Ph.D. DISSERTATION

Solving Multi-view Stereo and Image Restoration using a Unified Framework

통합 시스템을 이용한 다시점 스테레오 매칭과 영상 복원

BY

Haesol Park

February 2017

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

Ph.D. DISSERTATION

Solving Multi-view Stereo and Image Restoration using a Unified Framework

통합 시스템을 이용한 다시점 스테레오 매칭과 영상 복원

BY

Haesol Park

February 2017

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

Solving Multi-view Stereo and Image Restoration using a Unified Framework

통합 시스템을 이용한 다시점 스테레오 매칭과 영상 복원

지도교수 이 경 무
이 논문을 공학박사 학위논문으로 제출함

2017년 2월

서울대학교 대학원

전기 컴퓨터 공학부

박 해 솔

박해솔의 공학박사 학위 논문을 인준함

2017년 2월

위 원 장: _____
부위원장: _____
위　　원: _____
위　　원: _____
위　　원: _____

Abstract

Estimating camera pose and scene structures from seriously degraded images is challenging problem. Most existing multi-view stereo algorithms assume high-quality input images and therefore have unreliable results for blurred, noisy, or low-resolution images. Experimental results show that the approach of using off-the-shelf image reconstruction algorithms as independent preprocessing is generally ineffective or even sometimes counterproductive. This is because naive frame-wise image reconstruction methods fundamentally ignore the consistency between images, although they seem to produce visually plausible results.

In this thesis, from the fact that image reconstruction and multi-view stereo problems are interrelated, we present a unified framework to solve these problems jointly. The validity of this approach is empirically verified for four different problems, dense depth map reconstruction, camera pose estimation, super-resolution, and deblurring from images obtained by a single moving camera. By reflecting the physical imaging process, we cast our objective into a cost minimization problem, and solve the solution using alternate optimization techniques. Experiments show that the proposed method can restore high-quality depth maps from seriously degraded images for both synthetic and real video, as opposed to the failure of simple multi-view stereo methods. Our algorithm also produces superior super-resolution and deblurring results compared to simple preprocessing with conventional super-resolution and deblurring techniques.

Moreover, we show that the proposed framework can be generalized to handle more common scenarios. First, it can solve image reconstruction and multi-view stereo problems for multi-view single-shot images captured by a light field camera. By us-

ing information of calibrated multi-view images, it recovers the motions of individual objects in the input image as well as the unknown camera motion during the shutter time.

The contribution of this thesis is proposing a new perspective on the solution of the existing computer vision problems from an integrated viewpoint. We show that by solving interrelated problems jointly, we can obtain physically more plausible solution and better performance, especially when input images are challenging. The proposed optimization algorithm also makes our algorithm more practical in terms of computational complexity.

keywords: Computer Vision, Multi-view stereo, Image Deblurring, Image Super resolution, SLAM, Joint estimation

student number: 2011-30234

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Multi-view stereo is one of the most fundamental problems in computer vision. The goal of this problem is to find the scene structure and camera configuration from multiple images. The 3D information of the target scene and camera system is obtained by solving pixel-wise image correspondence problem with multi-view geometry constraints.

For the past decades, the performance of the multi-view stereo algorithms has increased to produce satisfactory results on most of the public datasets. Still, the experimental results reported so far in the literature are limited to the case where the input images are captured in a good condition. This is one of the main reasons preventing the multi-view stereo algorithms from being applied to real-world applications, where the quality of the input images is unreliable; the resolution could be low and images are often contaminated with severe motion blur. The accuracy of pixel-wise matching is directly influenced by the resolution of the input images. More importantly, the presence of blur breaks the most basic assumption in image matching, brightness consistency, if they appear differently across the images. To remedy this issue, one can

think of applying an image restoration method prior to feeding images to multi-view stereo methods.

Image restoration is another fundamental problem in computer vision. The goal of image restoration is to reconstruct the image with a better visual quality from the one or more degraded input images. According to the cause of degradation, the problem can be further categorized into, for example, image deblurring or image super resolution. Most conventional methods have focused on increasing visual quality of the output or the peak-signal-to-noise-ratio (PSNR) measure compared to the ground-truth image. However, this is not necessarily related to increasing consistency between multiple images when the images are reconstructed independently. In fact, applying single-view image restoration methods prior to multi-view stereo matching often produces worse results than the original degraded images.

In this dissertation, we suggest to solve image restoration and multi-view stereo matching jointly in a unified framework. We first show that these problems are inter-related closely by physical constraints, and show that joint formulation improves the performance significantly for both problems. We propose three different methods to handle three different scenarios, and experiments for each method indicate that they actually outperform the conventional methods.

1.1 Outline of Dissertation

The background and related works are summarized in Chapter 2. The basic equations and notations for image model and multi-view geometry are reviewed in Chapter 3. The generalized image capturing process modeled in this chapter are required to derive the important constraints used in the following chapters. In Chapter 4, the procedures

for making synthetic datasets to evaluate the proposed methods are introduced. The synthetic datasets in the following chapters are all made by one of these procedures. Three following chapters are presented to describe the proposed methods that solve the joint estimation of image restoration and multi-view stereo matching for three different scenarios. A method to handle one-shot multi-view images is discussed in Chapter 5. Chapter 6 deals with a batch processing method that can handle a single-view image sequence, and modification of it to process the similar input in a on-line manner, like SLAM, is described in Chapter 7. We conclude this dissertation with summary and future works in Chapter 8.

Chapter 2

Background

The coverage of multi-view stereo problem is wide and it has many subproblems. It is called stereo matching problem when the target variables are dense depth maps of input images [28, 63, 50]. Structure from motion (SfM) [62, 54, 3] and simultaneous localization and mapping (SLAM) [36, 40, 38, 56, 47, 17, 46, 16] are also subproblems of multi-view stereo, and their goal is to estimate the camera poses of input images with a global 3D map. The basic assumption of depth estimation problem is that the camera poses of the input images are all known, and thus the accuracy of the camera pose estimation have large influence on the performance of stereo matching since it gives the hard epipolar constraint [26]. In this sense, the accuracy of camera pose estimation is the key to the success of entire multi-view stereo pipeline.

Reconstructing camera poses from a sequence of images has long been a main research topic of computer vision. After a seminal paper by C. Tomasi and T. Kanade [62] was published, the researchers have tried to process the whole input images in a batch manner following to the standard SfM procedure in the early stage [54, 3]. These SfM methods first detect feature points for all the images and match them across the im-

ages. Once the reliable matches are obtained after the outlier removal technique, such as **RANSAC** [21], it solves global optimization problem in which the coordinates of sparse 3D points and camera poses of each image are estimated jointly. This global optimization process is called bundle adjustment and it is time-consuming process due to its computational complexity.

As the need of real-time or online application of camera pose estimation methods increases, the methods based on SLAM approach are more actively researched recently. The primitive SLAM algorithms are based on a state space method to model the scene observation and camera poses, and use filtering scheme to update the states [40, 38, 56]. While the use of filtering scheme in an online manner makes these SLAM algorithms faster than the original SfM algorithms, the reliability or accuracy of the reconstructed results tends to be worse due to the accumulation of error. A seminal work called **PTAM** [36], proposed by G. Klein and D. Murray, solves this issue by designing a hybrid system, key-frame based SLAM. This hybrid system combines the strengths of both SfM and traditional SLAM by continuously processing the every new incoming input image while performing batch bundle adjustment on sparsely sampled key frames.

The recent SLAM methods after **PTAM** [36] can be categorized into two approaches according to the way of establishing the relationship between the observation and the target variables. Given that the original observation is a set of images and the target variables are 3D map and poses, indirect methods [36, 46] first process the input images to get intermediate representations that will be used to estimate the target variables. Usually, the intermediate representation is sparse image feature points detected for each frame and matched across frames as in SfM. Since the **PTAM** [36] itself belongs to this category, it has long been a standard formulation. Feature-point based

SLAM algorithms is computationally efficient and can have illumination/exposure variance if the feature detection and matching is robust to those variations. However, when the scene has low texture areas and there is motion blur, both of the feature detection and matching become unreliable, which in turn reduces the SLAM system performance.

On the other hand, direct methods [47, 17, 16] compares the pixel values between images with different viewpoints, by modeling the image warping operation modeled as the function of 3D scene structure and camera poses. By assuming a photometric consistency, they can find the best scene structure and camera poses by optimizing the pixel color difference of the reference image and the target neighboring images. These approaches show robustness against the presence of low texture areas or motion blur to some degree. The earlier direct methods, however, are computationally expensive since they estimate dense [47] or semi-dense [17] depth maps for each key frame, and are weak to illumination/exposure variations. J. Engel *et al*. [16] proposed a key-point based direct method, where the warping equation and texture comparison is computed only for sparsely sampled key points of each key frame. Along with the use of photometric calibration between frames inside the SLAM system, this enables the direct SLAM methods to overcome its inherent weaknesses compared to the feature-point based SLAM methods.

Although there has been significant progress in solving camera pose estimation problem, none of these methods explicitly handle degraded input images. The resolution of input image is directly related to the accuracy of depth and camera pose estimation since the details of image texture and the basic unit of pixel-wise matching is dependent on the image resolution. Furthermore, and presence of severe motion blur often results in critical failures in feature-point based SLAM because it reduces

the number of detected feature and feature matching becomes unreliable.

A few works has been published to solve multi-view stereo with degraded input. H. S. Lee *et al*. proposed a SLAM framework [39] where the camera poses are estimated using blur-aware matching [32]. The idea is to model the blur kernel around each feature point using the camera motion and the corresponding 3D point coordinate. More specifically, it first compute the optical flow of a feature point to the previous frame, and use that information to approximate a linear blur kernel with the known frame rate and shutter speed. Use of blur-aware matching helps to establish reliable feature matching in the presence of severe motion blur. Furthermore, when the size of blur kernels are large, *i.e*., if the blur is so severe, then feature detection is done after deblurring the image to increase the repeatability of feature detection. Still, the use deblurring is not explicitly modeled in terms of SLAM pipeline. In terms of SLAM equations, this method implement blur-robust SLAM framework rather than joint estimation of camera poses and latent images.

H. S. Lee and K. M. Lee proposed joint estimation frameworks to handle dense depth estimation with deblurring [41], and dense depth estimation with super resolution [42]. The philosophy behind these methods are same as the one in [39]. Given camera poses as input along with degraded image, they model degradation process using depth and latent images. Unlike the method in [39], they explicitly model the main energy function using the latent image as one of target variables, and jointly estimate depth maps and images. Still, the main assumption behind these methods is that the accurate camera poses are given. Thus, the entire process should be sequential and independent use of separated methods if a user wants to apply these methods to process a degraded image sequence.

