrates statistics-based motion $\mathbf{u}_0$ with the variational model using the confidence map. The confidence map is coupled with a robust function based on a truncated L_1 norm. We define the confidence map at $\mathbf{x}$ to determine whether $\mathbf{u}_0(\mathbf{x})$ is reliable and it is formulated as follows,

$$c(\mathbf{x}) = \gamma_p \exp(-\frac{\|P_{\mathbf{x}} - Q_{\mathbf{x}}\|}{\sigma_c}), \tag{4.8}$$

where $P_{\mathbf{x}}$ denotes the derivative distribution within a local (observed) image patch centered at $\mathbf{x}$ and the derivative is calculated along a direction of $\mathbf{u}_0$. In addition, $Q_{\mathbf{x}}$ denotes the derivative distribution of a local image patch convolved with a linear kernel which is obtained by rotating the kernel parameterized by $\mathbf{u}_0$ through 90 degree angle. Thus, the derivative is calculated along a direction perpendicular to $\mathbf{u}_0$. The parameters $\gamma_p = 1$ and $\sigma_c = 2$ control the weight and shape of the penalty function. Based on the fact that blurring effects arise along the motion (kernel) direction when the kernel is linear [21], distribution $Q_{\mathbf{x}}$ becomes similar to $P_{\mathbf{x}}$ when

$\mathbf{u}_0(\mathbf{x})$ is similar to the real motion. Hence, the confidence value $c(\mathbf{x})$ gains high value at $\mathbf{x}$ when $\mathbf{u}_0(\mathbf{x})$ is reliable by calculating the statistical distance between two distributions.

4.3.2.2 Robust Regularization

Regularization for an ill-posed problem is important as it renders desirable results, and in this work, we apply different regularizers for the motion (kernel) and the latent image.

First, we regularize motion flows by means of a higher order regularization instead of regularizing the kernel directly. By contrast, total variation (TV) model [57] is employed in our previous work [8]. Although the TV model is effective in reducing noise and is robust to outliers, it favors a constant flow filed and raises staircase artifacts owing to its first-order regularization process. Notably, TGV regularization outperforms other conventional priors in reducing staircasing artifacts while preserving the edges [58, 59, 60, 61]. Hence, we naturally employ TGV in our motion flow estimation.

The general form of the the second-order TGV for regularizing motion $\mathbf{u}$ is given by,

$$TGV^2(\mathbf{u}) = \min_{\mathbf{v}} \sum_{\mathbf{x}} \alpha_1 |\nabla \mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})| + \alpha_0 |\nabla \mathbf{v}(\mathbf{x})|, \tag{4.9}$$

where an auxiliary vector field $\mathbf{v}$ is introduced to enforce second-order smoothness for $\mathbf{u}$ as described in [44], and weighting parameters α_0 and α_1 control the weights of the associated terms.

In addition, our TGV regularizer is coupled with edge-map because it is known

that the additional edge information from the image renders much sharper and fine edges in the flow field [45, 59, 62]. Thus, we couple an edge-map g to the general form of TGV, and allow motion discontinuity at the image edges by preferring similar intensities among neighboring pixels as follows:

$$\min_{\mathbf{u},\mathbf{v}} \sum_{\mathbf{x}} \alpha_1 \cdot g(\mathbf{x}) |\nabla \mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})| + \alpha_0 |\nabla \mathbf{v}(\mathbf{x})|, \tag{4.10}$$

and our edge-map is given by,

$$g(\mathbf{x}) = \exp(-\frac{\max((\nabla \mathbf{L}^0(\mathbf{x}))^2, (\nabla \mathbf{B}(\mathbf{x}))^2)}{\sigma_g^2}), \tag{4.11}$$

where σ_g controls shape of the Gaussian penalty function and set to $\sigma_g = 11/255$ in our experiments. Similar to [8, 19], $\mathbf{L}^0$ represents an initial latent image in the alternating and iterative optimization described in Sec. 4.4. Notably, higher order regularization is utilized to infer accurate kernel, meaning that this is the first work to use higher-order regularization for blur kernel estimation to the best of our knowledge.

Next, we employ the TV model to regularize the latent image $\mathbf{L}$. Although a hyper-Laplacian prior and a sparse gradient prior fit the natural statistics of the image gradient more correctly [26], we employ TV regularization in the proposed regularizing model because it is easy to minimize while providing moderate sparsity.

Thus, the proposed regularization model for the latent image and flow field is expressed as follows:

$$\min_{\mathbf{L},\mathbf{u},\mathbf{v}} \sum_{\mathbf{x}} |\nabla \mathbf{L}(\mathbf{x})| + \alpha_1 \cdot g(\mathbf{x}) |\nabla \mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})| + \alpha_0 |\nabla \mathbf{v}(\mathbf{x})|. \tag{4.12}$$

4.4 Optimization

The proposed dynamic scene deblurring model includes the adaptive data term and robust regularizer, and our final energy function is given as follows:

$$\begin{aligned}\mathbf{E}(\mathbf{L},\mathbf{u},\mathbf{v}) = \lambda_p \sum_{\partial_*} \|\partial_* \mathbf{K}(\mathbf{u})\mathbf{L} - \partial_* \mathbf{B}\| + \\ \sum_{\mathbf{x}} c(\mathbf{x}) \min(\tau, |\mathbf{u}(\mathbf{x}) - \mathbf{u}_0(\mathbf{x})|) + |\nabla \mathbf{L}(\mathbf{x})| + \\ \sum_{\mathbf{x}} \alpha_1 \cdot g(\mathbf{x}) |\nabla \mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})| + \alpha_0 |\nabla \mathbf{v}(\mathbf{x})|.\end{aligned} \tag{4.13}$$

The proposed final objective model in (4.13) is highly non-convex, and is very difficult to optimize. Thus, we divide the original problem into two easier subproblems, and the proposed energy model is minimized through an iterative and alternating optimization technique. To be specific, we restore the latent image $\mathbf{L}$ while fixing the other variables $\mathbf{u}$ and $\mathbf{v}$, and we estimate the motion flows $\mathbf{u}$ and the auxiliary field $\mathbf{v}$ while fixing the latent image $\mathbf{L}$.

Moreover, our approach is embedded into the traditional coarse-to-fine framework [16, 32] to handle large motion blur. However, we found that it gives rise to artifacts when applied to a dynamic scene, which has small structures with distinct motion blurs. Thus, we introduce a re-initialization method to solve this problem.

4.4.1 Motion estimation

For being the latent image $\mathbf{L}$ is fixed, the subproblem becomes motion (kernel) estimation problem, and it yields,

$$\begin{aligned}\min_{\mathbf{u},\mathbf{v}} \lambda_p \sum_{\partial_*} \|\partial_* \mathbf{K}(\mathbf{u})\mathbf{L} - \partial_* \mathbf{B}\| + \\ \sum_{\mathbf{x}} c(\mathbf{x}) \min(\tau, |\mathbf{u}(\mathbf{x}) - \mathbf{u}_0(\mathbf{x})|) + \\ \sum_{\mathbf{x}} \alpha_1 \cdot g(\mathbf{x}) |\nabla \mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})| + \alpha_0 |\nabla \mathbf{v}(\mathbf{x})|.\end{aligned} \tag{4.14}$$

Although our regularizers in the proposed model are convex, the data fidelity term with respect to $\mathbf{u}$ is non-convex and it makes the optimization intractable. To handle the non-convex data fidelity term, we employ the strategy used in [62, 27, 63] that approximates the data fidelity term with a second-order Taylor expansion. Then, we can apply the conventional convex optimization method [29] to minimize the approximated convex function.

In doing so, we first define a non-convex data function ρ_u, and it is given by,

$$\rho_u(\mathbf{x}, \mathbf{u}(\mathbf{x})) = \lambda_p \sum_{\partial_*} |\partial_* \mathbf{K}(\mathbf{u})\mathbf{L}(\mathbf{x}) - \partial_* \mathbf{B}(\mathbf{x})| + c(\mathbf{x}) \cdot \min(\tau, |\mathbf{u}(\mathbf{x}) - \mathbf{u}_0(\mathbf{x})|), \tag{4.15}$$

and we approximate ρ_u near an initial value $\hat{\mathbf{u}}$ as follows:

$$\begin{aligned}\rho_u(\mathbf{x}, \mathbf{u}(\mathbf{x})) \approx \rho_u(\mathbf{x}, \hat{\mathbf{u}}(\mathbf{x})) + (\mathbf{u}(\mathbf{x}) - \hat{\mathbf{u}}(\mathbf{x}))^T \nabla \rho_u(\mathbf{x}, \hat{\mathbf{u}}(\mathbf{x})) \\ + \frac{1}{2}(\mathbf{u}(\mathbf{x}) - \hat{\mathbf{u}}(\mathbf{x}))^T \nabla^2 \rho_u(\mathbf{x}, \hat{\mathbf{u}}(\mathbf{x}))(\mathbf{u}(\mathbf{x}) - \hat{\mathbf{u}}(\mathbf{x})),\end{aligned} \tag{4.16}$$

where $\nabla \rho_u$ denotes the first-order derivative and $\nabla^2 \rho_u$ denotes the Hessian matrix whose non-diagonal elements are set to zero to guarantee convexity [62, 27, 63].

Thus, our motion estimation problem using the approximated convex function

is formulated as follows:

$$\begin{aligned}\min_{\mathbf{u},\mathbf{v}} \sum_{\mathbf{x}} \alpha_1 \cdot g(\mathbf{x})|\nabla \mathbf{u}(\mathbf{x}) - \mathbf{v}(\mathbf{x})| + \alpha_0 |\nabla \mathbf{v}(\mathbf{x})| + \\ \sum_{\mathbf{x}} \rho_u(\mathbf{x}, \hat{\mathbf{u}}(\mathbf{x})) + (\mathbf{u}(\mathbf{x}) - \hat{\mathbf{u}}(\mathbf{x}))^T \nabla \rho_u(\mathbf{x}, \hat{\mathbf{u}}(\mathbf{x})) + \\ \sum_{\mathbf{x}} \frac{1}{2} (\mathbf{u}(\mathbf{x}) - \hat{\mathbf{u}}(\mathbf{x}))^T \nabla^2 \rho_u(\mathbf{x}, \hat{\mathbf{u}}(\mathbf{x}))(\mathbf{u}(\mathbf{x}) - \hat{\mathbf{u}}(\mathbf{x})).\end{aligned} \tag{4.17}$$

Finally, we apply the convex optimization technique in [29] to minimize the approximated convex function, and derive the primal-dual update scheme as follows:

$$\begin{cases} \mathbf{p}^{n+1} = \frac{\mathbf{p}^n + \sigma_p \alpha_1 \mathbf{G}(\nabla \overline{\mathbf{u}}^n - \overline{\mathbf{v}}^n)}{\max(\mathbf{1}^T,\ |\mathbf{p}^n + \sigma_p \alpha_1 \mathbf{G}(\nabla \overline{\mathbf{u}}^n - \overline{\mathbf{v}}^n)|)} \\ \mathbf{q}^{n+1} = \frac{\mathbf{q}^n + \sigma_q \alpha_0 \nabla \overline{\mathbf{v}}^n}{\max(\mathbf{1}^T,\ |\mathbf{q}^n + \sigma_q \alpha_0 \nabla \overline{\mathbf{v}}^n|)} \\ \mathbf{u}^{n+1} = \frac{\mathbf{u}^n + \tau_u(\nabla^T \mathbf{G}\mathbf{p}^{n+1}) - \tau_u(\nabla \rho_u - \hat{\mathbf{u}} \nabla^2 \rho_u)}{(\mathbf{1}^T + \tau_u \nabla^2 \rho_u)} \\ \mathbf{v}^{n+1} = \mathbf{v}^n + \tau_v(\alpha_0 \nabla^T \mathbf{q}^{n+1} + \alpha_1 \mathbf{G}\mathbf{p}^{n+1}) \\ \overline{\mathbf{u}}^{n+1} = 2\mathbf{u}^{n+1} - \overline{\mathbf{u}}^n \\ \overline{\mathbf{v}}^{n+1} = 2\mathbf{v}^{n+1} - \overline{\mathbf{v}}^n, \end{cases} \tag{4.18}$$

where $n \geq 0$ denotes the iteration number, and initially $\overline{\mathbf{u}}^0 = \mathbf{u}^0$ and $\overline{\mathbf{v}}^0 = \mathbf{v}^0$ (primal variables are refined through an over-relaxation step as in [29]). In addition, $\mathbf{p}$ and $\mathbf{q}$ are dual variables and we initially set the dual variables to zeros. The step sizes, $\sigma_p > 0$, $\sigma_q > 0$, $\tau_u > 0$, $\tau_v > 0$, are determined by means of preconditioning method as introduced in [64]. The linear operator $\mathbf{G}$ is a diagonal matrix given as $\mathbf{G} = diag(g)$, and we run the process for 50 iterations ($n = 50$). Notably, division operators used in the update steps denote the element-wise division.

4.4.2 Latent image restoration

For being $\mathbf{u}$ and $\mathbf{v}$ are fixed, the subproblem becomes the latent image restoration problem, and it is convex with respect to the latent image $\mathbf{L}$ as follows:

$$\min_{\mathbf{L}} \ |\nabla \mathbf{L}| + \lambda_p \sum_{\partial_*} \|\partial_* \mathbf{K}\mathbf{L} - \partial_* \mathbf{B}\|. \tag{4.19}$$

Although the proposed formulation is convex, it has some difficulties in minimization since the model has L_1 data fidelity term. Hence, we adopt a half-quadratic splitting method used in [34] to minimize the model more efficiently, and the formulation with additional auxiliary variables $\mathbf{e}_*$ is given by,

$$\min_{\mathbf{L},\mathbf{f}_*} \ |\nabla \mathbf{L}| + \sum_{\partial_*} \frac{1}{2\theta} \|\partial_* \mathbf{K}\mathbf{L} - \partial_* \mathbf{B} + \mathbf{f}_*\|^2 + \lambda_p \|\mathbf{f}_*\|. \tag{4.20}$$

When θ approaches to a very small number (i.e. $\theta \longrightarrow 0$), the solution of (4.20) approaches that of its original TV-L_1 model in (4.19). Using this decomposition, we optimize $\mathbf{f}_*$ and $\mathbf{L}$ iteratively with fixed θ, and it yields,

$$\begin{aligned} &\min_{\mathbf{f}_*} \ \sum_{\partial_*} \frac{1}{2\theta} \|\partial_* \mathbf{K}\mathbf{L} - \partial_* \mathbf{B} + \mathbf{f}_*\|^2 + \lambda_p |\mathbf{f}_*|, \\ &\min_{\mathbf{L}} \ |\nabla \mathbf{L}| + \sum_{\partial_*} \frac{1}{2\theta} \|\partial_* \mathbf{K}\mathbf{L} - \partial_* \mathbf{B} + \mathbf{f}_*\|^2. \end{aligned} \tag{4.21}$$

Hence, functions of $\mathbf{L}$ and $\mathbf{f}_*$ become easy to solve, and each model is solved by [29] in the alternating optimization manner. In the primal-dual update steps, primal variables are estimated in an over-relaxation step, and the step sizes are determined as $\sigma_s > 0$, $\tau_f > 0$, $\sigma_t > 0$, and $\tau_L > 0$ according to [29]. The iteration number m is fixed ($m = 30$), and we repeat the alternating optimization of $\mathbf{L}$ and $\mathbf{f}_*$ for five times.

First, the primal-dual update process for $\mathbf{f}_*$ while $\mathbf{L}$ is fixed is as follows:

$$\begin{cases} \mathbf{s}_*^{m+1} = \frac{\mathbf{s}_*^m + \sigma_s \bar{\mathbf{f}}_*^m}{\max(\mathbf{1}^T, \mathbf{s}_*^m + \sigma_s \bar{\mathbf{f}}_*^m)} \\ \mathbf{f}_*^{m+1} = \arg\min_{\mathbf{f}} \frac{\|\mathbf{f} - (\mathbf{f}_*^m - \tau_f \mathbf{s}_*^{m+1})\|^2}{2\tau_f} + \\ \qquad \frac{1}{2\theta\lambda_p} \sum_{\partial_*} \|\partial_* \mathbf{K}\mathbf{L} - \partial_* \mathbf{B} + \mathbf{f}_*\|^2 \\ \bar{\mathbf{f}}_*^{m+1} = 2\mathbf{f}_*^{m+1} - \bar{\mathbf{f}}_*^m, \end{cases} \tag{4.22}$$

where $\mathbf{s}_*$ denote dual variables and we set these variables to zeros initially, and use $\bar{\mathbf{f}}_*^0 = \mathbf{f}_*^0$ for over-relaxation step.