Chapter 3

Generalized Imaging Model

The main goal of this thesis is to verify that jointly solving two fundamental computer vision problems, image restoration and stereo matching, is theoretically plausible and also effective in practice. To that end, we revisit the formulation of imaging process in terms of single camera geometry, and then generalize the formulation to make it cover more blurred and low-resolution images. Because all the following practical methods in the following chapters are based on this physical interpretation of generalized imaging process, the notations and equations introduced in this chapter will be used frequently throughout this paper. Also, note that the target scene is assumed to be static, *i.e.*, only the camera is moving, in the following discussions.

3.1 Camera Projection Model

We review the imaging process of a simple pin-hole camera model. Suppose we have a 3D point $\mathbf{X}_w \in \mathbb{R}^3$, and it is projected to a pixel coordinate $\mathbf{x}_c \in \mathbb{R}^2$ of the reference camera c. The subscript w and c are used to mark the coordinate systems, w for the world-coordinate and c for the camera-coordinate, respectively.

The 3D point $\mathbf{X}_w$ is first transformed to a 3D camera coordinate $\mathbf{X}_c$, where the camera center is the origin and the axes are alligned with camera orientation as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{X}_c \\ 1 \end{bmatrix} &= \mathbf{P}_c \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_c & \mathbf{T}_c \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix}. \end{aligned} \tag{3.1}$$

where $\mathbf{P}_c$ denotes a 4×4 transformation matrix specific to the camera c and it consists of the rotation matrix $\mathbf{R}_c$ and translation vector $\mathbf{T}_c$.

This point $\mathbf{X}_c$ is now projected into camera using the camera-specific intrinsic parameters represented by 3×3 matrix $\mathbf{K}_c$ as follows:

$$\mathbf{x}_c = h\left(\mathbf{K}_c \mathbf{X}_c\right). \tag{3.2}$$

The function $h\left(\cdot\right)$ is used to make the inhomogenous 2D coordinate into homogeneous coordinate and works as follows:

$$h\left(\mathbf{X}\right) = \begin{bmatrix} \mathbf{X}\left(1\right)/\mathbf{X}\left(3\right) \\ \mathbf{X}\left(2\right)/\mathbf{X}\left(3\right) \end{bmatrix}. \tag{3.3}$$

Thorughout this paper, we only focus on the focal length and principal point in the intrinsic matrix $\mathbf{K}_c$. Other more complex intrinsic parameters or radial distortion phenomenon is assumed to be pre-processed to fit the simpler projection model mentioned above.

A pixel color value of image I at a pixel position $\mathbf{x}$ is denoted by $I\left(\mathbf{x}\right)$. This definition of pixel value should only be valid for integer indices within the range of image size since we deal with digitalized images. However, we slightly abuse the notation to expand the expression to cover non-integer coordinate values; for those cases, the

pixel values are interpolated from its surrounding valid neighbors using bilinear interpolation. The pixel value could be a scalar (grayscale image) or a three-dimensional vector (RGB color image).

3.2 Depth and Warping Operation

We define backward image warping operation from a target image domain t to a reference image domain r, based on the projection model in the previous section. First, we reproject the pixel coordinate x_r into the 3D world coordinate using the inverse depth map D_r, along with camera parameters $\mathbf{P}_r$ and $\mathbf{K}_r$ as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{X}_w \\ 1 \end{bmatrix} &= {\mathbf{P}_r}^{-1} \begin{bmatrix} \mathbf{X}_r \\ 1 \end{bmatrix}, \\ \mathbf{X}_r &= {\mathbf{K}_r}^{-1} \left(\frac{1}{D_r\left(\mathbf{x}_r\right)} \begin{bmatrix} \mathbf{x}_r \\ 1 \end{bmatrix} \right). \end{aligned} \tag{3.4}$$

Once the 3D world coordinate is recovered, we project $\mathbf{X}_w$ into a target image domain to obtain $\mathbf{x}_t$ using the equation (3.1) and (3.2). By using $\mathbf{R}$ and $\mathbf{T}$ instead of $\mathbf{P}$, we represent this warpiing operation as follows:

$$\begin{aligned} &W^{r\to t}\left(\mathbf{x}_r | D_r, \mathbf{P}_r, \mathbf{K}_r, \mathbf{P}_t, \mathbf{K}_t\right) = \\ &h\left(\mathbf{K}_t \left(\mathbf{R}_{rt} {\mathbf{K}_r}^{-1} \left(\begin{bmatrix} \mathbf{x}_r \\ 1 \end{bmatrix} \right) + D_r\left(\mathbf{x}_r\right) \mathbf{T}_{rt} \right) \right). \end{aligned} \tag{3.5}$$

The notations $\mathbf{R}_{rt}$ and $\mathbf{T}_{rt}$ are originated from $\mathbf{P}_{rt} = \mathbf{P}_t \mathbf{P}_r^{-1}$.

This warping operation will be frequently used, and we often omit the conditioning variables in the remaining parts of this paper, making it $W^{r\to t}\left(\mathbf{x}_r\right)$, for simplicity. Note that the operation is still function of all $\mathbf{x}_r, D_r, \mathbf{P}_r, \mathbf{K}_r, \mathbf{P}_t$, and $\mathbf{K}_t$.

3.3 Representation of Camera Pose in $\mathbb{SE}(3)$

The pose matrix $\mathbf{P}_c$ of a camera c has 16 elements, but its degree of freedom (DoF) is actually six; three for orientation and three for translation. The manifold it resides is $\mathbb{SE}(3)$, which is Lie group [6]. It is convenient to use corresponding Lie algebra of $\mathbb{SE}(3)$due to ease of arithmetic operations regarding the poses. We represent this Lie algebra as $\mathfrak{se}(3)$. The elements of $\mathfrak{se}(3)$are six-dimensional vectors and sequential transformations using the multiple poses become simple additions or subtractions of vectors in the $\mathfrak{se}(3)$space.

The mapping between two space is defined and denoted by exponential and logarithm mapping. For example, if $\varepsilon_c \in \mathfrak{se}(3)$is the corresponding vector to $\mathbf{P}_c \in \mathbb{SE}(3)$, then the following equations hold:

$$\begin{aligned} \varepsilon_c &= \log(\mathbf{P}_c) \ , \\ \mathbf{P}_c &= \exp(\varepsilon_c) \, . \end{aligned} \tag{3.6}$$

Please refer to [6] for more details.

3.4 Proposed Imaging Model

We define a generalized imaging model that can describe the downsampling and blur process occurring during the capture of an image in real-world scenario. The most basic assumption is that the acquired image B_{c_t} from a camera c at time t is the downsampled version of integration of imaginary high-resolution sharp images I_{c_τ} captured by the image sensor during the shutter time. Ignoring the effect of camera response function, this results in the following equation:

$$B_{c_t} = \frac{1}{t_c - t_o} \left(\int_{t_o}^{t_c} I_{c_\tau} d\tau \right) \downarrow_s . \tag{3.7}$$

The value of t_o and t_c represents the times when shutters are opened and closed, respectively ($t_o \leq t \leq t_c$). The down arrow symbol $\downarrow_s$ represents the downsampling process with the sampling rate s. Note that the image is blurred when the camera motion is large enough to make visual difference between the imaginary sharp images at time t_o and t_c.

Reconstruction of the latent image from the acquired image B_{c_t} means estimating one of the imaginary sharp high-resolution images, I_{c_t}, where t can be arbitrarily chosen from the range between t_o and t_c. Solving image restoration problem is not straightforward since the capturing process in Equation (3.7) contains too many other unknown variables I_{c_τ} other than B_{c_t} and I_{c_t}. The key observation to relieve this under-determined problem is that the other imaginary images I_{c_τ} can be modeled as the warped version of target latent image I_{c_t}, using the motion of camera and the scene structure.

We represent the pose of the camera and the depth map of the image I_{c_τ} as $\mathbf{P}_{c_\tau}$ and D_{c_τ}, respectively, and the time-invariant camera intrinsic matrix as $\mathbf{K}_c$. Using these notations, the images I_{c_τ} during the shutter time can be approximated as follows:

$$I_{c_\tau}(\mathbf{x}) \approx I_{c_t}\left(W^{c_\tau \to c_t}(\mathbf{x}|D_{c_\tau}, \mathbf{P}_{c_\tau}, \mathbf{K}_{c_\tau}, \mathbf{P}_{c_t}, \mathbf{K}_{c_t})\right). \tag{3.8}$$

The warping function $W^{c_\tau \to c_t}(\cdot)$ is defined in Equation (3.5).

In the following, the integral in Equation (3.7) is approximated by using a finite sum of images. With the insertion of Equation (3.8), it results in the following:

$$B_{c_t} \approx \Psi_{c_t} \circ I_{c_t}, \tag{3.9}$$

$$(\Psi_{c_t} \circ I)(\mathbf{x}) = \left(\frac{1}{M}\sum_{m=1}^{M} I\left(W^{c_{\tau_m} \to c_t}(\mathbf{x})\right)\right)\downarrow_s, \tag{3.10}$$

where $\tau_m = (m/M)(t_c - t_o) + t_o$. The parameter M controls the degree of discretization. Note that we define $\Psi_{c_t}(\cdot)$ as the operator on a general image I, to approximate

the degradation due to the capturing process of camera c at time t as a general operator.

In practice, the values of ${D_{c_{\tau_m}}}'s$ and ${\mathbf{P}_{c_{\tau_m}}}'s$ are approximated by using D_{c_t}, $\mathbf{P}_{c_t}$, and $\mathbf{P}_{c_{t_o}}$. First, $\mathbf{P}_{c_{t_c}}$ is extrapolated by using $\mathbf{P}_{c_t}$ and $\mathbf{P}_{c_{t_o}}$. Then, $\mathbf{P}_{c_{\tau_m}}$'s are sampled from the interpolated camera path between $\mathbf{P}_{c_{t_o}}$ and $\mathbf{P}_{c_{t_c}}$. The interpolation and extrapolation is conducted in the $\mathfrak{se}(3)$ space. Given $\Delta\varepsilon_{c_{t_o}c_{t_c}} = \log\left(\mathbf{P}_{c_{t_c}}\left(\mathbf{P}_{c_{t_o}}\right)^{-1}\right)$, the interpolation is made as follows:

$$\mathbf{P}_{C_{\tau_m}} = \exp\left(\frac{m}{M}\Delta\varepsilon_{c_{t_o}c_{t_c}}\right)\mathbf{P}_{c_{t_o}}. \tag{3.11}$$

Once we have the camera pose at time τ_m, then the depth map $D_{c_{\tau_m}}$ can be approximated by warping the depth map D_{c_t}. The warped depth value can be computed by reprojecting D_{c_t} to the world coordinate and projecting it to the virtual camera at $\mathbf{P}_{c_{\tau_m}}$. The value of $D_{c_{\tau_m}}$ at the projected pixel position is the actual depth value of the 3D point in terms of $\mathbf{P}_{c_{\tau_m}}$.

The capturing operator $\Psi_{c_t}(\cdot)$ is now only dependent on D_{c_t}, $\mathbf{P}_{c_t}$, and $\mathbf{P}_{c_{t_o}}$. The concept of this blur operation is briefly illustrated in Figure 3.1 in comparison to the conventional blur model that is based on the simple optical flow estimations as in [41, 49]. The benefit of using the proposed blur model is also verified by visualizing the deblurring results in Figure 3.2.

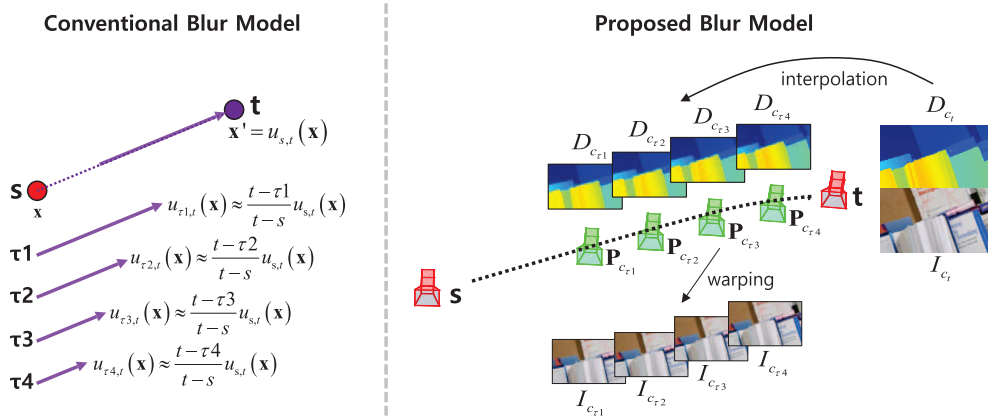
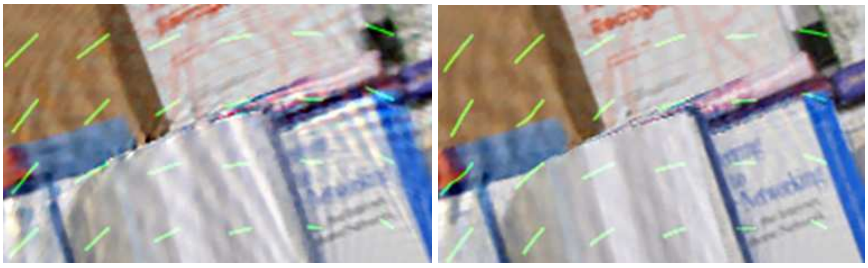


Figure 3.1: Comparison of the conventional blur model used in [41, 49] and the proposed one. Both models illustrate the blur procedures for frame at time t, where s is the time of the previous frame. The proposed model approximate the intermediate images I_{c_τ}'s during the shutter time using the interpolated camera poses $\mathbf{P}_{c_\tau}$'s and depth maps D_{c_τ}'s, while the conventional model relies on the single optical flow map from s to t, $u_{s,t}$.



(a) A deblurring result of the conventional blur model [41, 49]

(b) A deblurring result of the proposed blur model

Figure 3.2: Comparison of deblurring results of the conventional blur model used in [41, 49] and the proposed one. The deblurred images of each model are visualized with overlaid blur kernels. Although both are obtained by using the ground-truth depth map and camera poses, the image obtained by conventional blur model exhibits more artifacts due to inaccurately approximated blur kernels.

Chapter 4

Rendering Synthetic Datasets

The effectiveness of joint estimation of image restoration and multi-view stereo problem should be verified in various scenarios. Although the improvement over the conventional methods that solve each problem separately can be shown qualitatively by visual inspection for real-world data, it is not straightforward to compare the results quantitatively. This is because it is very difficult to get the ground-truth data for depth maps, camera poses, and latent images for a blurred image sequence.

The synthetic datasets introduced in this chapter are made to enable quantitative comparisons of the proposed algorithms. In the following subsections, we suggest two different ways to generate the synthetic datasets with a few visual samples of generated datasets. Then, we verify the effect of the generalized imaging process in Section 3.4 using the datasets, in terms of three separate problems; camera pose estimation, depth map estimation, and image restoration. Since the ground-truth data is fully known, we can perform modular tests where the variables other than the target variable are fully known.

4.1 Making Blurred Image Sequences using Depth-based Image Rendering

If we have clean images with the ground-truth depth maps, we can synthesize the images with simulated motion blur using the Equation (3.10). For example, the Middlebury stereo datasets [28] provide stereo image pairs with corresponding ground-truth depth maps and relative camera pose (known by the baseline). First, we set an imaginary camera motion path around those two stereo cameras, but the path should not necessarily be linear in $\mathfrak{se}(3)$domain . Given the camera motion path, the intermediate camera poses at the desired time stamps, which is again arbitrarily set, with the corresponding depth maps are interpolated by using the same approach as in Section 3.4. The image at a specific time can be generated by warping the both stereo images into the target image frame and then blending them to minimize the artifacts or holes due to occlusions. By properly adjusting the shutter time and degree of discretization M in the Equation (3.10), we can simulate a blurred image sequence with ground-truth camera poses, depth maps, and latent images.

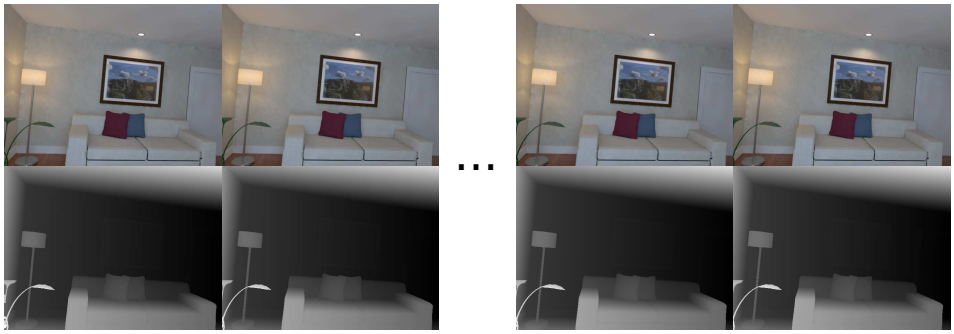
The same procedure can be applied to any datasets with ground-truth depth maps and camera poses. Figure 4.1 shows the example intermediate rendering results and the resultant image with simulated blur for a ICL-NUIM dataset [24].

4.2 Making Blurred Image Sequences using Blender

The Blender software [1] is widely used to make synthetic datasets for various computer vision tasks [9, 20, 48]. Basically, it is a rendering software that produces realistic rendering images given the 3D scene model and camera model. It is possible to generate an image sequence by defining the dynamics of the 3D scene, either for the object

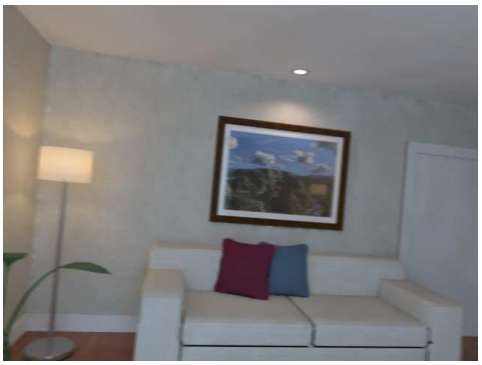
motions or for the camera motion. The ground-truth depth maps and camera poses for each frame in an image sequence can be easily obtained by using python scripting.

Using the Blender software, we generate blurred image sequences with desired ground-truth data. While the Blender software provides default options for blur simulation given the camera motion, it is found that the resultant blur images are not quite accurate. The output images seem blurred with a glance, but the blurring pattern is not consistent with the actual camera motion given the scene geometry. It is suspected that the internal operation to make fast blur simulation is based on some unrealistic approximations, but the exact operation is not known to the users. Thus, we manually render the images during the shutter time with the predefined discretization M and accumulate the images to approximate the Equation (3.7). Note that this is a better approximation than the case of Middlebury dataset in Section 4.1 since the actual intermediate images are used instead of the synthetically interpolated images.



(a) Intermediate images and depth maps during the shutter time

(b) An original clean image



(c) An image with simulated blur

Figure 4.1: Synthesis of a blurry image using depth-based image rendering is visualized. The image shown in this figure is from a single frame of **lr kt0** sequence of ICL-NUIM dataset [24].

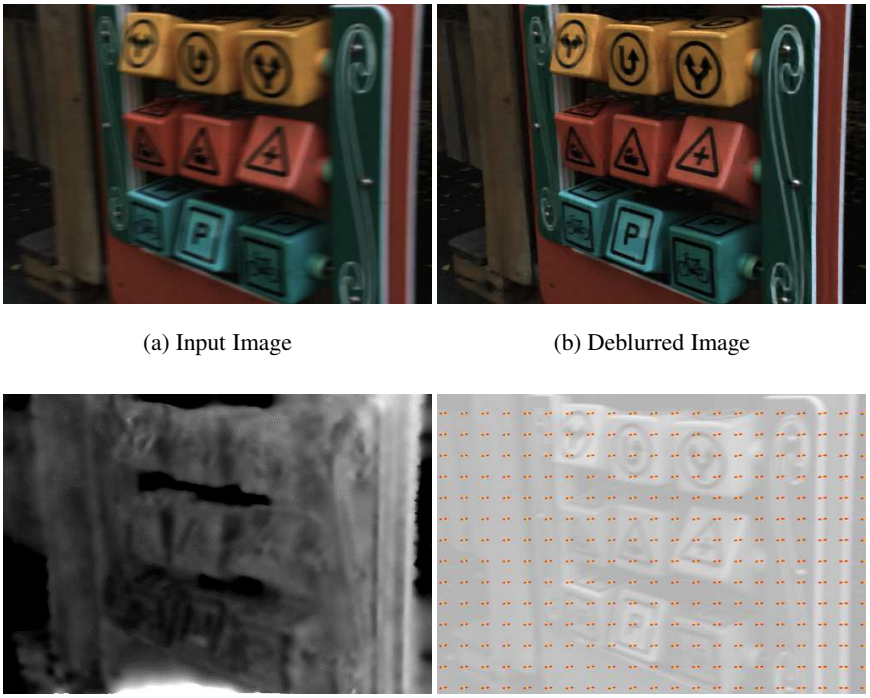
Chapter 5

A Unified Framework for Single-shot Multi-view Images

5.1 Introduction

A light field camera is an effective device that can embed the 3D structure of a scene in a single image. The 4D light field can record the direction of the incoming light through the lens, and this has the same effect as shooting a multi-view image of a narrow baseline using multiple single-lens cameras. Several years have passed since the light field camera was commercialized, and studies using light field images have been actively conducted in computer vision. Most studies have been related to depth estimation [11, 31, 43, 59, 60, 64, 65, 68] from light field images. In addition, the technique of estimating the characteristics of the object surface using the multi-view information of the light field [22, 44, 73] has been studied steadily.