Similarly, the primal-dual update scheme for $\mathbf{L}$ while $\mathbf{f}_*$ is fixed is given by,

$$\begin{cases} \mathbf{t}^{m+1} = \frac{\mathbf{t}^m + \sigma_t \nabla \mathbf{L}^m}{\max(\mathbf{1}^T, \mathbf{t}^m + \sigma_t \nabla \mathbf{L}^m)} \\ \mathbf{L}^{m+1} = \arg\min_{\mathbf{L}} \frac{\|\mathbf{L} - (\mathbf{L}^m - \tau_L \nabla^T \mathbf{t}^{m+1})\|^2}{2\tau_L} + \\ \qquad \frac{1}{2\theta\lambda_p} \sum_{\partial_*} \|\mathbf{K}\partial_* \mathbf{L} - \partial_* \mathbf{B} + \mathbf{f}_*\|^2 \\ \bar{\mathbf{L}}^{m+1} = 2\mathbf{L}^{m+1} - \bar{\mathbf{L}}^m, \end{cases} \tag{4.23}$$

where $\mathbf{t}$ denotes the dual variable and set as zero initially, and $\bar{\mathbf{L}}_*^0 = \mathbf{L}_*^0$ for over-relaxation step.

4.4.3 Kernel re-initialization

The conventional coarse-to-fine approach is widely used in numerous vision applications and also has been shown to yield successful results in the deblurring of static scene. This method can be used in dynamic scene deblurring, but it has limitations that have not been observed in the restoration of a static scene.

For example, a small structure, such as a thin line, that has distinctive motion blur, cannot be seen at the coarser level. Therefore, the motion flow of such small

structure is estimated from its neighbors, which exhibit different motion, via regularization. However, this process generates reconstruction error toward the finer level when the small structure appears suddenly in the blurry image. Notably, a similar artifact has been reported in other problems [65, 7].

Sparse reconstruction error may not reveal severe artifacts in removing blurs caused by global camera motion, which is more serious in a local approach, such as the method proposed in this work.

To mitigate this problem, we propose a novel kernel re-initialization method for both motion flow and the latent image. The key idea is to detect the erroneous region and deblur it using a deterministic filter [66], after which accurate motion flow is estimated. Hence, we cast the problem of detecting the erroneous region as a labeling problem, and it yields,

$$\min_{\mathbf{e}} \sum_{\mathbf{x}} \mathbf{e}(\mathbf{x})(\epsilon - |\mathbf{KL}(\mathbf{x}) - \mathbf{B}(\mathbf{x})|) + \frac{1}{\kappa}|\mathbf{e}(\mathbf{x})|, \tag{4.24}$$

where the vector variable $\mathbf{e}$ denotes the pixel-wise binary indicator variable, $\mathbf{e}(\mathbf{x}) \in \{0, 1\}$, and the constant ϵ is a positive threshold value.

The first term introduced is the likelihood term that enforces $\mathbf{e(x)}$ to be one when the reconstruction error, $|\mathbf{KL}(\mathbf{x}) - \mathbf{B}(\mathbf{x})|$, is higher than ϵ. The second term is a prior giving the sparsity on the variable $\mathbf{e}$, as we assume reconstruction error is sparse, and the parameter κ determines the weight of the sparse prior. Through the continuous relaxation of $\mathbf{e}$, we can obtain the approximated solution of (4.24) quickly by adapting [29].

After detection of the erroneous region, we re-initialize the propagated motion of this region to be zero, which denotes the impulse blur kernel. By applying sharp im-

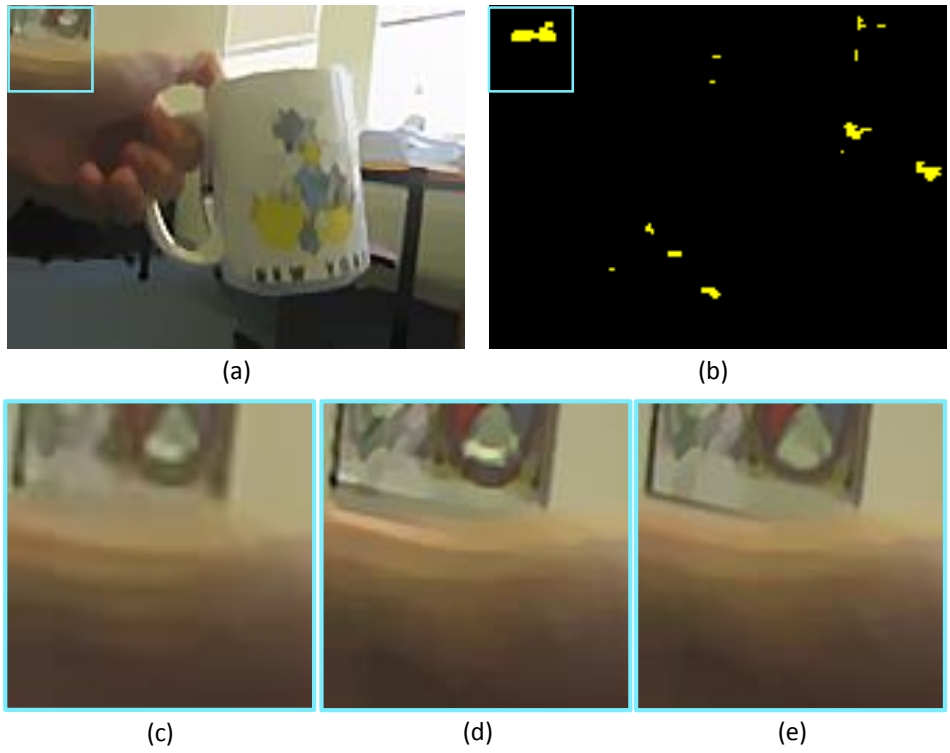


Figure 4.2: Effect of kernel-reinitialization. (a) Estimated latent image in a coarse level. (b) Yellow color denotes the detected erroneous region. (c) Cropped result in the coarse level. (d) Cropped result in the finest level without the use of re-initialization. (e) Cropped result in the finest level with the use of re-initialization.

age restoration in Sec. 4.4.2, we restore the small structure with distinct blur motion. However, the newly restored small structure remains blurry because it is estimated from the impulse blur kernel (i.e. zero motion). Thus, we apply a deterministic filter [66] to deblur the erroneous region and to facilitate the fast convergence of motion estimation. For this process, we use the prediction step introduced in [17]. Our re-initialization step is illustrated in Fig. 4.2. Notably an unseen thin line in the coarse level of pyramid in is successfully restored using the proposed method.

4.5 Experiments

Algorithm 3 Overview of the proposed method

Input: A blurry image $\mathbf{B}$

Output: Latent image $\mathbf{L}$ and motion flow $\mathbf{u}$

1: Estimate image-statistics-guided motion $\mathbf{u}_0$. (Sec. 4.3.1)
2: Build an image pyramid.
3: Resize and rescale $\mathbf{u}_0$ according to the size of the image in the pyramid.
4: Kernel re-initialization. (Sec. 4.4.3)
5: **for** $iter = 1$ to 3 **do**
6: Continuous optimization of motion flow $\mathbf{u}$ with fixed $\mathbf{L}$. (Sec. 4.4.1)
7: Continuous optimization of latent image $\mathbf{L}$ with fixed $\mathbf{u}$. (Sec. 4.4.2)
8: **end for**
9: Propagate variables to the next pyramid level if exists.
10: Repeat steps 3-9 from coarse to fine pyramid level.

In our experiments, we use fixed parameters for our experiments as $\{\eta = 0.02, \eta = 0.02, \lambda_p = 40, \alpha_0 = 1, \alpha_1 = 5, \theta = \frac{0.1}{\lambda_p}, \epsilon = 0.05, \kappa = 0.02\}$, and numerous experiments are carried out to demonstrate the good performance of the proposed method. To guide fast convergence and estimate large kernel (motion), we implement the proposed algorithm on the traditional coarse-to-fine framework . In the coarse-to-fine framework, we build an image pyramid of 10 levels with a scale factor of 0.8. In addition, we rescale and resize image-statistics-guided motion $\mathbf{u}_0$ for each pyramid level. Initially, we use $\mathbf{u}_0$ for initial motion flow $\mathbf{u}$, and the blurry image is used for initial latent image $\mathbf{L}$. For easier minimization, we constrain the horizontal motion values to have non-negative values, since our kernel is symmetric (i.e. $\mathbf{k}_{\mathbf{u}(\mathbf{x})}(x, y) = \mathbf{k}_{\mathbf{u}(\mathbf{x})}(-x, -y)$). The overall flow of the proposed method is summarized in algorithm 3.

Our current MATLAB code takes about 20 minutes to deblur a VGA image on

Intel core i5, and it can be further reduced via optimized C and GPU. Although we cannot guarantee a globally optimal solution by using the alternating optimization method like other methods [17], each subproblem is easy to solve, and we've achieved reliable solutions in our many experiments.

In Fig. 4.3, we compare the results with different settings of the proposed method using a synthetic image. Although ours without utilizing motion cues (i.e. $c(\mathbf{x}) = 0$) accurately estimates motion flows of the moving bus and restores the sharp edges at the motion boundaries, flows at the static background region are inaccurate due to the lack of strong edges. Hence it fails to restore the grass. By contrast, ours utilizing motion cues $\mathbf{u}_0$ and the confidence map estimates accurate motion flows even where edges are not strong, and successfully restores the grass as well the moving bus.

Additional comparisons with a real image are shown in Fig. 4.4. In the results, state-of-the-art uniform kernel deblurring methods [] fail in deblurring due to the differently moving car. Moreover, our segmentation-based method [10] for dynamic scene deblurring raises artifacts near the segmentation boundaries. By contrast, our methods with and without using motion cues $\mathbf{u}_0$ restore the edges at the motion boundaries of the moving car. However, ours without using motion cues fails to estimate the motion flow where edges are weak, thus loses small details in the sidewalk. By contrast, ours using motion cues estimates accurate motion flows in these areas and preserves details better.

In Fig. 4.6, deblurring results for camera shakes which includes rotational camera movements are illustrated. Although the blurs caused by rotational camera shake are locally non-linear, our linearly approximated kernel estimation method well restores strong edges from both synthetic image in the top row and the real image in the bottom row. These results demonstrate that our linear approximation is valid for

small rotational camera shake to some degree.

In Fig. 4.5, we demonstrate the good performance of our robust regularization. As a taxi moves forward, blur kernel varies spatially on the slanted surface of the taxi. Therefore typical staircasing artifacts arise in that area due to the TV regularizer. However, our higher order TGV regularizer effectively removes these artifacts and estimates piecewise affine flow field on the slanted and planar surface of the taxi. Notably, our segmentation-based method [10] in Fig. 4.5 (b) fails in segmentation because it cannot handle spatially varying blurs by forward or backward motion of moving objects. By contrast, ours can handle these blurs since it does not rely on accurate segmentation for specific type of motion blur.

Next, to verify the deblurring performance quantitatively and compare with our previous methods, several synthetic datasets generated by the method in 5.5 are used for evaluation, and PSNR values are compared in Table. 4.1. Notably, using the TGV regularization renders quantitatively better results than using the TV regularization, and our deblurring model with both the robust TGV regularization and motion cues significantly improves the deblurring quality.

In Fig. 4.7, deblurring results and estimated motion flows are illustrated for challenging dynamic scenes. In qualitative analysis, the edges in deblurred images are restored keenly, and the proposed method successfully removes the general non-uniform blurs caused by differently moving objects, depth variation, and camera shake as expected even at texture-less and homogeneous regions. Moreover, the color codes of estimated motion flows are as accurate as we expected. Notably, the motion flows in the the bottom rows are corresponding to the depth maps of the latent images, meaning that our work can be further utilized to infer depth map from a blurry single-image.

Seq.	segmentation	exemplar	parametrization (TV only)	parametrization (TGV only)	parametrization (all)
#1	20.25	26.48	25.09	25.82	29.40
#2	23.23	28.09	25.37	26.29	30.60
#3	22.46	26.22	25.83	25.84	26.54
#4	22.47	25.69	25.56	25.64	25.95
#5	22.20	23.26	22.56	22.29	26.53
#6	21.56	24.68	22.71	23.06	23.09
#7	23.54	25.73	26.45	27.29	28.57
#8	20.69	25.30	25.10	25.07	26.18
#9	20.35	25.03	24.17	24.38	24.82
#10	20.84	24.02	21.57	22.53	23.08
Avg.	21.76	25.45	24.44	24.82	26.48

Table 4.1: Deblurring performance evaluations in terms of PSNR values.

Lastly, we compared our method using kernel-parametrization with the uniform and non-uniform deblurring methods in various challenging situations. In Fig. 4.8 (a)-(b), comparison targets fail because of the differently moving multiple objects which are too small to handle, but ours deblurs well. In Fig. 4.8 (c), [6] fails in deblurring as depth discontinuity exist in the blurry image. Although it is possible to deblur with segmentation based method, but our segmentation-based method [10] also fails, because the background is too narrow to be segmented. Thus, conventional methods provide unsatisfactory results. In Fig. 4.8 (d)-(e), although the causes of locally varying blurs are specific forward motions which can not be handled in the comparison targets, ours outperforms and restores the edges of characters, arrow, and the taillight of the taxi to a significantly better degree.

4.6 Summary

In this chapter, we proposed a novel deblurring method that removes locally varying general blurs in dynamic scenes based on the approximation technique of the local blur kernel as a motion model. To estimate pixel-wise varying motion blur (kernel) even where strong edges are missing, we proposed a new deblurring method that adaptively combines blur cues from image-statistics with the conventional blur constraint. We formulated a new variational energy model, and employed a higher order regularizer into the proposed energy model to reduce staircasing artifacts among neighboring motion flows on slanted surfaces. Moreover, we proposed efficient and practical optimization techniques for minimizing the final non-convex energy model. By minimizing the proposed model with our efficient solvers, we achieved significant improvements in motion (kernel) estimation and deblurring qualities compared with the state-of-the art methods.

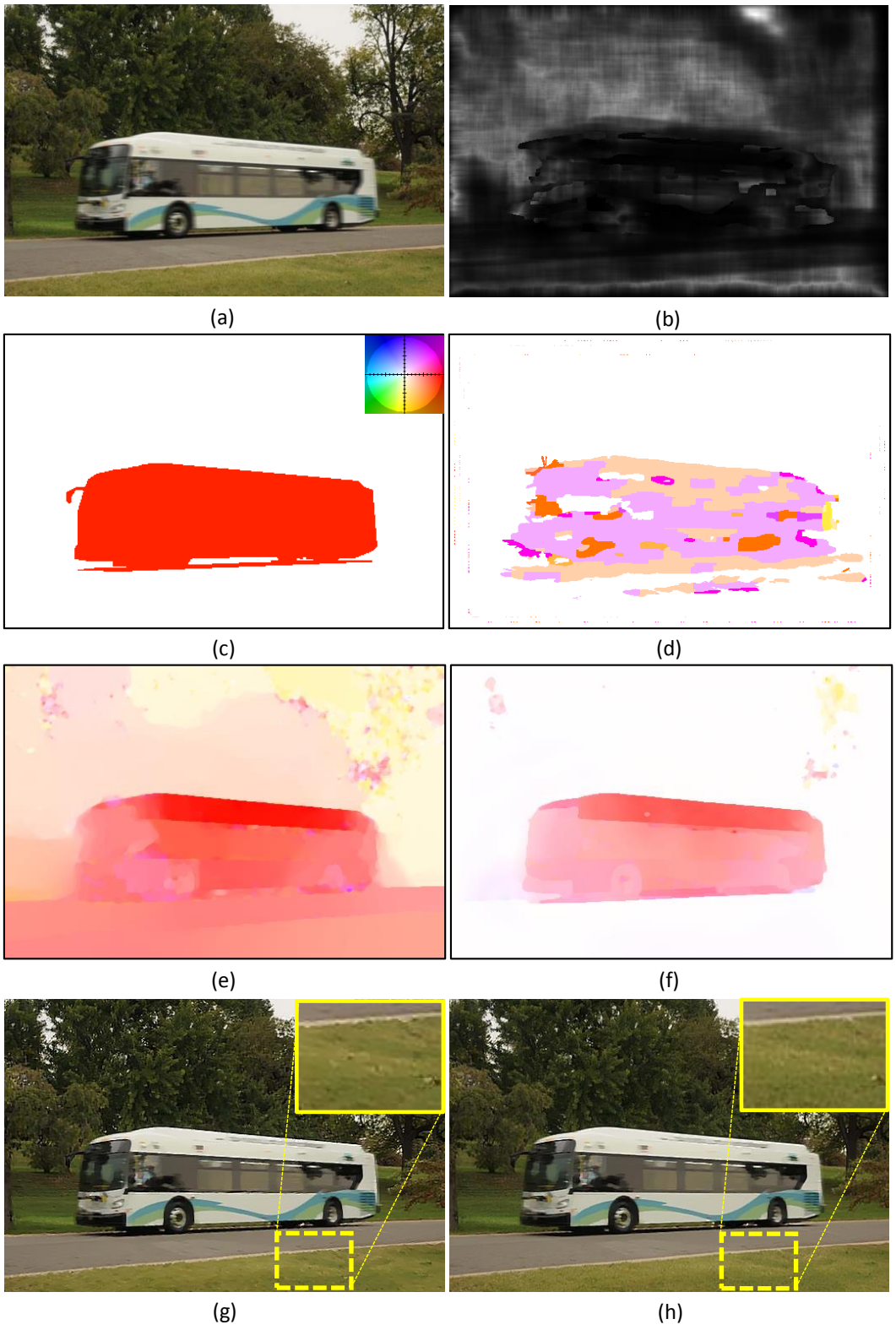


Figure 4.3: Effect of statistical motion cues. (a) Blurry image. (b) Confidence map. (c) Color coded ground truth flow. (d) $\mathbf{u}_0$. (e) Flow without using $\mathbf{u}_0$. (f) Flow using $\mathbf{u}_0$. (g) Deblurring result without using $\mathbf{u}_0$. (h) Deblurring result using $\mathbf{u}_0$.