Most of these studies assume the input is a sharp light field image without motion blur. However, motion blur in light field images is as frequent as that in the images of single-lens cameras in real-world scenario. If the image is taken in a low light envi-



(a) Input Image (b) Deblurred Image

(c) Depth Map (d) Visualized Blur Kernels

Figure 5.1: A blurry light field is our input. (a) Center view of input blurred light field image. (b) Deblurred image of (a). (c) Estimated depth map. (d) Visualization of global camera motion in point-wise kernels. The proposed algorithm jointly estimates latent image, camera motion and depth map from single light field image.

ronment or the camera is shaken largely during the shutter time, the degree of motion blur often become severe. Since the blurred image loses information regarding fine texture and object boundaries, directly applying a conventional matching algorithm to the blurred light field produces unsatisfactory results.

Many studies have been conducted on image deblurring where the images are assumed to be captured by using conventional cameras, and there are algorithms that can be used in real-world situations [12, 30, 35, 69, 67]. However, it is inappropriate

to apply the existing single image deblurring algorithms to the light field image. To remove the motion blur of the light field image using the conventional single image deblurring method, each sub-aperture image of the light field must be independently deblurred. This approach is computationally expensive and cannot guarantee photo consistency between sub-aperture images after deblurring. Therefore, the relation between sub-aperture images should be considered in terms of multi-view geometry in order to model the motion blur of light field image. Although several techniques for removing motion blur for light field images have been proposed [10, 14, 34, 55], these methods are based on unrealistic assumptions about the scene structure or the motion of the camera.

In this paper, we propose a joint estimation method that solves light field deblurring, depth estimation, and camera motion estimation in a unified framework. In the proposed method, the blur operation is parameterized as a function of the camera motion and pixel-wise depth, which can cover general camera motions and non-planar scene structures. The energy model model includes the observations from the all sub-aperture images. Since each sub-aperture image has slightly different camera motion when the camera moves in 6 degree-of-freedom (DOF) motion, use of all the observation prevents the camera motion from falling into local minima and helps to restore a sharp depth map. While the proposed energy model model includes all the motion blur of sub-aperture images, only the center view image, depth, and camera motion are estimated, since the variables of each sub-aperture image can be described by using the center view variables with a slight approximation; the latent images and depth maps of other viewpoints can be approximated by using those of the center view, given that the calibrations between sub-aperture images are given for a specific light field camera [7, 15]. This approximation allows the proposed model to use the rich information

from the multi-view observations while keeping the number of variables to be the same as that of single image deblurring and depth estimation. The main contribution of this method is in proposing a blind deblurring algorithm that can be applied to general motion and depth in the light field. The proposed model can recover not only the latent image but also the sharp depth map. Also, the multi-view constraints of our algorithm can be applied directly to the general multi-view camera environment.

5.2 Related Works

In this section, we briefly summarize previous works to single image deblurring and light field deblurring papers. It is challenging problem to deblur an image with motion blur caused by general camera motion and scene structure with non-uniform depth, since it forms a spatial-variant blur kernels. Estimating these blur kernels independently for each pixel is computationally expensive. One way to effectively remove the spatial-variant motion in single image deblurring is to first find the Motion Density Function (MDF) and then generate the pixel-wise kernel from it [23, 30, 29]. Gupta *et al*. [23] model the camera motion in discrete 3D motion space consisting of x, y translation and in-plane rotation. They deblur the image by iteratively optimizing the MDF and the latent image that best describe the blur image. A similar model is used in Hu and Yang [30], which models MDF with three-dimensional rotations. These methods of using MDF well parameterize the spatial-variant blur kernel into low dimensions. However, when the depth variation of the image is large, the complexity of the model increases. Since the motion blur is determined not only by the camera's motion but also by the depth of the scene, it is difficult to model the motion blur using MDF only in depth varying images. In Hu *et al*. [29], the image is segmented by using a matting

algorithm, and the MDF and the representative depth values of each region are found through the expectation-maximization algorithm. In the stereo camera case, Xu and Jia [71] decompose the regions of the image according to depth map and recombine after removing the blur independently. The method of decomposing image regions works well on images with only flat objects, but when the surfaces of the objects are uneven, the problem becomes more challenging.

A light field image is effectively same as a set of multi-view images with narrow baseline, and contains rich 3D geometry information in the single-shot image. This nature of light field can help blind deconvolution problem by adding multi-view constraints to optimization. In recent years, several approaches [10, 14, 34, 55] have proposed to address motion blur on light field, but all of them are based on simplifying assumptions regarding their blur models, and cannot be applied to deblur light field images with general camera motions and scene structures. Chandramouli *et al*. [10] addressed motion blur in light field for the first time. They assumed constant depth and uniform motion (shift-invariant) due to the complexity of imaging model. This assumption can only be applied when the objects are far away from camera, and the advantage of using light field image is rather unclear for this situation.

The method proposed by Snoswell and Singh [55] decomposes the scene using the depth planes and deblur the image belonging to each plane independently. This method can handle scene depth variations, but only can be applied to deblur x-axis linear motion blur. Jin *et al*. [34] quantized the depth map in two layers and remove motion blur in each layer. Their method assumes that the camera motion is in-plane translation, and exploits depth value as a scale factor of translation motion. The last two methods highly depend on initial depth estimation, but the ignored that the depth of the object boundary is not accurate in blurred light field. Furthermore, although

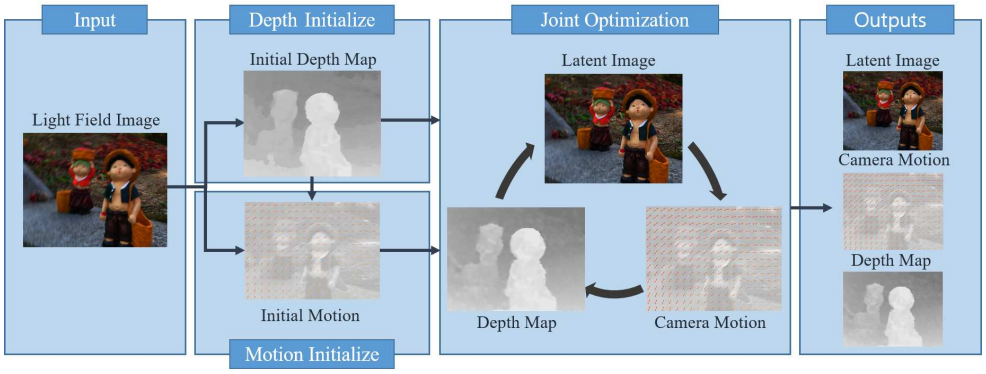


Figure 5.2: The pipeline of the proposed algorithm is visualized. First, depth and global camera motion is initialized using conventional depth estimation method and blur kernel estimation method (Section 5.3.2). Then, the main algorithm estimates the latent image, camera motion, and depth map all together in iterative and alternating optimization (Section 5.4). The final outputs are obtained after convergence of joint optimization.

their models handle non-uniform blur kernel related to depth map, more general depth variation and camera motion should be considered to apply to real world scenes.

Dansereau *et al*. [14] directly apply the Richardson-Lucy deblurring algorithm to light field to address the 6 DOF motion blur. Their method can deal with the most general form of motion blur, but assumes the ground truth camera motion is given. This is non-blind deconvolution problem, which cannot be applied to real-world images without aid of external devices or algorithms to get the accurate camera motion.

The proposed model addresses the blind deconvolution problem in light field and thoroughly handles 6 DOF motion blur and scene depth variation. Although, the proposed algorithm is described in terms of deblurring light field images, it can also be applied to general multi-view images.

5.3 Deblurring with 4D Light Fields

A pixel in a 4D light field $L(x, y, u, v)$ has four coordinates, first two for an image (x, y)-coordinate and the rest for an angular (u, v)-coordinate. Considering an angular coordinate vector records the position where the light passed through the camera lens, it is known that a light field can be decomposed in to a set of $u \times v$ multi-view images, often called sub-aperture images. Throughout this paper, we represent a sub-aperture image of a light field as $I^{\mathbf{u}}$ and its pixel value at a position as $I^{\mathbf{u}}(\mathbf{x})$, where $\mathbf{u} = (u, v)$ and $\mathbf{x} = (x, y)$, respectively. Each sub-aperture image has its own imaginary camera pose $\mathrm{P}^{\mathbf{u}}$ which is relatively defined by using the camera pose of center sub-aperture image $\mathrm{P}^{\mathbf{c}}$. The relationship between $\mathrm{P}^{\mathbf{u}}$ and $\mathrm{P}^{\mathbf{c}}$ is fixed and can be obtained by using calibration tools, such as [7].

We model the motion blur of the light field as using the 6 DOF motion of camera from shutter opened to closed. The minimal representation of a 3D camera pose is $\mathbf{p} \in \mathbb{R}^6$, which represents the displacement and orientation of the camera in $\mathfrak{se}(3)$space. The camera pose $\mathbf{p}$ is converted into a 3D rigid transformation matrix as follows.

$$\mathrm{P} = \begin{pmatrix} \mathrm{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \tag{5.1}$$

where $\mathrm{P} \in \mathbb{R}^{4\times4}$, and is a member of $\mathbb{SE}(3)$group. $\mathrm{R} \in \mathbb{R}^{3\times3}$ is rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is translation vector.

5.3.1 Motion Blur Formulation in Light Fields

We follow the imaging model in 3.7 to approximate the capturing process of the light field images. The blur model for the center view $\mathbf{c} = (u_c, v_c)$ of the light field below

this assumption is as follows:

$$B_{\mathbf{c}}(\mathbf{x}) \approx \Psi_{\mathbf{c}} \circ I_{\mathbf{c}}(\mathbf{x}). \tag{5.2}$$

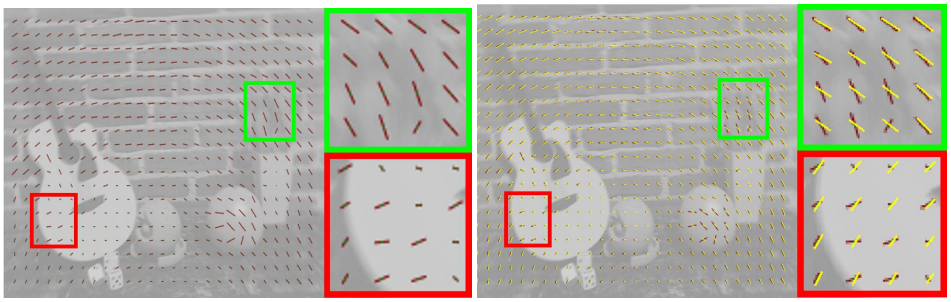
In the above, motion blur is modeled only for the center view image. To deblur the light field image, we must deblur all the angular domain images. Independently deblurring them does not guarantee consistent results between different views, and is computationally heavy. However, using the light field calibration algorithm [7], the relative pose between the center view and each view $\mathrm{P}_{\mathbf{u}}$ can be obtained. The angular coordinates of the light field keep the relative pose rigid while the camera is moving, so the camera pose of each coordinate $\mathbf{u}$ at t_o can be expressed as follows:

$$\mathrm{P}_{\mathbf{u}}^{t_o} = \mathrm{P}_{\mathbf{c}}^{t_o} \cdot \mathrm{P}_{\mathbf{u}}, \tag{5.3}$$

where $\mathrm{P}_{\mathbf{u}}^{t_o}$ is the relative pose of view $\mathbf{u}$ when the shutter is open. We can also warp $I_{\mathbf{c}}^{t_c}$ to intermediate image $B_{\mathbf{u}}^{\tau_m}$ of view $\mathbf{u}$ using $\mathrm{P}_{\mathbf{u}}^{t_o}$ and $D_{\mathbf{c}}^{t_c}$. This means that the capturing operator $\Psi_{\mathbf{u}}$ can be parameterized with $\mathrm{P}_{\mathbf{c}}^{t_o}$ and $D_{\mathbf{c}}^{t_c}$. Therefore, it is possible to generate blur images of all views using only $I_{\mathbf{c}}^{t_c}$, $D_{\mathbf{c}}^{t_c}$ and $\mathrm{P}_{\mathbf{c}}^{t_o}$, which are all variables in center view.

5.3.2 Initialization

Since deblurring is an ill-posed problem, we should start with good initials. In our method, all of the latent image, camera motion, and depth map of the center view should be estimated. As we mentioned in Section 3.1, these three variables can be modeled in closed form in one equation. Therefore, if we initialize two out of three, we can find the remaining variables that satisfy the condition. First, depth can be easiest to initialize using the multi-view benefits of light field. We use the existing light



(a) The initial local linear motions from [58]

(b) Refinement of local blur kernels from (a) using global camera motion

Figure 5.3: Refinement of local blur kernels using the global camera motion constraint can correct the errors of initialization. The yellow lines in (b) indicates that the orientation and size of the blur kernels after the refinement, which is more plausible than the initialization.

field depth estimation algorithm [68] for initialization. While it gives the result with uncertain object boundaries, this is sufficient for initialization. Our algorithm later refines the depth boundaries.

We initialize the camera motion as the next variable. The blur at the boundary of an object or texture contains information about the direction and size of the local blur kernel. We first estimate the coarse local linear motion of the center image using [58]. Our goal is to find a global camera motion $\mathbf{p}_{\mathbf{c}}^{\mathbf{to}}$ from the coarse and local linear blur kernels. A pixel point can be mapped to 3D space through the initial depth map, and it can be moved using the camera motion matrix. Each of these 3D points can be re-projected as a 2D image coordinate. We fit the pixel coordinates moved by the linear kernels and this re-projected coordinates as follows.

$$\min_{\mathbf{p}_{\mathbf{c}}^{\mathbf{to}}} \sum_{i=1}^{N} \| W^{\mathbf{c}_{t_o} \rightarrow \mathbf{c}}(\mathbf{x}_i) - \mathbf{x}_i^l \|_2^2 \tag{5.4}$$

where $\mathbf{x}_i$ is sampled pixel position, and $\mathbf{x}_i^l$ is the point that $\mathbf{x}_i$ is moved by the linear kernel.

Global camera motion is obtained by fitting the coordinates shifted by warping function $W^{\mathbf{c}_{t_o}\rightarrow\mathbf{c}}(\cdot)$ and $\mathbf{x}_i^l$. N is the number of sample pixels. N is fixed to 4 in our algorithm. However, the linear kernels used for fitting do not give direction of the motion. Because of this ambiguity, we randomly assign orientation of each linear kernel when fitting step.Another problem is that local motion candidates estimated in a patch-wise manner contain a lot of outliers and noise. So we use **RANSAC** [21] to solve the ambiguities in orientations of the linear kernels and find the camera motion vector that describes the reliable linear kernels well. The initial blur kernels and the result of refinement by using **RANSAC** are visualized in Figure 5.3.

5.4 Joint Estimation

5.4.1 Energy Formulation

Resolving the light field deblurring in our model is to recover $I_{\mathbf{c}}^{t_c}$, $D_{\mathbf{c}}^{t_c}$ and $\mathrm{P}_{\mathbf{c}}^{t_o}$, which describe the given multi-view blurred images well. To solve $I_{\mathbf{c}}^{t_c}$, $D_{\mathbf{c}}^{t_c}$ and $\mathrm{P}_{\mathbf{c}}^{t_o}$, we minimize the following energy.

$$\begin{aligned} E = &\sum_{\mathbf{u}}\sum_{\mathbf{x}} \lambda_u \left\| B_{\mathbf{u}}\left(W^{\mathbf{c}\rightarrow\mathbf{u}}(\mathbf{x})\right) - \Psi_{\mathbf{u}} \circ I_{\mathbf{c}}^{t_c}(\mathbf{x}) \right\|_1 \\ &+ \rho_L \left\| \nabla I_{\mathbf{c}}^{t_c} \right\|_2 + \rho_D \left\| \nabla D_{\mathbf{c}}^{t_c} \right\|_2 \end{aligned} \tag{5.5}$$

$\mathbf{u} \in \mathbb{N}^2$ is the angular domain, $\mathbf{x} \in \mathbb{N}^2$ is the image domain of the light field.

The first term is the data term containing all the angular coordinates. Multi-view data term prevents latent variables from falling to local minimum. In practice, we use different weight λ_c for the center view ($\lambda_c > \lambda_u$). We use the L1-based blur model

for the data term according to [35], which is more effective in removing ringing of object boundaries and gives more robust deblurring results in images with large depth changes.

The last two terms are the total variation (TV) regularizers [4] for the latent image and depth map, which have fixed weights λ_L and λ_D. In our energy, $D_{\mathbf{c}}^{t_c}$ and $\mathrm{P}_{\mathbf{c}}^{t_o}$ are implicitly included in the capturing operator $\Psi_{\mathbf{u}}$. Since warping operator has serious nonlinearity, it is complicated to optimize these three variables at once. So our strategy is to optimize these variables in an iterative and alternative way. We use a gradient method to minimize one variable while the other two are fixed. This is done in turn for three variables to find three variables that minimize energy.

Our entire algorithm is given in Algorithm 1 and the processing pipeline is visualized in Figure 5.2. Directly optimizing the L1-norm of the energy function is a difficult problem. We approximate L1 optimization using iteratively reweighted least squares (IRLS) method. Our algorithm converges in a small iterations(< 10) from the given initial $D_{\mathbf{c}}^{t_c}$ and $\mathrm{P}_{\mathbf{c}}^{t_o}$. Figure 5.4 shows the intermediate results of the proposed algorithm at each iteration.

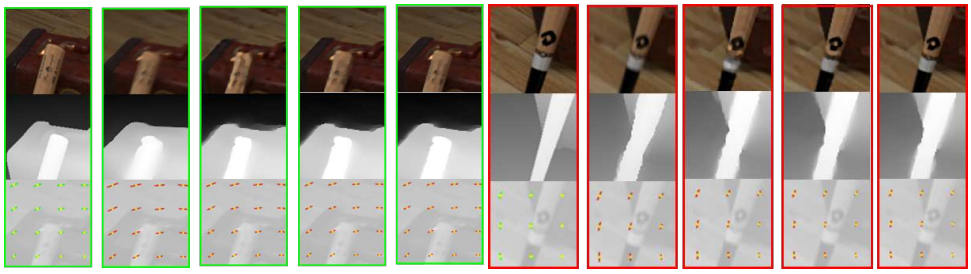
5.4.2 Update Latent Image

Proposed algorithm first update the latent image using the initialized depth map and camera motion. In our data term, the capturing operation is linear operation for fixed camera motion and depth map. When updating the latent image, it is like minimizing the energy as follow.

$$\min_{I_{\mathbf{c}}^{t_c}} \sum_{\mathbf{u}} \|B_{\mathbf{u}} - \mathcal{K}_{\mathbf{u}} \cdot I_{\mathbf{c}}^{t_c}\|_1 + \rho_L \|\nabla I_{\mathbf{c}}^{t_c}\|_2 \tag{5.6}$$

where $B_{\mathbf{u}}$ and $I_{\mathbf{c}}^{t_c}$ are vectorized images, and $\mathcal{K}_{\mathbf{u}}$ are the blur operations in square matrix form. The update of latent image solves linear equations for L1-norms with IRLS. TV regularization serves to guide the latent image with a clear boundary while eliminating the ringing of the solution.

(a) A center-view image



(b) Progress of the optimization for the patch in the green box

(c) Progress of the optimization for the patch in the red box

Figure 5.4: Visualization of iterative update during the joint optimization using a synthetic dataset, **Baseball**. For (b) and (c), First column shows ground truth, second column shows initial variables and the remaining columns show iteration 1, 3, 5 of the joint optimization. Starting with incorrect initial variables, the intermediate results get closer to ground truth in every iteration.