Figure 4.4: Qualitative comparisons. (a) Real Blurry image. (b) Result from [13]. (c) Result from [6]. (d) Result from our segmentation-based method [10]. (e) Deblurring result without motion cues. (f) Deblurring result with motion cues. (g) Flow result without using motion cues. (h) Flow result with motion cues.

Figure 4.5: Robust regularization. (a) Real blurry image. (b) Segmentation result from our segmentation-based deblurring method [10]. (c) Motion flow with TV regularization ($\mathbf{v} = 0$). (d) Motion flow with our TGV regularization.

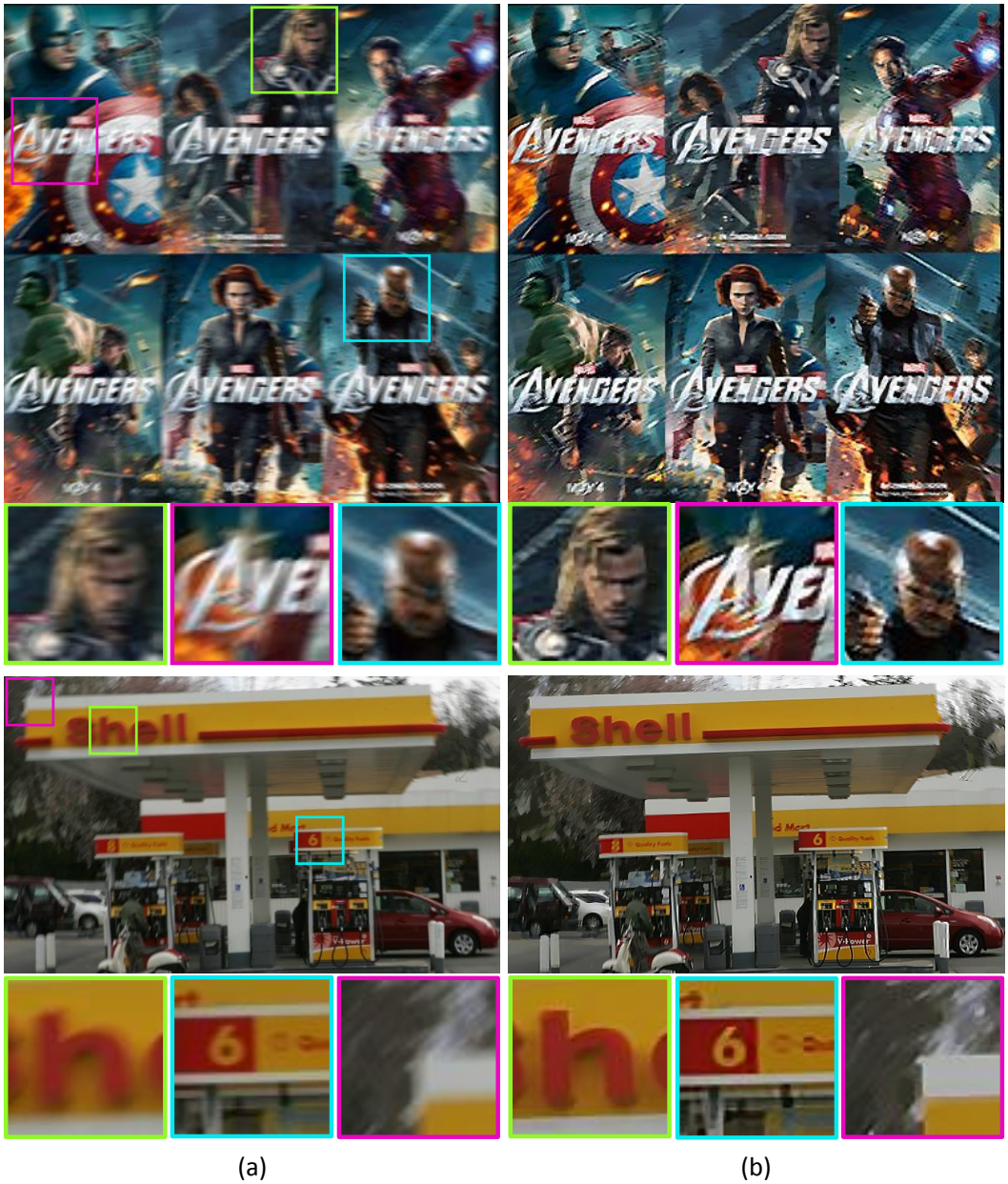


Figure 4.6: Validity check over rotational camera shake. (a) Single blurry images from rotational camera shakes. (b) Deblurring results using kernel-parametrization.

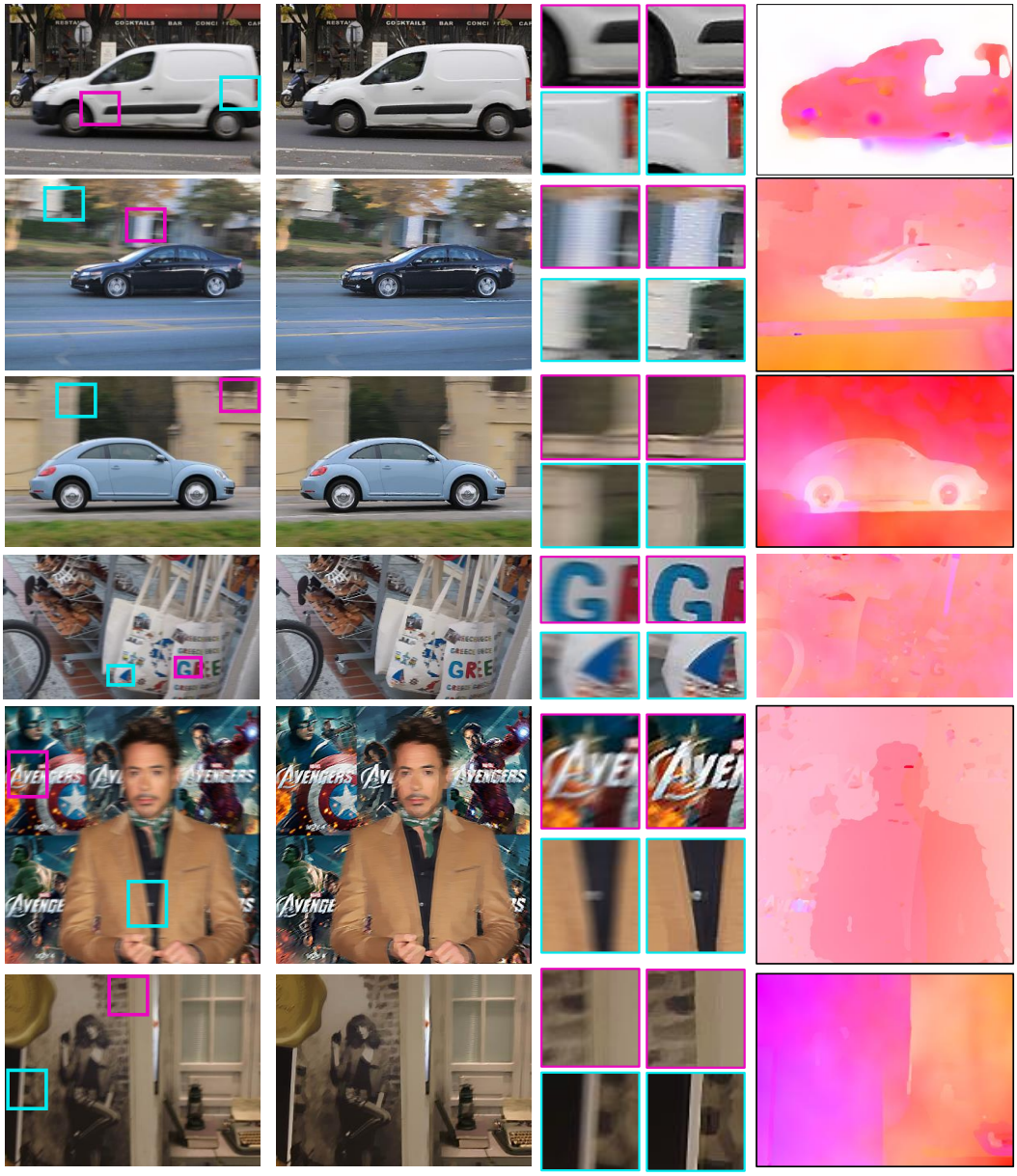


Figure 4.7: Deblurring and flow results. (a) Single blurry images of dynamic scenes. (b) Deblurring results. (c) Cropped results. (d) Estimated motion flows.

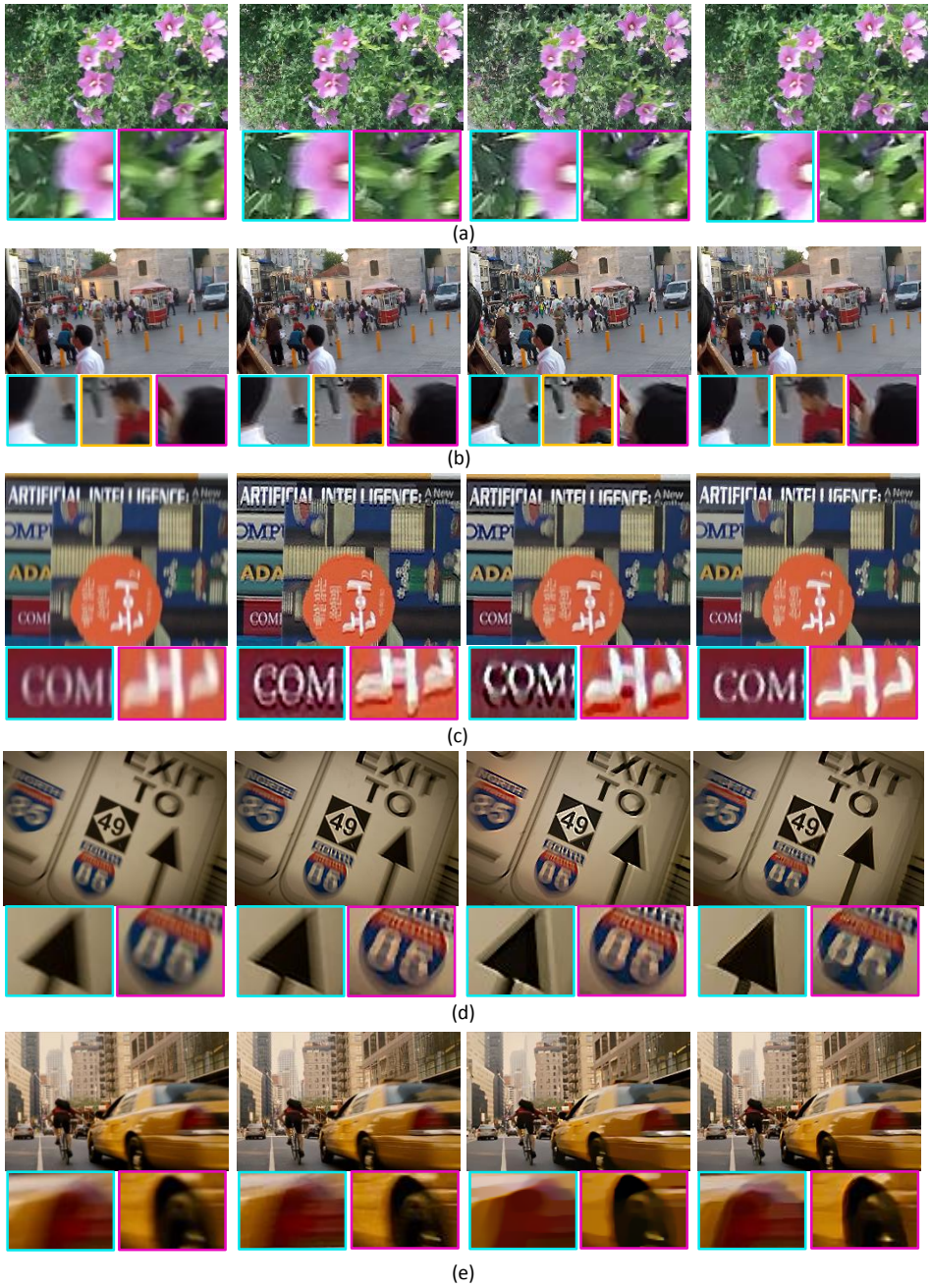


Figure 4.8: Other deblurring results on very challenging single blurry images. From left to right: Blurry images, results from Xu et al [6], results by using segmentation [10], and results by using kernel-parametrization.

Chapter 5

Video Deblurring with Kernel-Parametrization

In the previous chapter, we introduced a single image deblurring method based on kernel-parametrization that removes spatially varying blurs caused by various sources using a linear kernel approximation. In this chapter, we naturally extend the kernel-parameterization method to remove spatially and temporally varying blurs in videos. To improve deblurring qualities, we adopt more accurate blur kernel approximation model (i.e. piece-wise linear model using bi-directional optical flows) and handle defocus blurs as well as motion blurs. Moreover, we further utilize temporal information inherent among adjacent frames to render high quality videos.

5.1 Introduction and Related Work

In low-light conditions, motion blurs are caused by camera shake and object motions during exposure time. In addition, fast moving objects in the scene cause blurring

artifacts in a video even when the light conditions are acceptable. For decades, this problem has motivated considerable works on deblurring and different approaches have been sought depending on whether the captured scenes are static or dynamic.

Early works on a single image deblurring problem are based on assumptions that the captured scene is static and has constant depth [17, 16, 5, 4, 3, 2] and they estimated uniform or non-uniform blur kernel by camera shake. These approaches were naturally extended to video deblurring methods. Cai et al. [37] proposed a deconvolution method with multiple frames using sparsity of both blur kernels and clear images to reduce errors from inaccurate registration and render high-quality latent image. However, this approach removes only uniform blur caused by 2-dimensional translational camera motion, and the proposed approach cannot handle non-uniform blur from rotational camera motion around z-axis, which is the main cause of motion blurs [2]. To solve this problem, Li et al. [38] adopted a method parameterizing spatially varying motions with 3x3 homographies based on the previous work of Tai et al. [67], and could handle non-uniform blurs by rotational camera shake. In the work of Cho et al. [68], camera motion in 3-dimensional space was estimated without any assistance of specialized hardware, and spatially varying blurs caused by projective camera motion were obtained. Moreover, in the works of Paramanand et al. [69] and Lee and Lee [70], spatially varying blurs by depth variation in a static scene were estimated and removed.

However, these previous methods, which assume static scene, suffer from spatially varying blurs from not only camera shake but also moving objects in a dynamic scene. Because it is difficult to parameterize the pixel-wise varying blur kernel in the dynamic scene with simple homography, kernel estimation becomes more challenging task. Therefore, several researchers have studied on removing blurs in dynamic

Figure 5.1: (a) A blurry frame in a dynamic scene. (b) Our deblurring result. (c) Our color coded optical flow estimation result.

scenes, which are grouped into two approaches: segmentation-based deblurring approach, and exemplar-based deblurring approach.

Segmentation-based approaches usually estimate multiple motions, kernels, and associated segments. In the work of Cho et al. [33], a method that segments homogeneous motions and estimates segment-wise different (1-dimensional) Gaussian blur kernels, was proposed. However, it cannot handle complex motions by rotational camera shakes due to the limited capacity of Gaussian kernels. In the work of Bar et al. [71], a layered model was proposed that segments images into foreground and background layers, and estimates a linear blur kernel within the foreground layer. By using the layered model, explicit occlusion handling is possible, but the kernel is restricted to linear. To overcome these limitations, Wulff and Black [11] improved the previous layered model of Bar et al. by estimating the different motions of both foreground and background layers. However, these motions are restricted to affine models and it is difficult to extend to multi-layered scenes because such task requires depth order reasoning of the layers. To sum up, segmentation-based deblurring approaches have the advantage of removing blurs caused by moving objects in dynamic scenes. However, segmentation itself is very difficult problem and remains still an

challenging issue as reported in the previous chapter. Moreover, they fail to segment complex motions like motions of people, because simple parametric motion models used in [71, 11] cannot fit the complex motions accurately.