Algorithm 1 Blind Motion Deblur for Light Fields

Input: Blurry light field image $B_{\mathbf{c}}$

Output: Latent Image $I_{\mathbf{c}}^{t_c}$, Camera Motion $\mathrm{P}_{\mathbf{c}}^{t_o}$, Depth $D_{\mathbf{c}}^{t_c}$ $\%$ Initialization

Initialize $D_{\mathbf{c}}^{t_c}$ using [68]

Initialize local lindear blur kernels $\mathbf{x}^l$ using [58]

for $t \leftarrow 1, T$ **do**

Randomly select N pixels in $\mathbf{x}^l$

Compute $\mathrm{p}_{\mathbf{c}}^{t_o}$ by fitting (Eq. 5.4)

Update $\mathrm{p}_{\mathbf{c}}^{t_o}$ when $W^{\mathbf{c}_{t_o}\rightarrow\mathbf{c}}(\cdot)|_{\mathrm{p}=\mathrm{p}_{t_o}^{\mathbf{c}}}$ get most inliers in $\mathbf{x}^l$

end for

$\%$ Joint Optimization

for $i \leftarrow 1, \mathcal{M}$ **do**

Update $I_{\mathbf{c}}^{t_c}$ from current states $D_{\mathbf{c}}^{t_c}$ and $\mathrm{P}_{\mathbf{c}}^{t_o}$ (Eq. 5.6)

Approximate Energy (Eq. 5.5) using (Eq. 5.7)

Update $\mathrm{P}_{\mathbf{c}}^{t_o}$ and $D_{\mathbf{c}}^{t_c}$ solving (Eq. 5.8 and 5.9)

end for

5.4.3 Update Camera Pose and Depth map

Unlike the update of latent image, the derivative of capturing operation must be calculated for $\mathrm{P}_{\mathbf{c}}^{t_o}$ and $D_{\mathbf{c}}^{t_c}$. Since the 3D warping operation is a non-linear operation on the pose and depth, we use a first-order approximation of the capturing operation as follows.

$$\Psi_{\mathbf{u}} \circ I_{\mathbf{c}}^{t_c}(\mathbf{x}) \approx B_{\mathbf{u}}^{0}(\mathbf{x}) + \frac{\partial(\Psi_{\mathbf{u}} \circ I_{\mathbf{c}}^{t_c}(\mathbf{x}))}{\partial \mathbf{p}_{\mathbf{c}}^{t_c}} \cdot \Delta\mathbf{p}_{\mathbf{c}}^{t_c} \tag{5.7}$$

where $B^0_{\mathbf{u}}(\mathbf{x})$ is a blurred image created by simulating motion blur with the current state of variables. It is modeled as follows:

$$\min_{\Delta \mathbf{p}_{\mathbf{c}}^{t_c}} \sum_{\mathbf{u}} \sum_{\mathbf{x}} \| \tilde{B}_{\mathbf{u}}(\mathbf{x}) - \frac{\partial(\Psi_{\mathbf{u}} \circ I_{\mathbf{c}}^{t_c}(\mathbf{x}))}{\partial \mathbf{p}_{\mathbf{c}}^{t_c}} \cdot \Delta \mathbf{p}_{\mathbf{c}}^{t_c} \|_1 \tag{5.8}$$

where $\tilde{B}_{\mathbf{u}}(\mathbf{x}) = B_{\mathbf{u}}(\mathbf{x}) - B^0_{\mathbf{u}}(\mathbf{x})$.

Applying same method, the capturing operator can be expressed by the first approximation for the depth as follows:

$$\begin{aligned} \min_{\Delta D_{\mathbf{c}}^{t_c}} \sum_{\mathbf{u}} \sum_{\mathbf{x}} \| \tilde{B}_{\mathbf{u}}(\mathbf{x}) - \frac{\partial(\Psi_{\mathbf{u}} \circ I_{\mathbf{c}}^{t_c}(\mathbf{x}))}{\partial D_{\mathbf{c}}^{t_c}(\mathbf{x})} \cdot \Delta D_{\mathbf{c}}^{t_c}(\mathbf{x}) \|_1 \\ + \rho_D \| \nabla D_{\mathbf{c}}^{t_c} \|_2 \end{aligned} \tag{5.9}$$

In the case of depth, TV regularization preserves discontinuity and obtain smoothed result.

Since the first approximated energy function is a linear equation for motion and depth, the variables can be estimated effectively through IRLS. The resulting $\Delta \mathbf{p}_{\mathbf{c}}^{t_c}$ and $\Delta D_{\mathbf{c}}^{t_c}(\mathbf{x})$ are incremental values for the current $\mathrm{P}_{\mathbf{c}}^{t_o}$ and $D_{\mathbf{c}}^{t_c}(\mathbf{x})$. These variables can be updated as follows.

$$\begin{aligned} D_{\mathbf{c}}^{t_c}(\mathbf{x}) &= D_{\mathbf{c}}^{t_c}(\mathbf{x}) + \Delta D_{\mathbf{c}}^{t_c}(\mathbf{x}) \\ \mathrm{P}_{\mathbf{c}}^{t_o} &= \exp(\Delta \mathbf{p}_{\mathbf{c}}^{t_c}) \cdot \mathrm{P}_{\mathbf{c}}^{t_o} \end{aligned} \tag{5.10}$$

where $\mathrm{P}_{\mathbf{c}}^{t_o}$ is updates through the exponential mapping of the motion vector $\Delta \mathbf{p}_{\mathbf{c}}^{t_c}$.

5.5 Experimental Results

5.5.1 Synthetic Data

We use the approach in Section 4.2 to generate synthetic light field data. Our synthetic light field camera has a 3x3 angular resolution, and the camera calibration of all views

Table 5.1: Quantitative results of the proposed method for image deblurring. The performance is measured by comparing the deblurring results to the corresponding ground-truth sharp images using the synthetic datasets. The results from other competing methods are shown for comparison. Note that, for each method, this table shows the highest PSNR value among all the intermediate sharp images during the shutter time.

	Static Scene			**Fruit**			**Baseball**		
	forward	rotation	translation	forward	rotation	translation	forward	rotation	translation
input	27.52	24.51	27.39	27.38	25.12	23.63	33.13	32.97	31.08
Li xu [29]	26.61	22.24	28.72	25.89	23.48	25.50	30.87	29.47	32.33
T.H. Kim [35]	27.62	24.50	28.01	26.23	24.79	24.26	32.39	31.93	31.49
Jian Sun [58]	28.59	24.91	28.11	27.29	25.54	23.94	33.54	33.73	31.66
ours	30.63	27.19	30.66	29.33	28.78	24.68	34.55	35.08	33.50

is known. We render 9 light field images by generating 3 different camera motions for 3 different Blender [1] models. The three camera motions consisted of translation, rotation and forward motion, respectively. The blur image is generated by averaging the sharp images of the intermediate frames while the camera is moving. All our synthetic data contain non-uniform blur due to scene depth variation.

Comparision on deblurring

We evaluate the performance of [29, 35, 58] and our approach by measuring peaksignal-to-noise-ratio (PSNR) with ground truth sharp image. Table 6.1 shows the highest PSNR value among all the intermediate sharp images used to create the blur image. We observe that our algorithm showed better performance than other single image deblurring algorithms. This is because our algorithm properly models the size and shape of the kernel by 3D motion and depth variations. When the motion blur is formulated

assuming a constant depth [29], the overall performance is low. [35, 58] show low performance in rotation motion. [29] can not handle the blur kernel difference due to depth variations, so the artifacts occur in areas with depth discontinuities. In [35, 58], there is a limit to the size of the motion that can be estimated from the local blur region. Therefore, the sharp image can not be recovered well in large blur kernels at the image boundary with rotation or forward motion. [35, 58] is our baseline method for motion initialization. Even though started in the same motion, our algorithm converges to better deblurring results regardless of the type of motion. Figure 5.5 and Figure 5.6 shows how the proposed algorithm improves over the baseline initialization method.

5.5.2 Real Data

The performance of the proposed algotirhm is also tested by using real light-field images taken by a **Lytro ILLUM** camera. We compared the deblurring results of other algorithms [29, 35, 58] with various images. The results are shown in Figure 5.7. The deblurring performance of the proposed algorithm clearly outperforms the others as the magnified views of small regions with complex and meaningful textures indicate.

More specifically, the first column of Figure 5.7 shows that originally unrecognizable English letters become clearly recognizable after deblurring in the proposed method while the results of other methods except the result of [29]. The second column shows that the proposed algorithm can handle a scene with large depth variation with many depth discontinuities. While the other methods fails to reconstruct the plausible deblur image around the face of the dolls, the proposed algorithm shows visually pleasing results. This is mainly due to the other deblurring algorithm takes too much attention to the larger background areas. Note that the pixels in the background are also well deblurred in the proposed method. Third column shows that even the

scene structure is very complex, containing bunch of thin and tiny objects with many occlusions, the proposed method works reliably. The last column also shows the performance of the proposed deblurring algorithm is significantly better than the other competing methods; the shape of the bicycle sign is most clear in the result of the proposed method.

5.6 Conclusion

In this chapter, we present a deblurring algorithm that recovers latent image, camera motion and sharp depth map in light field. We propose a multi-view constraint that efficiently utilizes the angular domain of the light field. We develop an algorithm to initialize the 6 DOF camera motion from the local linear blur kernel and scene depth. In joint optimization, we solve the problem in an effective form for IRLS by first approximating the nonlinear 3D warping function. Evaluation on synthetic and real data reveals that our model works well with all kinds of spatial-variant motion and scene depth variation.

Figure 5.5: The qualitative results of deblurring are visualized for synthetic datasets. Each column shows the results of different datasets; **Baseball**, **Fruit**, and **Static Scene** from the left to right. The first two rows show the input images and the corresponding ground-truth clean images. The following rows show the results obtained by using the baseline initialization method [58] and the proposed method. The estimated blur kernels of each method are visualized in the last two rows. Green lines represent the ground-truth blur kernels while the red lines represent the estimated blur kernels.

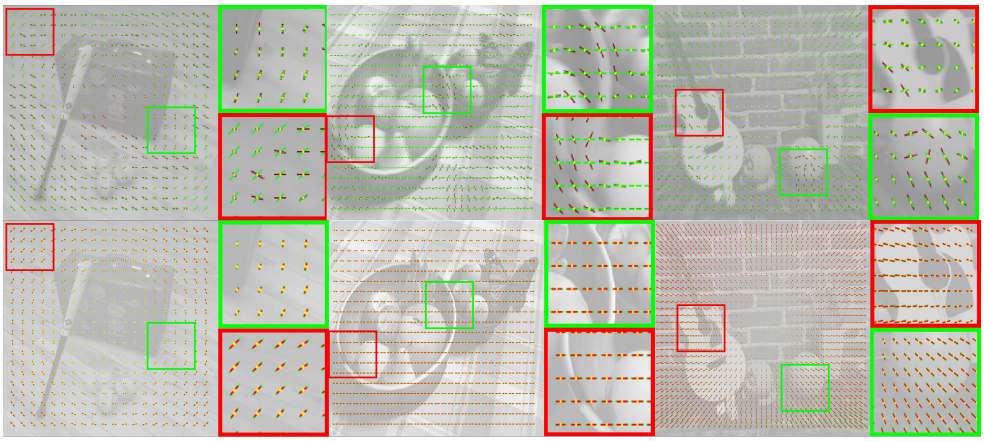


Figure 5.6: The qualitative results of deblurring are visualized for synthetic datasets. Each column shows the results of different datasets; **Baseball**, **Fruit**, and **Static Scene** from the left to right. The estimated blur kernels of the baseline initialization method [58] and the proposed method are visualized in each row. The green lines represent the ground-truth blur kernels while the red lines represent the estimated blur kernels.

Figure 5.7: The qualitative results of different deblurring methods are visualized for real datasets. Each column shows the results of different datasets. The first row shows the input images, and the following rows show the results obtained by using the methods in [29], [35], [58], and the proposed method.

Chapter 6

A Unified Framework for a Monocular Image Sequence

6.1 Introduction

Structure from motion or multi-view stereo has been one of the most interesting problems in computer vision. The goal of this problem is to determine the underlying 3D scene structure and camera configuration from multiple images. Despite the inherent difficulty of this problem being an actually inverse one, contemporary algorithms indicate satisfactory performance in public datasets [28, 53]. Based on these results, multi-view stereo is now actively applied to many interesting applications: the reconstruction of cultural heritage sites [63] for archiving, or reconstructing cities for virtual tourism [3, 54].

Although these successes are very encouraging, there still are some limitations that prevent the methods from being reliable in more realistic applications. For example, the algorithms in [63] or [3] might not result in the high-quality solutions as in the papers, when applied to a video captured by a moving hand-held camera. One main reason to blame is the simple brightness constancy assumption that many multi-view stereo algorithms rely on. The assumption is false when the images have different

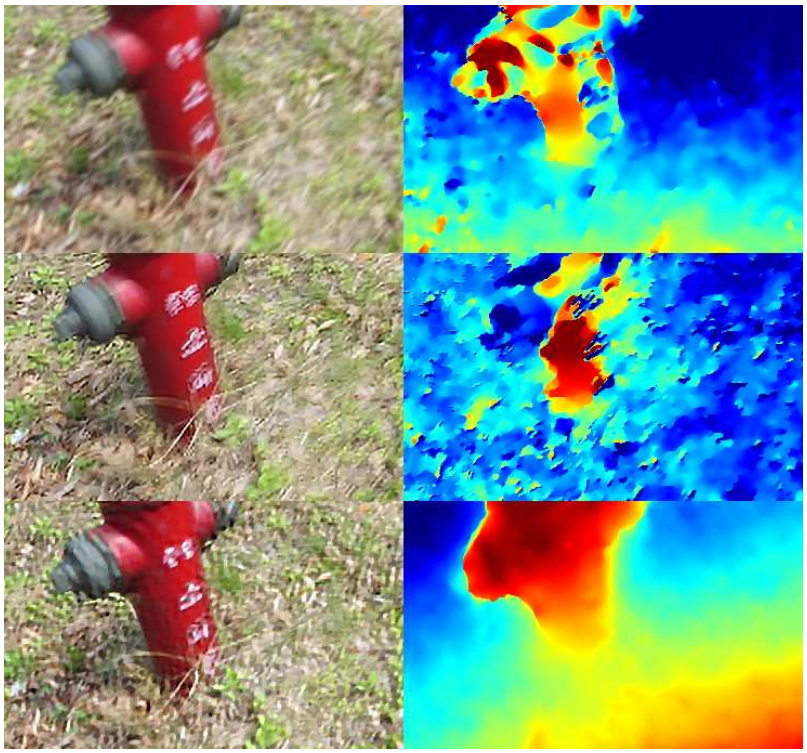


Figure 6.1: Comparison of depth estimation and image restoration results on blurry, LR images. The left column shows the estimated latent images, with their corresponding depth maps on the right column. From top to bottom, the images are obtained by (a) a simple bicubic interpolation, (b) independent use of the super resolution [61] after applying the deblurring algorithm [72], (c) the proposed method, respectively. The depth maps for the first two rows are estimated by using baseline variational depth estimation.

illuminations/exposures [28] or have severe noise [27].

However, even more challenging facts about the real-world images are blurs caused by camera motion [41, 49] or insufficient resolution [5, 42]. This becomes serious in the case of hand-held cameras or cameras attached to moving vehicles. Blur operation

acts differently on each pixel and it breaks BR assumption even for the images with the same illuminations/exposures. When we have low-resolution (LR) image frames, the correspondence problem becomes much harder. Note that even in a high-resolution (HR) image, the effective resolutions of farther objects are still low. This causes severe error in both detection and matching features in distant areas, resulting unsatisfactory reconstructions of camera poses and scene depths. Furthermore, if we have image frames corrupted by blur and low resolution at the same time, the problem becomes extremely challenging.

One straight forward approach to this problem is to apply existing super resolution [61] or debluring method [72] before matching. This preprocessing might produce visually pleasing images. However, it often leads to worse results than the use of original images in terms of matching, as shown in Figure 6.1 (a) and (b). It tells us that the image restoration problems and multi-view stereo matching problems are inter-related and should be handled jointly.

In this chapter, we consider the four inter-related problems of camera pose estimation, dense depth reconstruction, deblurring and super-resolution as a whole and cast them in a single problem by formulating it as a unified energy function. A generative image formation model is proposed by analyzing the imaging process in terms of the multi-view geometry. In this model, the input images are assumed to have been obtained by a mapping of the target variables i.e., the latent images, depths, and camera poses. Thus, the solution is obtained by minimizing the energy that penalizes the difference between the simulated LR images and the observed input.

To the best of our knowledge, this is the first study that solves the four challenging problems jointly in a single framework. The proposed method clearly outperforms the independent use of existing techniques for each problem. Furthermore, the blur model

used in the proposed energy function differs from the conventional optical-flow based ones as in [49] and [41]. By exploiting the multi-view geometry more explicitly, it can handle more general blur kernels possibly originating from camera rotations and forward motions.

We briefly summarize related works in Section 6.2. In Section 6.3, the proposed blur model and the notations are introduced. The differences from the conventional blur model are also clarified. In Section 6.4, the definition of the proposed energy function is presented. We briefly explain the optimization strategy regarding it in Section 6.5. The performance of the method is tested on both synthetic and real datasets, and the results are reported in Section 6.6. The final conclusion and discussions are made in Section 6.7.

6.2 Related Works

Attempts to perform image matching on blurry images have been conducted by a few researchers. Portz *et al*. [49] proposed an optical flow method that uses blur-aware matching firstly proposed in tracking methods [32, 45]. The idea is based on the assumption of commutativity of blur operations. To obtain the correspondence between two images, it first blurs the images with the kernels of each other, instead of deblurring them by their own kernels. The underlying insight of it is that modeling blur operation as a function of correspondence is much simpler than the modeling deblurring operation.

Recently, Lee *et al*. adopted this idea and proposed methods to effectively handle blurred input images in camera pose estimation [39] and dense stereo matching [41] problems. However, considering the fact that the scene depth and camera motion can

generate the exact blur kernels only when the both values are correct, their approach of estimating them separately is inappropriate. More importantly, the works mentioned above [41, 49] are limited by a simple assumption that may not hold for general images taken by moving hand-held cameras. Both of them model the blur kernel by using the optical flow vectors of the previous frame. They use the same optical flow maps with different weights according to the time stamps to generate intermediate frames during the shutter time. This assumption is true only when the optical flows are constant across the pixels. Therefore, it would fail if the motion of camera is rotational or out of the image plane, where the optical flows vary largely in spatial domain.

In contrast, the proposed blur model, which will be explained in Section 6.3, covers more general camera motions by adopting linear model in $\mathbb{SE}(3)$space. The blur kernel is explicitly approximated by interpolating the camera path and depth maps between adjacent frames. Figure 3.1 shows the difference between the conventional blur model and the proposed blur model.

Researchers have also found that the resolution of the camera affects the accuracy of reconstructed depth values. The precision of the depth maps is bounded by the focal length regardless of the accuracy of the pixel-wise matching. Furthermore, the details of the textures often vanish in the LR images or regions, increasing ambiguity in matching. To overcome this limitation, some methods perform super resolution and multi-view stereo problem in one framework [5, 42]. It is shown that solving the stereo matching and super resolution jointly helps increase the accuracy of both reconstructed images and depth maps. However, the multi-frame super resolution framework used in [5, 42] works only when the matching information is accurate in sub-pixel units. Therefore, the joint estimation of super-resolved images and depth maps for blurry input images cannot be successful by these methods due to large errors in correspon-

dences.

6.3 Modeling Imaging Process

In this paper, we deal with an image sequence captured by a single moving camera, where the target scene is assumed to be static to enable stereo matching and camera pose estimation. In the following, the input images are denoted by B_t's, with t representing the time when the images are captured. Note that, in this chapter, we will omit the camera camera index c for simplicity, and slightly abuse the notation t to indicate both a specific time and the camera at that time. Am acquired image B_t is assumed to be the accumulation of the sensor output from the moment when the shutter of the camera opens and to the moment when it closes, as explained in Section 3.4. The capturing process for frame t is expressed by using a capaturing operator Ψ_t as in Equation (3.9).

Originally, the operator Ψ_t is dependent on D_t, $\mathbf{P}_t$, and $\mathbf{P}_{t_o}$, where t_o represents the time when the shutter opens for frame t. We can further simplifiy the dependency by utilizing the fact that we deal with sequential image frames taken by a single moving camera. The value of $\mathbf{P}_{t_o}$ can be approximated by assuming smooth camera motion in $\mathfrak{se}(3)$space, as follows:

$$\mathbf{P}_{t_o} = \exp\left(\frac{t_o - s}{t - s}\Delta\varepsilon_{st}\right)\mathbf{P}_s, \tag{6.1}$$

where $\Delta\varepsilon_{st} = \log\left(\mathbf{P}_t(\mathbf{P}_s)^{-1}\right)$, and s represents the time index of the frame previous to t. If we can estimate the relative pose transformation between two consecutive frames, s and t, with dense, pixel-wise correspondences, then we can compute the capturing operator Ψ_t.

6.4 Unified Energy Formulation

This study aims to estimate the latent images I_t's with the corresponding depth maps D_t's and the camera poses $\mathbf{P}_t$'s from a blurred, LR image sequence, B_t's. We assume that the intrinsic parameters are previously known and given as $\mathbf{K}$. As the target variables are interrelated, as mentioned earlier, the proposed method estimates them all together by optimizing a single unified energy function.

The total energy function E is defined by the sum of energy functions, E_t, defined for each one single frame. E_t is composed of three terms, each having different physical meanings,

$$E = \sum_t E_t, \tag{6.2}$$

$$E_t = E_t^m + E_t^s + E_t^r. \tag{6.3}$$

From left to right, the terms are called the matching term, self-consistency term, and regularization term, respectively.