Exemplar-based approaches were proposed in the works of Matsushita et al. [72] and Cho et al. [12]. These methods usually do not rely on accurate segmentation and deconvolution. Instead, the latent frames are rendered by interpolating lucky sharp frames that frequently exist in videos, thus avoiding severe ringing artifacts. However, the work of Matsushita et al. [72] cannot remove blurs caused by moving objects. In addition, the work of Cho et al. [12] allows only slow-moving objects in dynamic scenes because it searches sharp patches corresponding to blurry patch within a small window after registration with homography. Therefore, it cannot handle fast moving objects which have distinct motions from those of backgrounds. Moreover, since it does not use deconvolution with spatial priors but simple interpolation, it degrades mid-frequency textures such as grasses and trees, and renders smooth results.

Besides motion, defocus from limited depth-of-field (DOF) of conventional digital cameras also results in blurry effects in videos. Although shallow DOF is often used to render aesthetic images and highlight the focused objects, frequent defocus or misfocus of moving objects in video yields image degradation when the motion is large and fast. Moreover, depth variation in the scene generates spatially varying defocus blurs, making the estimation of defocus blur map is also a difficult problem. Thus many researches have studied to estimate defocus blur kernel. Most of them have approximated the kernel as Gaussian or disc model, thus the kernel estimation problem becomes a parameter (e.g. standard deviation of Gaussian blur or disc radius) estimation problem [73, 74, 55, 75].

To magnify focus differences, Bae and Durand [73] estimated defocus blur map at the edges first, and then propagated the results to other regions. However, the estimated blur map is inaccurate where the blurs are strong, since it is image-based approach and depends on the detected edges that can be localized. Similarly, Zhuo and Sim [75] propagated the amount of blur at the edges to elsewhere, that obtained by measuring the ratio between the gradients of the defocused input and re-blurred input with a Gaussian kernel. To reduce reliance on strong edges in the defocused image, Zhu et al. [55] utilized statistics of blur spectrum within the defocused image, since statistical models could be applicable where there are no strong edges. Specifically, local image statistics is used to measure the probability of defocus scale and determine the locally varying defocus blur map in a single image. However, local image statistics-based methods do not work when there are motion blurs as well as defocus blurs within a single image; Motion blurs change local statistics and yield much complex blurs combined with defocus blurs usually.

To alleviate the problems in previous works, we propose a new generalized video deblurring method that can handle not only motion blur but also defocus blur which further improves the deblurring quality significantly. Under an assumption that, the complex non-linear blur kernel can be decomposed into motion and defocus blur kernels, we estimate bidirectional optical flows to approximate pixel-wise varying motion blur kernel, scales of Gaussian blurs to approximate defocus blur kernel, and the latent frames jointly. Therefore, we can naturally handle coexisting blurs by camera shake, moving objects with complex motions, depth variations, and defocus. However, sharp frames are required to estimate accurate blur kernels, and accurate kernels are necessary to restore sharp frames. This case is a typical chicken-and-egg problem, and thus we simultaneously estimate unknown variables, latent frames, op-

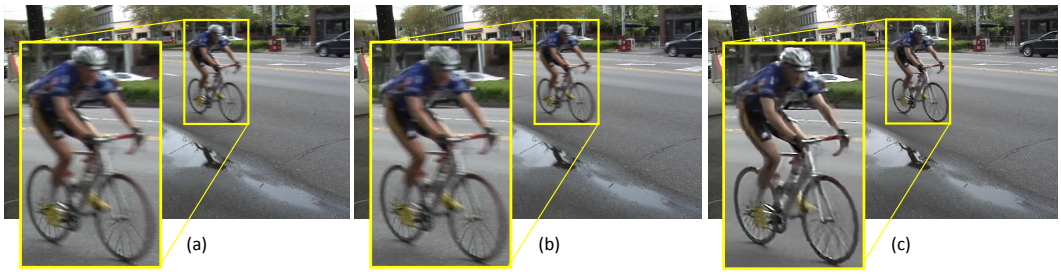


Figure 5.2: (a) Blurry frame from a dynamic scene. (b) Deblurring result by Cho et al.[12]. (c) Our result.

tical flows, and defocus blur maps. Therefore, we propose a new single energy model to solve our joint problem. We also provide a framework and efficient techniques to optimize the model. The result of our system is shown in Fig.5.1, in which the motion blurs of differently moving people and Gaussian blurs in the background are successfully removed, and accurate optical flows are jointly estimated.

Finally, we provide a new realistic blur dataset with ground truth sharp frames captured by a high-speed camera to overcome the lack of realistic ground truth dataset in this field. Although there have been some evaluation datasets for deblurring problem, they are not appropriate to carry out meaningful evaluation for the deblurring of spatially varying blurs. First, synthetically generated uniform blur kernels and blurry images from sharp images were provided in the work of Levin et al. [32]. Next, 6D camera motion in 3D space was recorded with a hardware-assisted camera to represent blur from camera shake during exposure time in the work of Köhler et al. [76]. Moreover, there have been some recent approaches to generate synthetic dataset for the sake of machine learning algorithms. To benefit from large training data, lots of blur kernels and blurry images were synthetically generated. In the work of Xu et al. [77], more than 2500 blurry images are generated using decom-

posable symmetric kernels. Schuler et al. [78] sampled naturally looking blur kernels with Gaussian Process, and Sun et al. [9] used a set of linear kernels to synthesize blurry images. However, all these datasets are generated under an assumption that the scene is static and cannot capture infinitely many and complex blurs in real world. Real blurs in dynamic scenes are complicated and spatially varying, so synthesizing realistic dataset is a difficult problem. To solve this problem, we construct a new blur dataset that provides pairs of realistically blurred videos and sharp videos with the use of a high-speed camera.

Using the proposed dataset and real challenging videos as shown in Fig.5.2, we demonstrate the significant improvements of the proposed deblurring method in both quantitatively and qualitatively. Moreover, we show empirically that more accurate optical flows are estimated by our method compared with the state-of-the-art optical flow method that can handle blurry images.

5.2 Generalized Video Deblurring

Most conventional video deblurring methods suffer from the coexistence of various motion blurs from dynamic scenes because the motions cannot be fully parameterized using global or segment-wise blur models. To make things worse, frequent defocus or misfocus of moving objects in dynamic scenes yields more complex (non-linear) blurs combined with motion blurs.

To handle these joint motion and defocus blurs, we propose a new blur model that estimates locally (pixel-wise) different blur kernels rather than global or segment-wise kernel estimation. In this work, we propose a single energy model consists of not only data and spatial regularization terms but also a temporal term and the

proposed model is expressed as follows:

$$\mathbf{E} = \mathbf{E}_{data} + \mathbf{E}_{temporal} + \mathbf{E}_{spatial}, \tag{5.1}$$

and the detailed models of each term are given in the following subsections.

5.2.1 A new data model based on kernel-parametrization

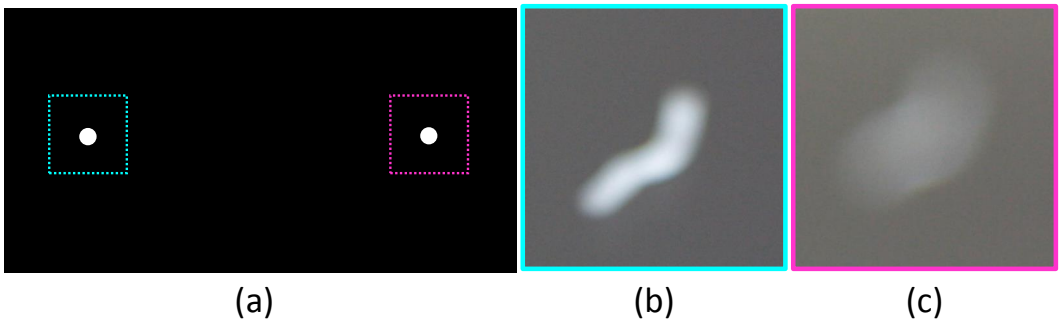


Figure 5.3: (a) Two light sources. (b) Light streak of the focused light source by camera motion. (c) Light streak of the defocused light source by camera motion.

Motion blurs are generally caused by camera shake and moving objects, and defocus blurs are mainly due to the aperture size, focal length, and the distance between camera and focused object. When these two different blurs are combined, they yield more complex blurs in real video. For example, Fig. 5.3 shows how different the blurred images are when point light sources are captured by a single moving camera with and without defocus blur. We observe that the light streak of the defocused light source is much smoother and complex in comparison with that of the focused one. Notably, the light streaks indicate the blur kernel shapes.

However, it is difficult to directly remove the complex blur in Fig. 5.3 (c). Thus, to alleviate the problem, we assume that the combined blur kernel can be decomposed into two different kernels, which are defocus blur kernel and motion blur kernel.

Our assumption holds when the depth change in the scene during exposure period is relatively small, and it is acceptable since we treat video of rather short exposure time. Thus, the underlying blurring-procedure can be modeled as sequential process of defocus blurring followed by motion blur as illustrated in Fig. 5.4.

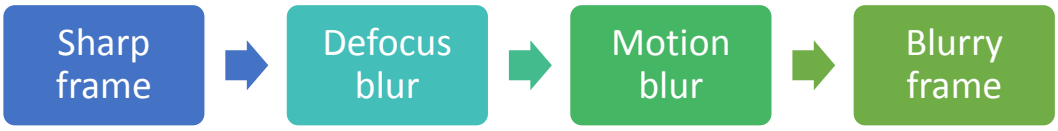


Figure 5.4: Blurring process underlying in the proposed method.

Therefore, under the assumption that the latent frames are blurred by defocus, and subsequently blurred by motion, our blur model to handle combined blurs is expressed as follows:

$$\mathbf{B}_i(\mathbf{x}) = (k_{i,\mathbf{x}} \otimes g_{i,\mathbf{x}} \otimes \mathbf{L}_i)(\mathbf{x}), \tag{5.2}$$

where $\mathbf{B}_i$ and $\mathbf{L}_i$ denote the blurry frame and the latent frame at the i_{th} frame, respectively, and $\mathbf{x}$ denotes pixel location on 2D image domain. At $\mathbf{x}$, the motion blur kernel is denoted as $k_{i,\mathbf{x}}$ and the defocus blur kernel is denoted as $g_{i,\mathbf{x}}$, and the operator $\otimes$ means convolution.

Spatially varying defocus blur $g_{i,\mathbf{x}}$ is approximated by using Gaussian or disc model in conventional works [74, 55]. Therefore, the defocus blur maps are determined by simply estimating the standard deviations of Gaussian models or the radii of disc models. In particular, local image statistics is widely used to estimate spatially varying defocus blur map. Specifically, within a uniformly blurred patch, local frequency spectrum provides information on the blur kernel and can be used to determine the likelihood of specific blur kernel [55]; thus scales of defocus blurs can be

estimated by comparing the fidelities of the likelihood model. However, it is difficult to apply this statistics-based technique when the blurry image has motion blurs in addition to defocus blurs. In Fig. 5.5, we observe that the maximum likelihood (ML) estimator used in [55] finds the optimal defocus blur kernel when a patch is blurred by only defocus blur, however ML cannot estimate true defocus kernel when a blurry patch contains motion blur as well as defocus blur. Therefore we cannot adopt local image statistics to remove defocus blurs in dynamic scenes with severe motion blurs. In this study, we approximate the pixel-wise varying defocus blur using Gaussian model as shown in Fig. 5.6 (a), and determine the locally varying standard deviation $\boldsymbol{\sigma}_i(\mathbf{x})$ of the Gaussian kernel $g_{i,\mathbf{x}}$.

Meanwhile, the motion blurs of each frame are usually approximated by global motion models such as homographies and affine models in conventional video deblurring works [71, 12, 38, 11]. However, these global motion models are only valid when the motions are globally or segment-wise rigid, and thus cannot cope with general and pixel-wise varying motion blurs in real dynamic scenes. By contrast, to deal with pixel-wise varying motion blurs, we should approximate and parametrize the locally different blur kernel, because the solution space of spatially varying kernel in video is extremely large; the dimension of unknown kernel is $W \times H \times T \times w \times h$ when the size of image is $W \times H$, length of the image sequence is T, and the size of local kernel is $w \times h$. Therefore, we approximate the motion blur kernel as piece-wise linear using bidirectional optical flows by extending suggestions in previous works [48, 8, 15], and our linearly approximated kernel is illustrated in Fig. 5.6 (b). Although our motion blur kernel is based on simple approximation, our model is valid since we assume that the videos have relatively short exposure time. The pixel-wise kernel $k_{i,\mathbf{x}}$ using bidirectional flows can be written by,

$$k_{i,\mathbf{x}}(u,v) = \begin{cases} \frac{\delta(uv_{i\to i+1} - vu_{i\to i+1})}{2\tau_i \|\mathbf{u}_{i\to i+1}(\mathbf{x})\|}, & \text{if } u \in [0, \tau_i u_{i\to i+1}], v \in [0, \tau_i v_{i\to i+1}] \\ \frac{\delta(uv_{i\to i-1} - vu_{i\to i-1})}{2\tau_i \|\mathbf{u}_{i\to i-1}(\mathbf{x})\|}, & \text{if } u \in (0, \tau_i u_{i\to i-1}], v \in (0, \tau_i v_{i\to i-1}] \\ 0, & \text{otherwise} \end{cases}, \tag{5.3}$$

where (u, v) denotes a location in 2-dimensional kernel space, and $\mathbf{u}_{i\to i+1}(\mathbf{x}) = (u_{i\to i+1}, v_{i\to i+1})$, and $\mathbf{u}_{i\to i-1}(\mathbf{x}) = (u_{i\to i-1}, v_{i\to i-1})$ denote pixel-wise bidirectional optical flows at $\mathbf{x}$ on the i_{th} frame. Camera duty cycle of the frame is τ_i and it denotes relative exposure time as used in [38]. Kronecker delta is denoted as δ.

Now, the proposed data model that handles both motion and defocus blurs is expressed as follows:

$$\mathbf{E}_{data}(\mathbf{L}, \mathbf{u}, \boldsymbol{\sigma}; \mathbf{B}) = \lambda_v \sum_i \sum_{\partial_*} \|\partial_* \mathbf{K}_i(\tau_i, \mathbf{u}_{i\to i+1}, \mathbf{u}_{i\to i-1}) \mathbf{G}_i(\boldsymbol{\sigma}_i) \mathbf{L}_i - \partial_* \mathbf{B}_i\|^2, \tag{5.4}$$

where the row vector of the motion blur kernel matrix $\mathbf{K}_i$, which corresponds to the motion blur kernel at pixel $\mathbf{x}$, is the discretized vector form of $k_{i,\mathbf{x}}$, and its elements are non-negative and their sum is equal to one. Similarly, the row vector of the defocus blur kernel matrix $\mathbf{G}_i$ is associated with $g_{i,\mathbf{x}}$ and $\boldsymbol{\sigma}_i$ denotes the scales (standard deviations of Gaussian kernel) of defocus blurs. Linear operator ∂_* denotes the Toeplitz matrices corresponding to the partial (e.g., horizontal and vertical) derivative filters. Parameter λ_v controls the weight of the data term, and $\mathbf{L}$, $\mathbf{u}$, $\boldsymbol{\sigma}$, and $\mathbf{B}$ denote the set of latent frames, optical flows, scales of defocus blurs and blurry frames, respectively.

As we handle motion and defocus blurs simultaneously in the proposed model, ours outperforms the state-of-the-art defocus blur map estimator [14] when there exist both motion and defocus blurs s shown in Fig. 5.7. Even when the motion blurs are not existing, we achieve competitive result as shown in Fig. 5.8 with the aid of joint estimation of the latent frames. Moreover, using our pixel-wise motion blur kernel approximation, we can easily manage multiple different motion blurs in a single frame, unlike conventional methods. The superiority of our locally varying kernel model is shown in Fig. 5.9. Our kernel model fits motion blurs from differently moving objects and camera shake much better than the conventional homography-based model.

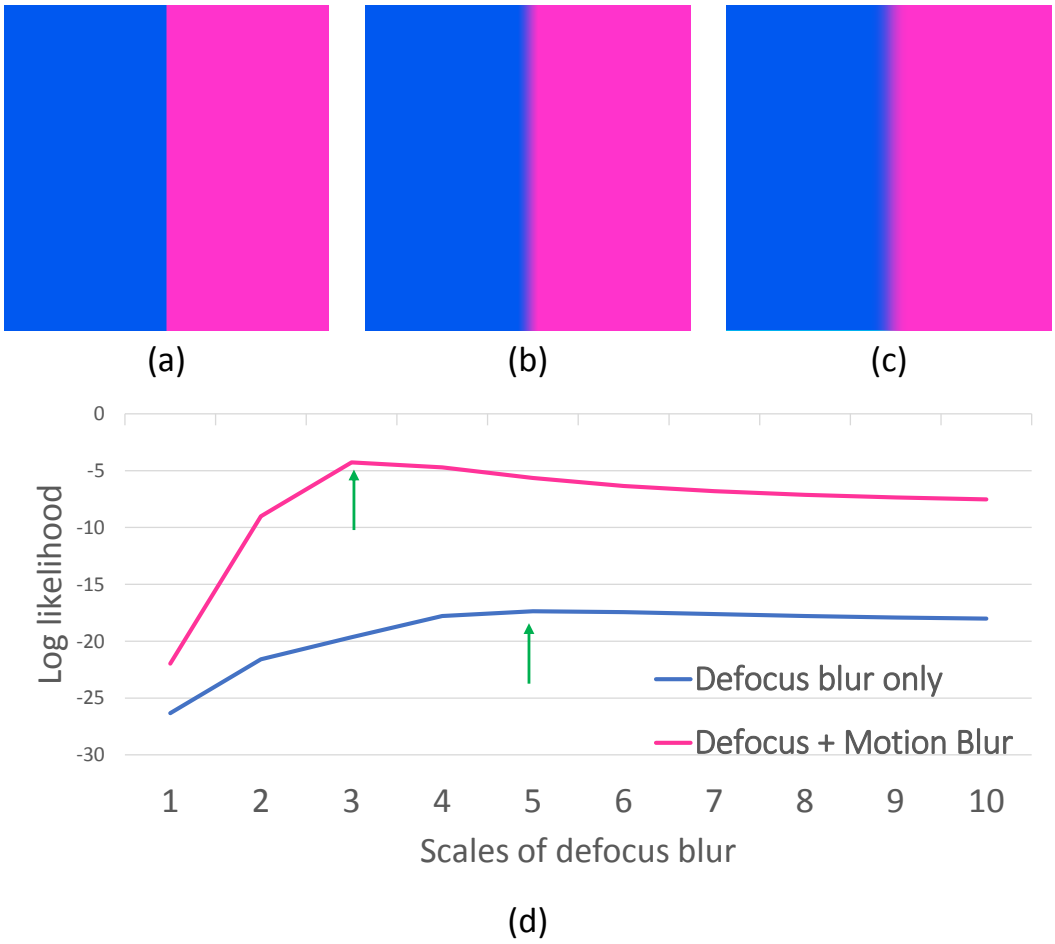


Figure 5.5: (a) A sharp patch. (b) A patch blurred by defocus blur only (Gaussian blur with standard deviation 5). (c) A patch blurred by defocus blur (Gaussian blur with standard deviation 5) and motion blur (linear kernel with length 11). (d) Comparisons of fidelities at the center of the blurry patches by changing the scale of defocus blur. The ground truth scale of the defocus blur is 5 and the arrows indicate peaks estimated by ML estimator.

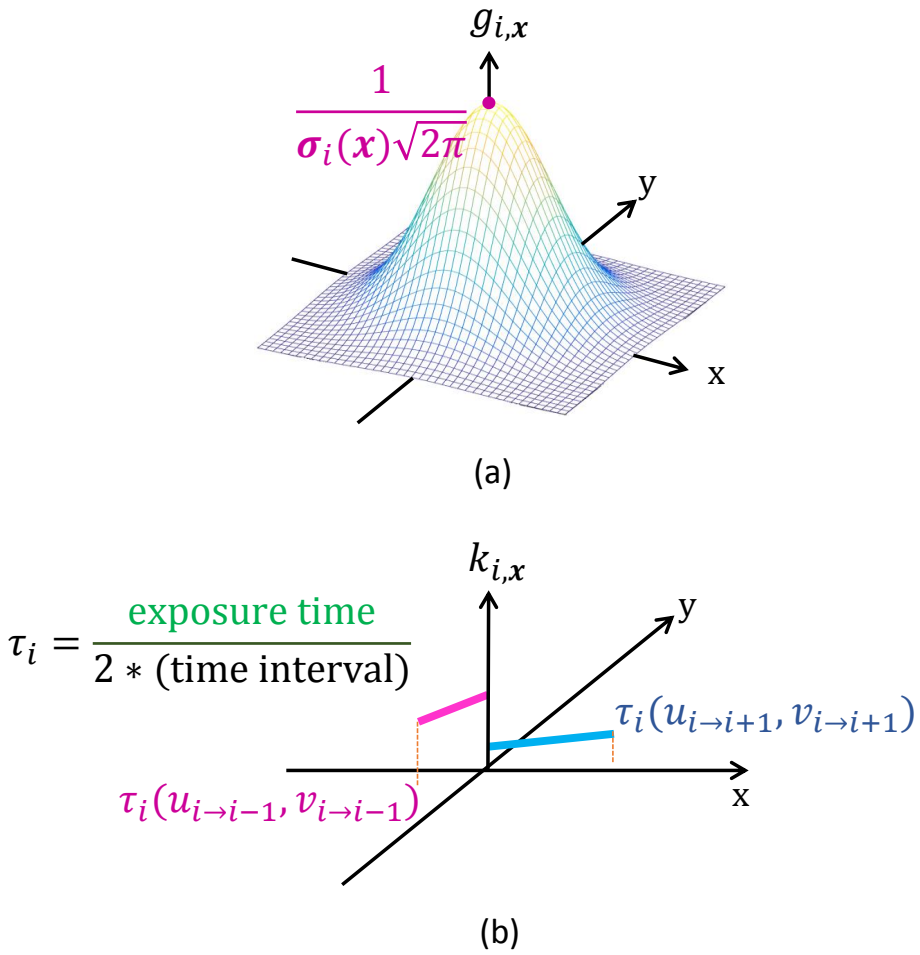


Figure 5.6: Defocus blur kernel and motion blur kernel. (a) Gaussian blur kernel with standard deviation $\boldsymbol{\sigma}_i(\mathbf{x})$ at a pixel location $\mathbf{x}$ to handle blur from defocus. (b) Bidirectional optical flows and corresponding piece-wise linear motion blur kernel at a pixel location $\mathbf{x}$.

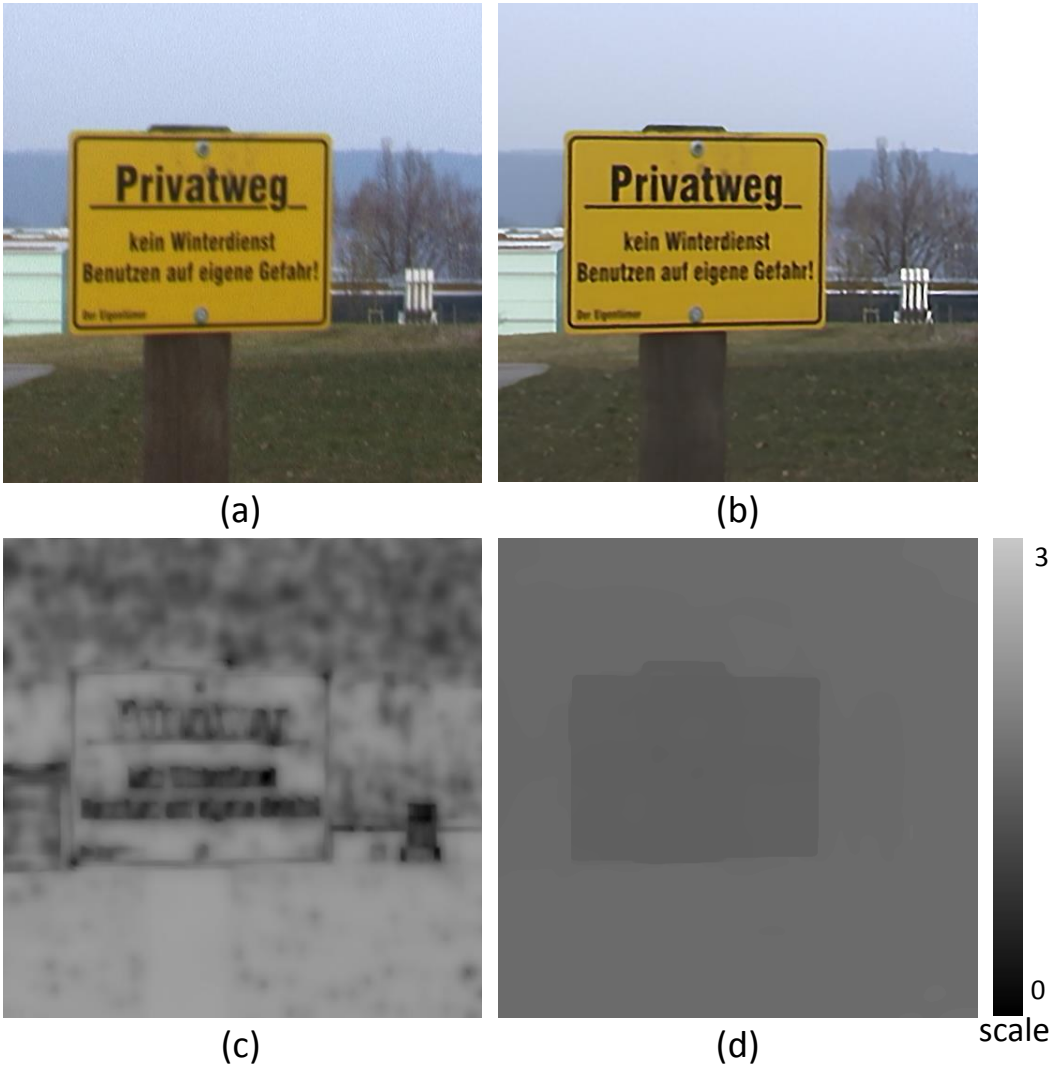


Figure 5.7: (a) A real blurry frame. (b) Our jointly estimated latent frame. (c) Defocus blur map of Shi et al. [14]. (d) Our defocus blur map.

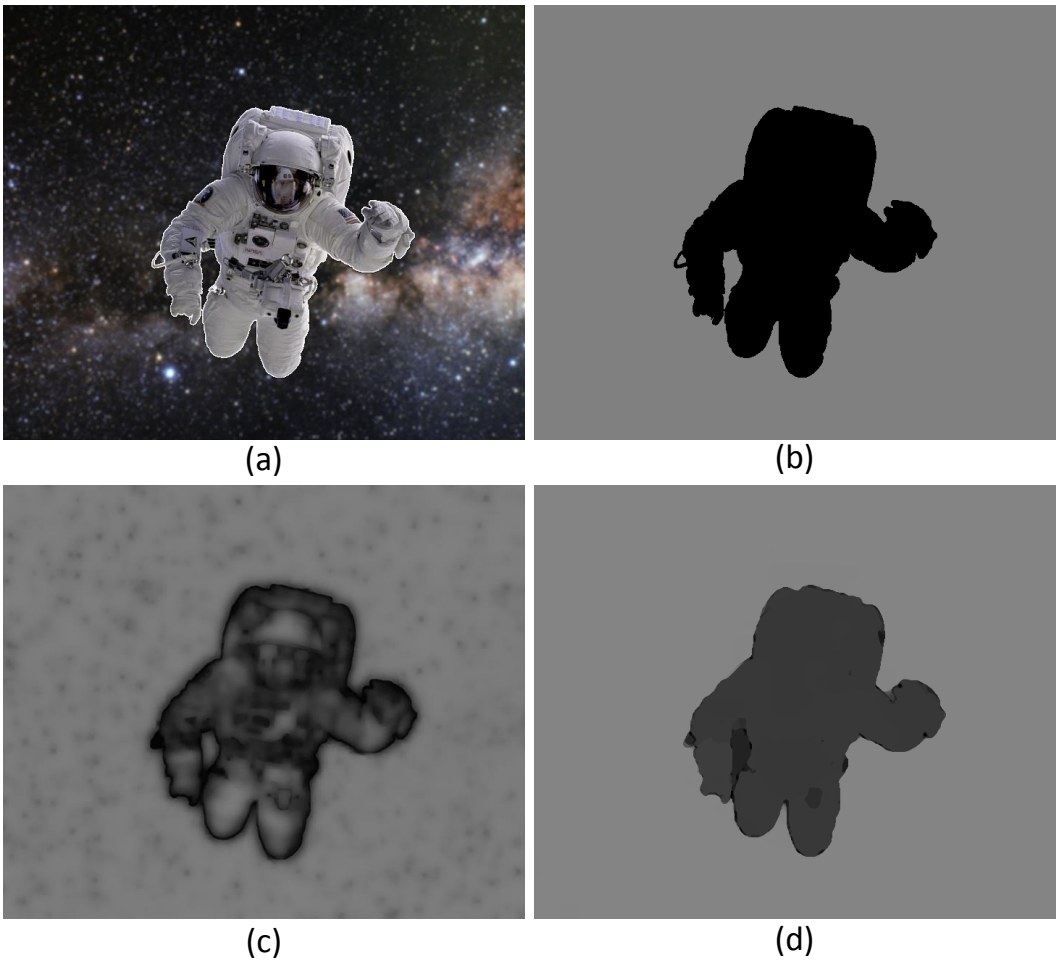


Figure 5.8: (a) Partially blurred image which has sharp foreground and blurred background by Gaussian blur. (b) Ground truth blur map. (c) Defocus blur map of Shi et al. [14]. (d) Our defocus blur map.

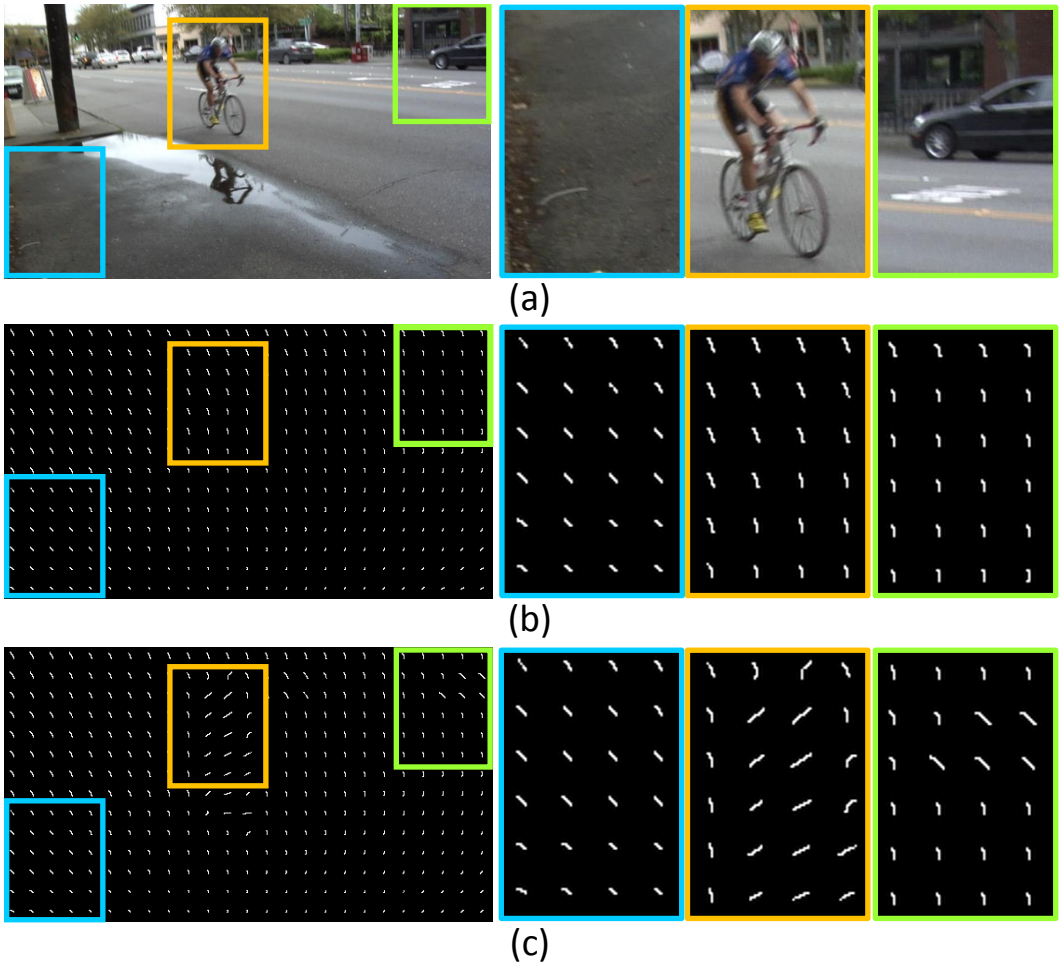


Figure 5.9: (a) A blurry frame of a video in dynamic scene. (b) Locally varying kernel using homography. (c) Our pixel-wise varying motion blur kernel using bidirectional optical flows.

5.2.2 A new optical flow constraint and temporal regularization

To remove locally varying motion blurs, we employ bidirectional optical flow model in (5.4). However, for optical flow estimation, conventional optical flow constraints such as brightness constancy and gradient constancy can not be utilized directly, since such constraints are not valid between two blurry frames. Thus a blur-aware optical flow estimation method from blurry images has been proposed by Portz et al. [15], and this method is based on the commutative law of shift-invariant kernels such that the brightness of the corresponding points is constant after convolving the blur kernel of each image with the other image. However, the commutative law does not hold when the motion is not translational and when the blur kernels vary spatially. Therefore, this approach only works when the motion is very smooth.

To address this problem, we propose a new model that estimates optical flow between two latent sharp frames to enable abrupt changes in motions and the blur kernels. Specifically, our model is based on the conventional optical flow constraint between latent frames, that is, brightness constancy. By doing so, we need not restrict our motion blur kernels to be shift invariant, and the formulation of the proposed model is given by,

$$\mathbf{E}_{temporal}(\mathbf{L},\mathbf{u}) = \mu \sum_{i} \sum_{j=-N}^{N} \sum_{\mathbf{x}} |\mathbf{L}_i(\mathbf{x}) - \mathbf{L}_{i+j}(\mathbf{x}+\mathbf{u}_{i\rightarrow i+j})|, \tag{5.5}$$

where n denotes the index of neighboring frames of the frame at i, and the parameter μ controls the weight. We apply the robust L_1 norm for robustness against outliers and occlusions.

Notably, a major difference between the proposed model and the conventional optical flow estimation methods is that our method is a joint solution. That is, the

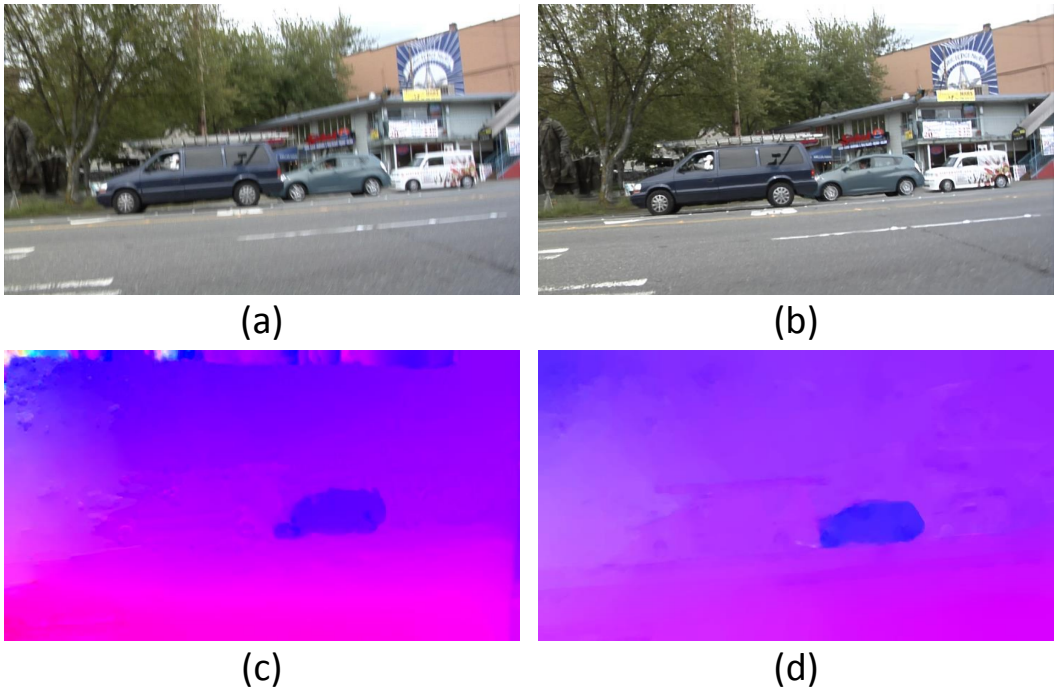


Figure 5.10: (a) A real blurry frame. (b) Our jointly estimated latent frame. (c) Color coded optical flow from [15]. (d) Our optical flow.

latent frames and optical flows should be solved simultaneously in our model. Therefore, the proposed model in (5.5) restores the latent frames which are temporally coherent among neighboring frames, and estimates optical flows between neighboring frames, jointly. Thus we can estimate accurate flows at the motion boundaries as shown in Fig. 5.10. Notice that, our flows at the motion boundaries of moving car is much clearer in comparison with the blur-aware flow estimation method by [15].

5.2.3 Spatial regularization

To alleviate the difficulties of highly ill-posed deblurring, optical flow estimation, and defocus blur map estimation problems, it is important to adopt well-suited spatial regularizers. In this work, we enforce spatial coherence to penalize spatial

fluctuations while allowing discontinuities in latent frames, flow fields, and defocus blur maps. With an assumption that spatial priors for the latent frames, optical flows, and defocus blur maps are independent, we can formulate for the spatial regularization as follows:

$$\begin{aligned}\mathbf{E}_{spatial}(\mathbf{L},\mathbf{u},\boldsymbol{\sigma}) = &\sum_i |\nabla \mathbf{L}_i| + \nu_\sigma \sum_i \sum_{\mathbf{x}} \zeta_i(\mathbf{x}) |\nabla \boldsymbol{\sigma_i}(\mathbf{x})| + \\ &\nu_u \sum_i \sum_{j=-N}^{N} \sum_{\mathbf{x}} \zeta_i(\mathbf{x}) |\nabla \mathbf{u}_{i\rightarrow i+j}(\mathbf{x})|),\end{aligned} \tag{5.6}$$

where parameters ν_σ and ν_u control the weights of the second and third terms.

The first term in (5.6) denotes the spatial regularization term for the latent frames. Although more sparse L_p norm (e.g. $p = 0.8$) fits the gradient statistics of natural sharp images better [26, 79, 25], we use conventional total variation (TV) based regularization [80, 10, 8], as TV model is computationally less expensive and easy to minimize. The second and third terms enforce spatial smoothness for defocus blur maps and optical flows, respectively. These regularizers are also based on TV model, and coupled with an edge-map to preserve discontinuities at the edges in both vector fields. Similar to the edge-map used in conventional optical flow estimation method [63], our edge-map is expressed as follows:

$$\zeta_i(\mathbf{x}) = \exp(-(\frac{|\nabla \mathbf{L}_i^0(\mathbf{x})|^2}{v_I})), \tag{5.7}$$

where the fixed parameter v_I controls the weight of the edge-map, and $\mathbf{L}_i^0$ is an initial latent image in the iterative optimization framework in Sec. 5.3.

5.3 Optimization Framework

Under the condition that the camera duty cycle τ_i is known, by combining $\mathbf{E}_{data}$, $\mathbf{E}_{temporal}$, and $\mathbf{E}_{spatial}$, we have the final objective function as follows:

$$\begin{aligned}\min_{\mathbf{L},\mathbf{u},\boldsymbol{\sigma}} \ & \lambda_v \sum_i \sum_{\partial_*} \|\partial_* \mathbf{K}_i(\mathbf{u}_{i\to i+1}, \mathbf{u}_{i\to i-1})\mathbf{G}_i(\boldsymbol{\sigma_i})\mathbf{L}_i - \partial_* \mathbf{B}_i\|^2 + \\ & \mu \sum_i \sum_{j=-N}^{N} \sum_{\mathbf{x}} |\mathbf{L}_i(\mathbf{x}) - \mathbf{L}_{i+j}(\mathbf{x} + \mathbf{u}_{i\to i+j})| + \\ & \sum_i |\nabla \mathbf{L}_i| + \nu_\sigma \sum_i \sum_{\mathbf{x}} \zeta_i(\mathbf{x}) |\nabla \boldsymbol{\sigma_i}| + \nu_u \sum_i \sum_{j=-N}^{N} \sum_{\mathbf{x}} \zeta_i(\mathbf{x}) |\nabla \mathbf{u}_{i\to i+j}|).\end{aligned} \tag{5.8}$$

Notably, in contrast with the work of Cho et al. [12] that performs multiple approaches sequentially, our model finds a solution by minimizing the proposed single objective function in (5.8). However, because of its non-convexity, our model needs to adopt a practical optimization method to obtain an approximated solution. Therefore, we divide the original problem into several simple sub-problems and then use conventional iterative and alternating optimization techniques [17, 8, 11] to minimize the original non-convex objective function. In the following subsections, we introduce efficient solvers and describe how to estimate unknowns $\mathbf{L}$, $\mathbf{u}$, and $\boldsymbol{\sigma}$ alternatively.

5.3.1 Sharp video restoration

While the motion blur kernels, and the defocus blur kernels are fixed, the objective function becomes convex with respect to $\mathbf{L}$, and it can be expressed as follows:

$$\begin{aligned}\min_{\mathbf{L}} \quad & \lambda_v \sum_i \sum_{\partial_*} \|\partial_* \mathbf{K}_i \mathbf{G}_i \mathbf{L}_i - \partial_* \mathbf{B}_i\|^2 + \sum_i |\nabla \mathbf{L}_i| + \\ & \mu \sum_i \sum_{j=-N}^{N} \sum_{\mathbf{x}} |\mathbf{L}_i(\mathbf{x}) - \mathbf{L}_{i+j}(\mathbf{x} + \mathbf{u}_{i \to i+j})|.\end{aligned} \tag{5.9}$$

To restore the latent frames $\mathbf{L}$, we adopt the conventional convex optimization method proposed in [29], and derive the primal-dual update scheme as follows:

$$\begin{cases} \mathbf{s}^{m+1} = \frac{\mathbf{s}^m + \eta_L \mathbf{A}\mathbf{L}^m}{\max(\mathbf{1}^T,\ \mathrm{abs}(\mathbf{s}^m + \eta_L \mathbf{A}\mathbf{L}^m))} \\ \mathbf{q}^{m+1} = \frac{\mathbf{q}^m + \eta_L \mu \mathbf{D}\mathbf{L}^m}{\max(\mathbf{1}^T,\ \mathrm{abs}(\mathbf{q}^m + \eta_L \mu \mathbf{D}\mathbf{L}^m))} \\ \mathbf{L}^{m+1} = \arg\min_{\mathbf{L}^{m+1}} \lambda_v \sum_i \sum_{\partial_*} \|\partial_* \mathbf{K}_i \mathbf{G}_i \mathbf{L}_i^{m+1} - \partial_* \mathbf{B}_i\|^2 + \\ \frac{\|\mathbf{L}^{m+1} - (\mathbf{L}^m - \epsilon_L(\mathbf{A}^T \mathbf{s}^{m+1} + \mu \mathbf{D}^T \mathbf{q}^{m+1}))\|^2}{2\epsilon_L}, \end{cases} \tag{5.10}$$

where $m \geq 0$ indicates the iteration number, and $\mathbf{L}^m$ denotes concatenation of all latent frames in a vector form. And $\mathbf{s}$ and $\mathbf{q}$ denote the dual variables. Parameters η_L and ϵ_L denote the update steps. The linear operator $\mathbf{A}$ calculates the spatial difference between neighboring pixels, and the operator $\mathbf{D}$ calculates the temporal differences among neighboring frames using fixed optical flows. The last formulation in (5.10) is to update and optimize the primal variable $\mathbf{L}^{m+1}$, and we apply the conjugate gradient method to minimize it, since it is a quadratic function. Notably, division operators used in the update steps denote the element-wise division.

5.3.2 Optical flows estimation

While the latent frames and the defocus blur kernels are fixed, the objective function in (5.8) becomes motion estimation model. However, our motion estimation model is non-convex, because the temporal coherence term $\mathbf{E}_{temporal}$ and the data term $\mathbf{E}_{data}$ are non-convex. Thus, we denote those two terms as a non-convex function $\rho_u(.)$ to simplify as follows:

$$\begin{aligned}\rho_u(\mathbf{u}) =& \mu \sum_i \sum_{j=-N}^{N} \sum_{\mathbf{x}} |\mathbf{L}_i(\mathbf{x}) - \mathbf{L}_{i+j}(\mathbf{x} + \mathbf{u}_{i\to i+j})| + \\ & \lambda_v \sum_i \sum_{\partial_*} \|\partial_* \mathbf{K}_i(\mathbf{u}_{i\to i+1}, \mathbf{u}_{i\to i-1})\mathbf{G}_i\mathbf{L}_i - \partial_* \mathbf{B}_i\|^2,\end{aligned} \tag{5.11}$$

and we convexify the non-convex function $\rho_u(.)$ by applying the first-order Taylor expansion to find the optimal optical flows $\mathbf{u}$. Similar to the technique used in [8], we linearize the function near an initial $\mathbf{u}_0$ in the iterative process as follows:

$$\rho_u(\mathbf{u}) \approx \rho_u(\mathbf{u}_0) + \nabla\rho_u(\mathbf{u}_0)^T(\mathbf{u} - \mathbf{u}_0). \tag{5.12}$$

In doing so, (5.8) can be approximated as a convex function w.r.t $\mathbf{u}$ for being fixed $\mathbf{G}$ and $\mathbf{L}$ as follows:

$$\min_{\mathbf{u}} \rho_u(\mathbf{u}_0) + \nabla\rho_u(\mathbf{u}_0)^T(\mathbf{u} - \mathbf{u}_0) + \nu_u \sum_i \sum_{j=-N}^{N} \sum_{\mathbf{x}} \zeta_i(\mathbf{x})|\nabla \mathbf{u}_{i\to i+j}|. \tag{5.13}$$

Now, we can apply the convex optimization technique in [29] to the approximated convex function, and the primal-dual update process is expressed as follows:

$$\begin{cases} \mathbf{p}^{m+1} = \frac{\mathbf{p}^m + \eta_u(\nu_u \mathbf{W}_u \mathbf{A}_u)\mathbf{u}^m}{\max(\mathbf{1}^T,\ \mathrm{abs}(\mathbf{p}^m + \eta_u(\nu_u \mathbf{W}_u \mathbf{A}_u)\mathbf{u}^m))} \\ \\ \mathbf{u}^{m+1} = (\mathbf{u}^m - \epsilon_u(\nu_u \mathbf{W}_u \mathbf{A}_u)^T \mathbf{p}^{m+1}) - \epsilon_u \nabla \rho_u(\mathbf{u}_0), \end{cases} \tag{5.14}$$

where $\mathbf{p}$ denotes the dual variable of $\mathbf{u}$. Weighting matrix $\mathbf{W}_u$ is diagonal, and its sub-matrix associated with $\mathbf{u}_{i \to i+j}$ is defined as $\mathrm{diag}(\zeta_i)$. The linear operator $\mathbf{A}_u$ calculates the spatial difference between four nearest neighboring pixels, and parameters η_u and ϵ_u denote the update steps. Notice that, division operators in the formulation denote the element-wise division.

5.3.3 Defocus blur map estimation

While the latent frames and the motion blur kernelsare fixed, we can estimate the defocus blur maps. Because the data term is non-convex, an approximation technique is required to solve the sub-problem. Similar to our optical flows estimation technique, we approximate and convexify the function using linearization.

First, we define a non-convex data function $\rho_{\boldsymbol{\sigma}}(.)$ and approximate it near an initial $\boldsymbol{\sigma}_0$ as follows:

$$\begin{aligned} \rho_\sigma(\boldsymbol{\sigma}) &= \lambda_v \sum_i \sum_{\partial_*} \|\partial_* \mathbf{K}_i \mathbf{G}_i(\boldsymbol{\sigma}_i)\mathbf{L}_i - \partial_* \mathbf{B}_i\|^2 \\ &\approx \rho_\sigma(\boldsymbol{\sigma}_0) + \nabla \rho_\sigma(\boldsymbol{\sigma}_0)^T(\boldsymbol{\sigma} - \boldsymbol{\sigma}_0), \end{aligned} \tag{5.15}$$

and the approximated convex function for defocus blur map estimation is given by,

$$\min_{\boldsymbol{\sigma}}\ \rho_\sigma(\boldsymbol{\sigma}_0) + \nabla \rho_\sigma(\boldsymbol{\sigma}_0)^T(\boldsymbol{\sigma} - \boldsymbol{\sigma}_0) + \nu_\sigma \sum_i \sum_{\mathbf{x}} \zeta_i(\mathbf{x}) |\nabla \boldsymbol{\sigma}_i|. \tag{5.16}$$

Figure 5.11: Temporally consistent optical flows over three frames.

Similarly, (5.16) can be optimized by using [29], and the primal-dual update formulation is given by,

$$\begin{cases} \mathbf{r}^{m+1} = \frac{\mathbf{r}^m + \eta_\sigma(\nu_\sigma \mathbf{W}_\sigma \mathbf{A})\boldsymbol{\sigma}^m}{\max(\mathbf{1},\ \mathrm{abs}(\mathbf{r}^m + \eta_\sigma(\nu_\sigma \mathbf{W}_\sigma \mathbf{A})\boldsymbol{\sigma}^m))} \\ \boldsymbol{\sigma}^{m+1} = (\boldsymbol{\sigma}^m - \epsilon_\sigma(\nu_\sigma \mathbf{W}_\sigma \mathbf{A})^T \mathbf{r}^{m+1}) - \epsilon_\sigma \nabla \rho_\sigma(\boldsymbol{\sigma}_0), \end{cases} \tag{5.17}$$

where $\mathbf{r}$ denotes the dual variable of $\boldsymbol{\sigma}$ on the vector field. Weighting matrix $\mathbf{W}_\sigma$ is diagonal, and its sub-matrix associated with $\boldsymbol{\sigma}_i$ is defined as $\mathrm{diag}(\zeta_i)$. Parameters η_σ and ϵ_σ denote the update steps, and division operator means the element-wise division.

5.4 Implementation Details

To handle large blurs and guide fast convergence, we implement our algorithm on the conventional coarse-to-fine framework with empirically determined parameters. In the coarse-to-fine framework, we build image pyramid with 17 levels for a high-definition (1280x720) video, and use the scale factor 0.9. Moreover, to re-

duce the number of unknowns in optical flows, we only estimate $\mathbf{u}_{i\to i+1}$ and $\mathbf{u}_{i\to i-1}$. For example, we approximate $\mathbf{u}_{i\to i+2}$ using $\mathbf{u}_{i\to i+1}$ and $\mathbf{u}_{i+1\to i+2}$, since it satisfies $\mathbf{u}_{i\to i+2} = \mathbf{u}_{i\to i+1} + \mathbf{u}_{i+1\to i+2}$ as illustrated in Fig. 5.11. We can easily apply this for $n \neq 1$.

The overall process of our algorithm is in Algorithm 4. Further details on initialization, estimating the duty cycle τ_i and post-processing step that reduces artifacts are given in the following subsections.

Algorithm 4 Overview of the proposed method

Input: Blurry frames $\mathbf{B}$

Output: Latent frames $\mathbf{L}$, optical flows $\mathbf{u}$, and defocus blur maps $\boldsymbol{\sigma}$

1: Initialize $\mathbf{u}$, τ_i, and $\boldsymbol{\sigma}$. (Sec. 5.4.1)
2: Build image pyramid.
3: Restore $\mathbf{L}$ with fixed $\mathbf{u}$ and $\boldsymbol{\sigma}$. (Sec. 5.3.1)
4: Estimate $\mathbf{u}$ with fixed $\mathbf{L}$ and $\boldsymbol{\sigma}$. (Sec. 5.3.2)
5: Estimate $\boldsymbol{\sigma}$ with fixed $\mathbf{L}$ and $\mathbf{u}$. (Sec. 5.3.3)
6: Detect occlusion and perform post-processing. (Sec 5.4.2)
7: Propagate variables to the next pyramid level if exists.
8: Repeat steps 3-7 from coarse to fine pyramid level.

5.4.1 Initialization and duty cycle estimation

In this study, we assume that the camera duty cycle τ_i is known for every frame. However, when we conduct deblurring with conventional datasets, which do not provide exposure information, we apply the technique proposed in [12] to estimate the duty cycle. Contrary to the original method [12], we use optical flows instead of homographies to obtain initially approximated blur kernels. Therefore, we first

Figure 5.12: (a) Ground truth sharp frames. (b) Generated blurry frames. Spatially varying blurs by object motions and camera shakes are synthesized realistically.

estimate flow fields from blurry images with [81], which runs in near real-time. We then use them as initial flows and approximate the kernels to estimate the duty cycle. Moreover, we use $\boldsymbol{\sigma}_i(\mathbf{x}) = 0.25$ for initial scale of defocus blur.

5.4.2 Occlusion detection and refinement

Our pixel-wise kernel estimation naturally results in approximation error and it causes problems such as ringing artifacts. Specifically, our data model in (5.4), and temporal coherence model in (5.5) are invalid at occluded regions.

To reduce such artifacts from kernel approximation errors and occlusions, we use

spatio-temporal filtering as a post-processing:

$$\mathbf{L}_i^{m+1}(\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \sum_{j=-N}^{N} \sum_{\mathbf{y}} w_{i,j}(\mathbf{x},\mathbf{y}) \cdot \mathbf{L}_{i+j}^{m}(\mathbf{y}), \tag{5.18}$$

where $\mathbf{y}$ denotes a pixel in the 3x3 neighboring patch at location $(\mathbf{x}+\mathbf{u}_{i\to i+j})$ and Z is the normalization factor (e.g. $Z(\mathbf{x}) = \sum_{j=-N}^{N} \sum_{\mathbf{y}} w_{i,j}(\mathbf{x},\mathbf{y})$). Notably, we enable $j = 0$ in (5.18) for spatial filtering. Our occlusion-aware weight $w_{i,j}$ is defined as follows:

$$w_{i,j}(\mathbf{x},\mathbf{y}) = o_{i,j}(\mathbf{x},\mathbf{y}) \cdot \exp\left(-\frac{\|P_i(\mathbf{x}) - P_{i+j}(\mathbf{y})\|^2}{2\sigma_w^2}\right), \tag{5.19}$$

where occlusion state $o_{i,j}(\mathbf{x},\mathbf{y}) \in \{0.01, 1\}$ is determined by cross-checking forward and backward flows similar to the occlusion detection technique used in [82]. The 5x5 patch $P_i(\mathbf{x})$ is centered at $\mathbf{x}$ in frame i. The similarity control parameter σ_w is fixed as $\sigma_w = 25/255$.

5.5 Motion Blur Dataset

Because conventional evaluation datasets for deblurring [32, 76] are generated under static scene assumption, complex and spatially varying blurs in dynamic scenes are not provided. Therefore, in this section, we provide a new method generating blur dataset for the quantitative evaluation of non-uniform video deblurring algorithms and later studies of learning-based deblurring approaches.

5.5.1 Dataset generation

As we assume motion blur kernels can be approximated by using bidirectional optical flows in (5.3), we can generate blurry frames adversely by averaging consecutive

sharp frames whose relative motions between two neighboring frames are smaller than one pixel. In doing so, we use GOPRO Hero4 hand-held camera which supports taking 240 fps video of 1280x720 resolution. Similar approach was introduced in [83], which uses a high-speed camera to generate blurry images. However, they captured only linearly moving objects with the fixed (static) camera.

Our captured videos include various dynamic scenes as well as static scenes. We calculate the average of k successive frames to generate a single blurry frame. By averaging k successive frames, realistic motion blurs from both moving objects and the camera shake can be rendered in the blurry frame, and the $240/k$ fps blurry video can be generated (i.e. 16 fps video is generated by averaging every 15 frames). Notably, ground truth sharp frame is chosen to be the mid-frame used in averaging, since we aim to restore the latent frame captured in the middle of exposure time as shown in fig. 5.6. Thus the duty cycle is $\tau_i = 0.5$, in our whole dataset. The videos are recorded with caution so that the motions should be no greater than 1 pixel between two neighboring frames to render more smooth and realistic blurry frame.

Our dataset mainly captured outdoor scenes to avoid flickering effect of fluorescent light which occurs when we capture indoor scenes with the high-speed camera. We captured numerous scenes in both dynamic and static environments, and each frame has HD (1280x720) size. In Fig. 5.12, some examples of our ground truth frames and rendered blurry frames are shown. We can see that the generated blurs are locally varying according to the depth changes and moving objects.

5.6 Experiments

In this section, we empirically demonstrate the superiority of our method over conventional methods.

In Table 5.2, our deblurring results are quantitatively evaluated with the proposed motion blur dataset. For evaluation, we use fixed parameters and the values are $\lambda_v = 250$, $\mu = 2$, $\nu_u = \nu_\sigma = 0.08\lambda_v$, $v_I = (\frac{25}{255})^2$, and $N = 2$. Since the source codes of other video deblurring methods that can handle non-uniform blur are not available, we quantitatively compare our method by changing settings in terms of the PSNR and SSIM values. To demonstrate the good performance of the proposed method in removing defocus blurs as well as motion blurs, we added different strengths of Gaussian blurs ($\sigma = 1, 1.5, 2$) to the original sharp videos before averaging. Using these datasets including both motion and defocus blurs, we verify that, the proposed approach improves the deblurring results significantly in terms of PSNR and SSIM by removing defocus blurs as well as motion blurs.

Moreover, in Fig. 5.13, qualitative comparisons using our dataset are shown. Ours restores the edges of buildings, letters, and moving persons, clearly. However, we observe some failure cases in our results. In Fig. 5.14, we could not estimate accurate motions of fast moving hand, and thus fail in deblurring. Notice that, it is difficult to estimate motion flows of small structure with distinct motions in the coarse-to-fine framework as in [7].

In Table. 5.1, we compare all the proposed methods. For evaluation, some sequences from our synthetic datasets are used, and we compare PSNR values of selected 10 frames in the sequences. As expected, the proposed video deblurring method outperforms our previously methods with the aid of accurate kernel estima-

Seq.	segmentation	exemplar	parametrization (single)	parametrization (video)
#1	20.25	26.48	29.40	30.55
#2	23.23	28.09	30.60	31.41
#3	22.46	26.22	26.54	33.61
#4	22.47	25.69	25.95	30.52
#5	22.20	23.26	26.53	23.62
#6	21.56	24.68	23.09	27.79
#7	23.54	25.73	28.57	34.78
#8	20.69	25.30	26.18	29.81
#9	20.35	25.03	24.82	29.07
#10	20.84	24.02	23.08	27.68
Avg.	21.76	25.45	26.48	**29.88**

Table 5.1: Deblurring performance evaluations in terms of PSNR values. Our methods with segmentation, exemplar, and kernel-parametrization are compared.

tion and additional temporal information.

Next, we qualitatively compare our deblurring results with those of the state-of-the-art exemplar-based method [12] with the videos used in their work. As shown in Fig. 5.15, the captured scenes are dynamic and contain multiple moving objects. The method [12] fails in restoring the moving objects, because the object motions are large and distinct from the backgrounds. By contrast, our results show better performances in deblurring moving objects and backgrounds. Notably, the exemplar-based approach also fails in handling large blurs, as shown in Fig. 5.16, as the initially estimated homographies in the largely blurred images are inaccurate. Moreover, this approach renders excessively smooth results for mid-frequency textures such as trees, since the method is based on interpolation without spatial prior for latent frames.

Figure 5.13: Comparative deblurring results using our blur dataset. (a) Ground truth sharp frames. (b) Generated blurry frames including both motion and defocus blurs. (c) Deblurring results from without handling defocus blurs. (d) Deblurring results handling defocus blurs.

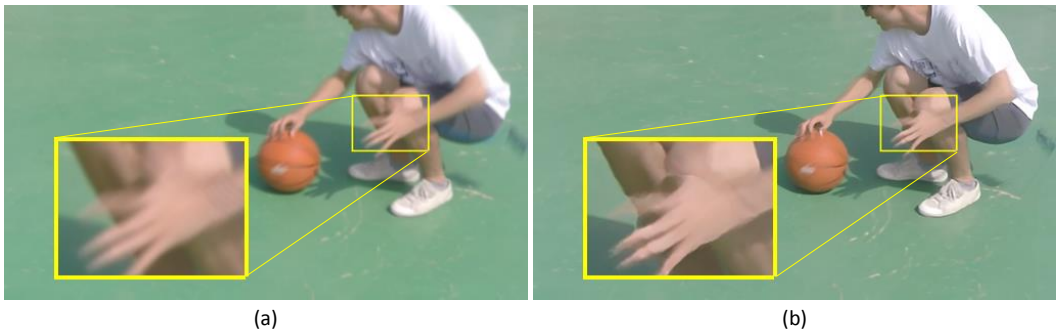


Figure 5.14: A failure case. (a) A blurry frame in the proposed dataset. (b) Our deblurring result.

Table 5.2: Deblurring performance evaluations in terms of PSNR (SSIM). The matrix $\mathbf{I}$ denotes identity matrix.

	($\sigma = 1.0$)		($\sigma = 1.5$)		($\sigma = 2.0$)	
Seq.	ours ($\mathbf{G}_i = \mathbf{I}$)	ours (full)	ours ($\mathbf{G}_i = \mathbf{I}$)	ours (full)	ours ($\mathbf{G}_i = \mathbf{I}$))	ours (full)
#1	28.02 (0.8552)	27.97 (0.8608)	27.53 (0.8374)	27.85 (0.8473)	26.84 (0.8135)	27.25 (0.8250)
#2	26.62 (0.8407)	26.97 (0.8519)	25.18 (0.7809)	25.61 (0.7982)	23.91 (0.7160)	24.28 (0.7334)
#3	32.89 (0.9182)	34.07 (0.9353)	31.27 (0.8849)	32.49 (0.9063)	29.94 (0.8501)	30.75 (0.8680)
#4	36.77 (0.9684)	36.60 (0.9675)	36.50 (0.9665)	36.61 (0.9668)	36.12 (0.9639)	36.38 (0.9649)
#5	24.15 (0.7260)	24.01 (0.7306)	23.05 (0.6598)	24.29 (0.7252)	24.03 (0.6989)	24.18 (0.7077)
#6	27.04 (0.8577)	27.51 (0.8712)	25.18 (0.7880)	25.70 (0.8078)	23.80 (0.7187)	24.20 (0.7374)
#7	29.07 (0.8863)	29.62 (0.8986)	27.31 (0.8360)	27.93 (0.8529)	25.87 (0.7809)	26.35 (0.7973)
#8	27.93 (0.8828)	28.46 (0.8940)	26.05 (0.8320)	26.62 (0.8468)	24.52 (0.7768)	24.96 (0.7914)
#9	30.38 (0.8793)	30.90 (0.8919)	29.05 (0.8427)	29.79 (0.8593)	27.81 (0.8028)	28.43 (0.8188)
#10	29.13 (0.8843)	29.61 (0.8961)	27.61 (0.8432)	28.23 (0.8587)	26.25 (0.7982)	26.76 (0.8131)
#11	32.42 (0.9471)	32.77 (0.9519)	30.47 (0.9283)	31.64 (0.9383)	28.81 (0.9064)	29.74 (0.9171)
Avg.	29.49 (0.8769)	29.86 (0.8863)	28.11 (0.8363)	28.80 (0.8863)	27.08 (0.8023)	27.57 (0.8158)

We also compare our method with the state-of-the-art segmentation-based deblurring approach [11]. The test video is shown in Fig. 5.17, which is a bilayer scene used in their work. Although the bi-layer scene is a good example to verify the performance of the layered model, inaccurate segmentation near the boundaries causes serious artifacts in the restored frame. By contrast, since our method does not need segmentation, ours restores the boundaries much better than the layered model.

In Fig. 5.18, we quantitatively compare the optical flow accuracies with [15] on synthetic blurry images. As publicly available code of [15] cannot handle Gaussian blur, we synthesized blurry frames which have motion blurs only. Although [15] was proposed to handle blurry images in optical flow estimation, its assumption does not hold in motion boundaries, which is very important for deblurring. Therefore, their optical flow is inaccurate in the motion boundaries of moving objects. By contrast, our model can cope with abrupt motion changes, and thus performs better than the conventional model.

Moreover, we show the deblurring results with and without using the temporal coherence term in (5.5), and verify that our temporal coherence model clearly restores edges and significantly reduces ringing artifacts near the edges in Fig. 5.19.

Finally, other deblurring results from numerous real videos are shown in Fig. 5.20. Notice that, our model successfully restores the face which has highly non-uniform blurs because the person moves rotationally (e.g. Fig. 5.20 (e)).

5.7 Summary

In this study, we introduced a novel method that removes general blurs in dynamic scenes which conventional methods fail to. We inferred bidirectional optical flows

to parametrize motion blur kernels, and estimated the scales of Gaussian blurs to approximate defocus blur kernels. Therefore the proposed method could handle general blurs, by estimating a pixel-wise different blur kernel. In addition, we proposed a new single energy model that estimates optical flows, defocus blur maps and latent frames, jointly. We also provided a framework and efficient solvers to minimize the proposed energy function and it has been shown that our method yields superior deblurring results to several state-of-the-art deblurring methods through intensive experiments with real challenging blurred videos. Moreover, we provided the publicly available benchmark dataset to evaluate the non-uniform deblurring methods and we quantitatively evaluated the performance of the proposed method using the proposed dataset.

Figure 5.15: Left to right: Blurry frames of dynamic scenes, deblurring results of [12], and our results.

Figure 5.16: Left to right: Blurry frame, deblurring result of [12], and ours.

Figure 5.17: Comparison with segmentation-based approach. Left to right: Blurry frame, result of [11], and ours.

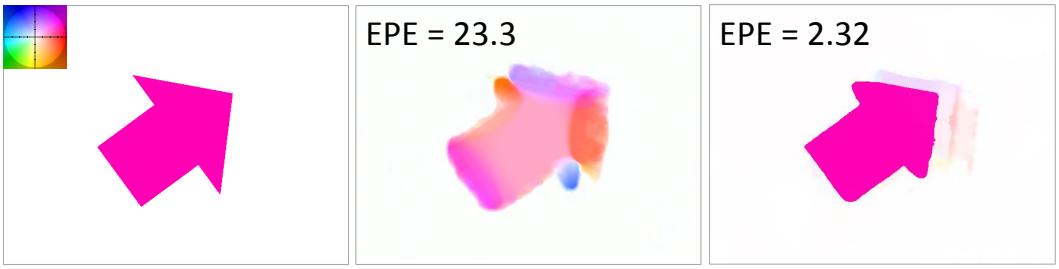


Figure 5.18: Left to right: Color coded ground truth optical flow between blurry images. Optical flow estimation result of [15]. Our result.

(a)

(b)

(c)

Figure 5.19: (a) A blurry frame of a video. (b) Our deblurring result without using $\mathbf{E}_{temporal}$. (c) Our deblurring result with $\mathbf{E}_{temporal}$.

Figure 5.20: Left to right: Numerous real blurry frames and our deblurring results.

Chapter 6

Conclusion

In this dissertation, deblurring problems of dynamic scenes which include spatially varying blurs caused by camera shake, object motions, and depth variations were addressed. In particular, to solve this problem, several approaches were proposed and removed spatially varying blurs in dynamic scenes unlike conventional methods that were built on a static scene assumption. The proposed deblurring methods were based on segmentation, exemplar, and kernel-parametrization.

• **Deblurring with segmentation**: As moving objects and background have different motion blurs in dynamic scenes, (motion) segmentation has been used to solve dynamic scene deblurring problem, and a joint approach was proposed in this dissertation to restore the sharp edges in the latent image. In particular, to handle locally varying blurs, the weighted sum of multiple blur data models was proposed, which jointly estimates the latent sharp image, different motion blurs, and their associated pixel-wise varying weights. As local weights are determined adaptively and get high values when the corresponding data models have high data fidelities, the

weight estimation process implies (soft) segmentation after all. Moreover, non-local regularization for weights were employed to render more reliable segmentation results and our intensive experiments demonstrated the performance of the proposed segmentation-based method.

• **Deblurring with exemplar**: Although our segmentation-based deblurring method renders good results, it requires large and texture-rich segments to estimate accurate blur kernels. However, motion blurs in real-world images, especially those captured with a long exposure time or containing fast moving objects, are often large. Thus this makes it difficult to estimate accurate blur kernels from a small and texture-less segment, and other state-of-the-art dynamic scene deblurring methods may perform poorly because of poor kernel approximation by inaccurate segmentation. Therefore, to facilitate more accurate kernel estimation for dynamic scenes, a new method utilizing a sharp exemplar image in addition to the blurred photograph was presented, and the limitations of the previous segmentation-based approach were overcome. The core of the exemplar-based method was a blur-aware flow estimation technique for establishing dense correspondences between the two images based on the notion that pixels with similar motion exhibit similar blur. Warping the sharp image by these flows produces an exemplar that is used to assist in the estimation of blur kernels, their associated image segments, and the latent image.

• **Deblurring with kernel-parametrization**: Previous segmentation-based and exemplar-based methods could handle segment-wise varying blurs by segment-wise rigid motions. However, they could not handle pixel-wise varying blurs in real dynamic scenes. Therefore, to handle pixel-wise varying blurs, locally varying blur

kernel was approximated as linear and was parameterized with motion flows. Thus, a new energy model that simultaneously estimates motion flow and the latent image was proposed for the first time. A highly effective convex optimization-based solution mitigating the computational difficulties of the proposed model was established, and comparative experimental results on challenging real blurry images demonstrated the efficiency of the proposed method.

Moreover, this image deblurring method with kernel-parametrization could be naturally extended for video deblurring, and thus a video deblurring method which could handle general blurs inherent in dynamic scenes was proposed using kernel-parametrization. Specifically, to handle locally varying blurs in videos of dynamic scenes, motion blur kernel was parameterized with bidirectional optical flows and defocus blur kernel was approximated with Gaussian kernel. Thus, the proposed energy model jointly estimated optical flows, defocus blur maps, and latent frames. By minimizing the proposed energy function, significant improvements in removing blurs and estimating accurate optical flows among blurry frames were achieved. Extensive experimental results demonstrated the superiority of the proposed video deblurring method in real and challenging videos that state-of-the-art methods failed in either deblurring or optical flow estimation.

Nevertheless, the proposed models have some limitations. Since they were implemented on Matlab, it is time consuming, and more, needs large computational resources. Thus, for further practical applications, reducing the running time by code optimization and parallel implementation as well as efficient memory management will be considered in future work. Moreover, the proposed deblurring methods

assume that blur kernel is partially (segment-wise) uniform or locally linear. Thus, proposed methods fail to deblur where blur kernel is locally difficult to approximate as linear or parameterize with motion flows (i.e. occluded regions). To address this problem, a learning-based dynamic scene deblurring method with a convolutional neural network (CNN) can be considered using our realistic dataset captured with the high-speed-camera, and the possibility of the learning-based approach will be investigated in the future.

Bibliography

[1] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *European Conference on Computer Vision*, 2010.

[2] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, 2012.

[3] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Transactions on Graphics*, 2008.

[4] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf, "Fast removal of non-uniform camera shake," in *International Conference on Computer Vision*, 2011.

[5] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *European Conference on Computer Vision*, 2010.

[6] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Computer Vision and Pattern Recognition*, 2013.

[7] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2012.

[8] T. H. Kim and K. M. Lee, "Segmentation-free dynamic scene deblurring," in *Computer Vision and Pattern Recognition*, 2014.

[9] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Computer Vision and Pattern Recognition*, 2015.

[10] T. H. Kim, B. Ahn, and K. M. Lee, "Dynamic scene deblurring," in *International Conference on Computer Vision*, 2013.

[11] J. Wulff and M. J. Black, "Modeling blurred video with layers," in *European Conference on Computer Vision*, 2014.

[12] S. Cho, J. Wang, and S. Lee, "Video deblurring for hand-held cameras using patch-based synthesis," *ACM Transactions on Graphics*, 2012.

[13] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via l0-regularized intensity and gradient prior," in *Computer Vision and Pattern Recognition*, 2014.

[14] J. Shi, L. Xu, and J. Jia, "Just noticeable defocus blur detection and estimation," in *Computer Vision and Pattern Recognition*, 2015.

[15] T. Portz, L. Zhang, and H. Jiang, "Optical flow in the presence of spatially-varying motion blur," in *Computer Vision and Pattern Recognition*, 2012.

[16] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics*, 2006.

[17] S. Cho and S. Lee, "Fast motion deblurring," *ACM Transactions on Graphics*, 2009.

[18] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Computer Vision and Pattern Recognition*, 2010.

[19] T. H. Kim and K. M. Lee, "Generalized video deblurring for dynamic scenes," in *Computer Vision and Pattern Recognition*, 2015.

[20] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Transactions on Graphics*, 2007.

[21] A. Levin, "Blind motion deblurring using image statistics," in *Advances in Neural Information Processing Systems*, 2006.

[22] S. Harmeling, H. Michael, and B. Schoelkopf, "Space-variant single-image blind deconvolution for removing camera shake," in *Advances in Neural Information Processing Systems*, 2010.

[23] H. Ji and K. Wang, "A two-stage approach to blind spatially-varying motion deblurring," in *Computer Vision and Pattern Recognition*, 2012.

[24] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Transactions on Graphics*, 2010.

[25] A. Levin and Y. Weiss, "User assisted separation of reflections from a single image using a sparsity prior," *IEEE Transactions Pattern Analysis Machine Intelligence*, 2007.

[26] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Advances in Neural Information Processing Systems*, 2009.

[27] M. Werlberger, T. Pock, and H. Bischof, "Motion estimation with non-local total variation regularization," in *Computer Vision and Pattern Recognition*, 2010.

[28] Y.-W. Tai and S. Lin, "Motion-aware noise filtering for deblurring of noisy and blurry images," in *Computer Vision and Pattern Recognition*, 2012.

[29] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, 2011.

[30] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*. Springer, 1996.

[31] C. Michelot, "A finite algorithm for finding the projection of a point onto the canonical simplex of rn," *Journal of Optimization Theory and Applications*, 1986.

[32] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Computer Vision and Pattern Recognition*, 2009.

[33] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *International Conference on Computer Vision*, 2007.

[34] F. Couzinie-Devy, J. Sun, K. Alahari, and J. Ponce, "Learning to estimate and remove non-uniform image blur," in *Computer Vision and Pattern Recognition*, 2013.

[35] Y. Hacohen, E. Shechtman, and D. Lischinski, "Deblurring by example using dense correspondence," in *International Conference on Computer Vision*, 2013.

[36] J. Chen, L. Yuan, C.-K. Tang, and L. Quan, "Robust dual motion deblurring," in *Computer Vision and Pattern Recognition*, 2008.

[37] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring using multiple images," *Journal of computational physics*, 2009.

[38] Y. Li, S. B. Kang, N. Joshi, S. M. Seitz, and D. P. Huttenlocher, "Generating sharp panoramas from motion-blurred videos," in *Computer Vision and Pattern Recognition*, 2010.

[39] J. Hu, O. Gallo, K. Pulli, and X. Sun, "Hdr deghosting: How to deal with saturation ?" in *Computer Vision and Pattern Recognition*, 2013.

[40] 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*, 2013.

[41] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," *Pattern Recognition*, 2007.

[42] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[43] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[44] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, 2010.

[45] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof, "Pushing the limits of stereo using variational stereo estimation," in *Intelligent Vehicles Symposium*, 2012.

[46] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision*, 2012.

[47] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM Transactions on Graphics*, 2011.

[48] S. Dai and Y. Wu, "Motion from blur," in *Computer Vision and Pattern Recognition*, 2008.

[49] A. Levin, D. Lischinski, and Y. Weiss, "A closed form solution to natural image matting," *IEEE Transactions Pattern Analysis Machine Intelligence*, 2008.

[50] A. Chakrabarti, T. Zickler, and W. T. Freeman, "Analyzing spatially-varying blur," in *Computer Vision and Pattern Recognition*, 2010.

[51] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradientshop: A gradient-domain optimization framework for image and video filtering," *ACM Transactions on Graphics*, 2010.

[52] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Image enhancement and denoising by complex diffusion processes," *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2004.

[53] N. Joshi, R. Szeliski, and D. Kriegman, "Psf estimation using sharp edge prediction," in *Computer Vision and Pattern Recognition*, 2008.

[54] K. Schelten and S. Roth, "Localized image blur removal through non-parametric kernel estimation," in *International Conference on Pattern Recognition*, 2014.

[55] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar, "Estimating spatially varying defocus blur from a single image," *IEEE Transactions on Image Processing*, 2013.

[56] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2006.

[57] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, 1992.

[58] T. Pock, L. Zebedin, and H. Bischof, *TGV-Fusion.* Springer, 2011.

[59] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *International Conference on Computer Vision*, 2013.

[60] J. Braux-Zin, R. Dupont, and A. Bartoli, "A general dense image matching framework combining direct and feature-based costs," in *International Conference on Computer Vision*, 2013.

[61] C. Vogel, S. Roth, and K. Schindler, "An evaluation of data costs for optical flow," *Pattern recognition*, 2013.

[62] R. Ranftl, K. Bredies, and T. Pock, "Non-local total generalized variation for optical flow estimation," in *European Conference on Computer Vision*, 2014.

[63] T. H. Kim, H. S. Lee, and K. M. Lee, "Optical flow via locally adaptive fusion of complementary data costs," in *International Conference on Computer Vision*, 2013.

[64] T. Pock and A. Chambolle, "Diagonal preconditioning for first order primal-dual algorithms in convex optimization," in *International Conference on Computer Vision*, 2011.

[65] F. Steinbrücker, T. Pock, and D. Cremers, "Advanced data terms for variational optic flow estimation," in *Proceedings of International Workshop on Vision, Modeling, and Visualization*, 2009.

[66] S. Osher and L. I. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM Journal on Numerical Analysis*, 1990.

[67] Y.-W. Tai, P. Tan, and M. S. Brown, "Richardson-lucy deblurring for scenes under a projective motion path," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[68] S. Cho, H. Cho, Y.-W. Tai, and S. Lee, "Registration based non-uniform motion deblurring," in *Computer Graphics Forum*, 2012.

[69] C. Paramanand and A. N. Rajagopalan, "Non-uniform motion deblurring for bilayer scenes," in *Computer Vision and Pattern Recognition*, 2013.

[70] H. S. Lee and K. M. Lee, "Dense 3d reconstruction from severely blurred images using a single moving camera," in *Computer Vision and Pattern Recognition*, 2013.

[71] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro, "A variational framework for simultaneous motion estimation and restoration of motion-blurred video," in *Computer Vision and Pattern Recognition*, 2007.

[72] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[73] S. Bae and F. Durand, "Defocus magnification," in *Computer Graphics Forum*, 2007.

[74] E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," in *International Conference on Computational Photography*, 2011.

[75] S. Zhuo and T. Sim, "Defocus map estimation from a single image," *Pattern Recognition*, 2011.

[76] 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 *European Conference on Computer Vision*, 2012.

[77] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Advances in Neural Information Processing Systems*, 2014.

[78] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[79] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Computer Vision and Pattern Recognition*, 2009.

[80] Z. Hu, L. Xu, and M.-H. Yang, "Joint depth estimation and camera shake removal from single blurry image," in *Computer Vision and Pattern Recognition*, 2014.

[81] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for tv-l 1 optical flow," *Statistical and Geometrical Approaches to Visual Motion Analysis*, 2009.

[82] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Computer Vision and Pattern Recognition*, 2011.

[83] A. Agrawal and R. Raskar, "Optimal single image capture for motion deblurring," in *Computer Vision and Pattern Recognition*, 2009.

국문초록

카메라 셔터가 열려있는 동안 카메라 혹은 물체의 움직임은 블러가 있는 흐릿한 영상을 만든다. 따라서 영상의 질을 확보하고 여러 가지 컴퓨터 비전 응용 문제의 전처리로 활용하기 위해, 영상 내 블러를 제거하기 위한 많은 연구가 진행되었고, 특히 블러가 균일한 정적 환경에서의 디블러링 기법들이 다수 제안되었다. 그러나 카메라 모션, 움직이는 물체, 뎁스 변화 등 다양한 요인이 복합되어 있는 실제 동적 환경에서는 블러가 균일하지 않기 때문에 기존의 디블러링 기법으로는 동적 환경에서의 블러를 제거하기 어렵다.

본 학위 논문에서는 동적 환경에서 발생하는 균일하지 않은 블러를 제거하는 여러 가지 디블러링 알고리즘을 제시한다. 특히 동적 환경에서의 블러를 제거하기 위해서는, 국지적으로 모양이 다른 블러 커널을 정확하게 추정하는 것이 가장 중요한 부분이고, 이를 위해 영상 분할(segmentation), 표본 영상(exemplar), 그리고 커널 매개화(kernel-parametrization)를 활용하는 세가지 다른 기법들을 제안한다. 제안하는 기법들은 비교적 큰 세그먼트 단위로 변화하는 커널을 다루는 기법부터 픽셀 단위로 변화하는 커널을 다루는 기법 순으로 소개 된다.

먼저, 영상 분할에 기반한 기법은, 영상 분할과 디블러링을 순차적으로 수행하는 기존의 기법들이 야기하는 모션 경계에서의 결함들을 제거하기 위해 영상 분할과 디블러링 문제를 함께 해결하는 조인트 에너지 모델을 최초로 제안하고 이를 효율적으로 최적화 하는 방법을 연구한다. 다음으로, 선명한 표본 영상을 활용하는 기법은 비디오 혹은 브라켓 영상에서 쉽게 획득할 수 있는 선명한 영상을 이용하여 블러가 있는 영상

을 디블러링 하는 기법으로, 영상 분할에 기반한 기법이 분할하는 세그먼트의 크기가 작거나 텍스처가 약한 영역에서 커널을 정확하게 추정할 수 없는 문제를 해결한다. 특히, 선명한 영상과 흐릿한 영상 사이의 모션 정보를 디블러링에 활용함으로써, 더욱 우수한 디블러링 결과물을 얻을 수 있는 기법이다. 마지막으로, 커널 매개화에 기반한 디블러링 기법은 각 픽셀마다 서로 다른 모션을 추정하고, 이를 이용하여 커널을 매개화 함으로써 궁극적으로는 동적 환경에서 픽셀마다 변화하는 블러를 다룰 수 있도록 한다. 특히, 블러 커널을 선형 근사 함으로써 복잡한 커널 추정 문제를 단순화 하고, 제안하는 에너지 모델을 최적화 함으로써 흐릿한 단일 영상으로부터 선명한 영상과 모션 정보를 동시에 얻을 수 있다. 그리고, 단일 영상의 디블러링에 적용하는 커널 매개화 기법을 동영상 디블러링에 확장 적용하고, 더 나아가 동영상 내 존재하는 시/공간상의 정보를 함께 활용하면 매우 우수한 디블러링 결과물을 획득할 수 있을 뿐만 아니라, 블러가 있는 영상들 사이에서도 정확한 정합 관계를 추정할 수 있다.

제안한 알고리즘의 성능을 면밀히 평가하기 위하여 다양한 실험이 시행된다. 특히 본 학위 논문에서는 고속 카메라를 이용하여 새롭게 제작한 블러 데이터 셋을 이용하여 제안하는 알고리즘들 간, 그리고 최신 기법과의 객관적인 성능 비교 결과가 제공되고, 동적 환경에서 촬영한 실제의 영상들을 이용하여 다양한 기법들과 주관적인 화질 비교가 가능하도록 하였다.

주요어: 동적환경 디블러링, 비균일 디블러링, 영상 분할, 표본 영상, 커널 매개화

학번: 2011-30224