The matching term, E_t^m, imposes photo consistency between corresponding pixels across the frames. The self-consistency term, E_t^s, makes the reconstructed latent image, I_t, consistent with the observed blurred, LR image, B_t. Finally, the regularization term, E_t^r, is used for smoothness of the solution I_t and D_t. In the following subsections, each term in Equation (6.3) will be explained in details.

6.4.1 Matching term

The first term relates the images from the consecutive frames based on the scene structure and camera motion. Since the target scene is static, the images warped into a specific frame should coincide if the warping is based on correct depth maps and camera poses. There could be small differences between the images, of course, because of

noise and occlusion.

The blur inherent in the input images should also be considered. In [41], the authors proposed to compensate for the different degradations caused by different blur operations between two images to be matched, by blurring them again with the blur operations of each other. The double-blurred images are then matched.

In the proposed matching term, we want to match the input blurred LR image, B_t, with the latent images of the neighboring frames, I_s's, where $s \in N(t)$ denotes the time index for neighboring frames of t. Therefore, the additional blur operation for matching is one-way; I_s's should be blurred and downsampled by the capturing operator Ψ_t. Finally, the matching term is

$$E_t^m = \sum_{s \in N(t)} \sum_{\mathbf{x} \in \Omega_{ts}} \left\| B_t(\mathbf{x}) - \Psi_t \circ I_s \left(W^{t \to s}(\mathbf{x}) \right) \right\|_1. \tag{6.4}$$

Note that the matching term only considers the pixels in the set Ω_{ts}, which represents the visible area of image domain at time t in terms of camera at s. How it is determined is discussed in Section 6.5.4 We use L1-norm because it is known to be more robust to the presence of noise and occlusion, and give reliable results [66].

In terms of multi-view stereo matching, the proposed methods try to find the plausible scene structure and camera poses that satisfy the brightness constancy assumption by minimizing the matching term. On the other hand, the same matching term is used for evidence of super resolution for reconstructing I_s's from LR observations in terms of the estimation of the latent images.

6.4.2 Self-consistency term

The term E_t^s is derived from the imaging process in Equation (3.9).

$$E_t^s = \lambda_s \sum_{\mathbf{x}} \left\| B_t(\mathbf{x}) - \Psi_t \circ I_t(\mathbf{x}) \right\|_1. \tag{6.5}$$

This equation serves to make the solution consistent with the observation. Given the depth maps and camera poses, capturing operator $\Psi_t(\cdot)$ is constant, and the equation is similar to the conventional data term in deblurring methods. By minimizing the above equation, we can obtain the latent image I_t.

In terms of the depth maps and camera poses, the self-consistency term imposes that the variables result in a plausible blur kernel. The blur kernel should ensure that the observed blurred LR image and the given latent image are well-matched. This helps to stabilize the depth map estimation. The parameter λ_s controls the strength of this constraint.

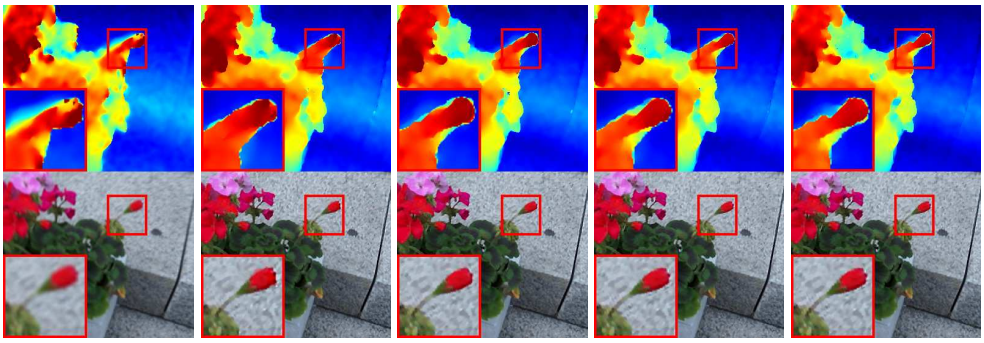
6.4.3 Regularization term

Although the matching term and self-consistency term can compensate each other, both terms rely on the possibly noisy input images. The additional term regularizes the solutions to suppress the errors. In the proposed framework, we use typical total variation (TV) priors for the depth maps and latent images. Although they were firstly introduced for denoising signals, the use of TV priors is now popular in the image deblurring problem [70], the super resolution problem [19], and even the stereo matching problem [50].

The TV priors used in the proposed method is defined as follows:

$$E_t^r = \lambda_d \sum_{\mathbf{x}} \|\nabla D_t(\mathbf{x})\|_2 + \lambda_h \sum_{\mathbf{x}} \|\nabla I_t(\mathbf{x})\|_2, \tag{6.6}$$

where $\nabla I(\mathbf{x})$ represents the gradient value of the image I at pixel $\mathbf{x}$. We use the magnitude of L2-norm to make the TV priors isotropic, while preserving the discontinuities in images and depth maps. The parameters λ_d and λ_h determine the degree of regularization on depth maps and latent images, respectively.



(a) initial (b) iteration = 1 (c) iteration = 2 (d) **iteration = 3** (e) iteration = 4

Figure 6.2: The convergence of solutions.

6.5 Optimization

Optimizing Equation (6.2) is complicated. It is a function of many variables (D_t's, $\mathbf{P}_t$'s, and I_t's for all frames), and it is highly nonlinear because of the warping operations. Therefore, instead of obtaining the global optimum, we attempt to secure a good approximated solution by taking several strategies. The core of it is divide-and-conquer strategy, i.e., an iterative and alternating optimization of variables. The proposed framework uses two-phase iterations, in which structures (cameras and depth maps) and latent images are alternatively updated.

The overall procedure for optimization is depicted in the Algorithm 2. The variable T denotes the number of frames in the input image sequence and *max_iter* is the number of iterations set by users. Figure 6.2 shows the convergence of solutions as the iteration goes on during optimization. The solutions almost converge after three iterations, which is the chosen value of *max_iter* of the proposed method. The top row shows the estimated depth maps and the bottom row shows the estimated latent images.

In the following subsections some details of the optimizations are elaborated.

Algorithm 2 The overall optimization procedure

% initialization

for $t = 1$ **to** T **do**

Initialize D_t, $\mathbf{P}_t$ by minimizing Equation (6.10)

end for

% main loop

for $iteration = 1$ **to** *max_iter* **do**

% first phase : update images

for $t = 1$ **to** T **do**

update I_t by minimizing Equation (6.9)

end for

% second phase : update depths and cameras

approximate Equation (6.2) using Equation (6.7)

update D_t's and $\mathbf{P}_t$'s by using IRLS

end for

6.5.1 Update of the depth maps and camera poses

In the first phase of each iteration, we optimize the variables on the scene structure, D_t's and $\mathbf{P}_t$'s, with fixed latent images, I_t's. The energy function then becomes similar to that of the variational framework for optical flow [57]. Therefore, we follow the conventional optimization scheme for this type of energy function as in [57]. This optimization itself is, again, iterative. In each iteration in this inner loop, the functions in the L1-norm for Equations (6.4) and (6.5) are approximated by using the first-order Taylor expansion at the current solution.

The linear approximation is conducted by calculating the partial derivatives of the

warping equation in terms of individual depth value and camera pose parameterized by the six-dimensional vector on $\mathfrak{se}(3)$. Suppose the current solution of our iterative algorithm lies at a point in the solution space, D_t^0, $\mathbf{P}_t^0$, and $\mathbf{P}_s^0$, warping of an image using the same warping from the frame at time t to s can be approximated, as follows:

$$
\begin{aligned}
&I^0(\mathbf{x}) = I\left(W^{s\to t}(\mathbf{x})\right)\big|_{D_t=D_t^0, P_t=P_t^0, P_s=P_s^0}, \\
&I\left(W^{s\to t}(\mathbf{x})\right) \\
&= I^0(\mathbf{x}) + \frac{\partial I^0}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{u}}{\partial D_t(\mathbf{x})}\Delta D_t(\mathbf{x}) + \frac{\partial \mathbf{u}}{\partial \varepsilon_t}\varepsilon_t + \frac{\partial \mathbf{u}}{\partial \varepsilon_s}\varepsilon_s\right).
\end{aligned}
\tag{6.7}
$$

In Equation (6.7), $\mathbf{u}$ is the flow generated by warping and is a function of depth and camera parameters. The partial derivatives are actually Jacobians [6].

Notably, $\Delta D_t(\mathbf{x})$, ϵ_t, and ϵ_s are the variables to estimate. Once computed, they contribute to the solution as follows:

$$
\begin{aligned}
D_t(\mathbf{x}) &= D_t^0(\mathbf{x}) + \Delta D_t(\mathbf{x}), \\
\mathbf{P}_t &= \exp(\varepsilon_t)\,\mathbf{P}_t^0, \\
\mathbf{P}_s &= \exp(\varepsilon_s)\,\mathbf{P}_s^0.
\end{aligned}
\tag{6.8}
$$

Because the all the terms in the L1-norm is now linearized, the variables can be efficiently estimated using the simple iteratively reweighted least square (IRLS) method [52].

6.5.2 Update of the latent images

The second phase of the outer loop is to optimize the latent images. Collecting the L1-norm functions for the target image I_t^H in the matching term, Equation (6.4), gives information about the different blur and sampling of the latent image I_t^H. The self-consistency term in Equation (6.5) and the smoothness imposed by the regularization term in Equation (6.6) are also considered. It results in frame-by-frame representation

of the energy function on I_t^H as follows:

$$\begin{aligned}
&\sum_{s\in N(t)}\sum_{\mathbf{x}\in\Omega_{ts}}\left\|\Psi_s\circ I_t\left(W^{s\to t}\left(\mathbf{x}\right)\right)-B_s\left(\mathbf{x}\right)\right\|_1\\
&+\lambda_s\sum_{\mathbf{x}}\left\|\Psi_t\circ I_t\left(\mathbf{x}\right)-B_t\left(\mathbf{x}\right)\right\|_1\\
&+\lambda_h\sum_{\mathbf{x}}\left\|\nabla I_t\left(\mathbf{x}\right)\right\|_2.
\end{aligned}\tag{6.9}$$

Optimizing the Equation (6.9) is simply finding the most plausible values that satisfy these competing constraints simultaneously.

Since we use the bilinear interpolation to sample color values of non-grid points in image warping, and a simple box filtering is used for downsampling in the capturing operation. This makes the warping and capturing operations be linear operators on the latent image once we fix the depth maps and camera poses. Consequently, the Equation (6.9) is a sum of L1-norm and L2-norm on linear functions of I_t. It can be easily optimized using the IRLS method [52].

6.5.3 Initialization

This type of iterative optimization could be sensitive to initialization. In our implementation, we first initialize the camera poses of the first two frames in the input data by computing the fundamental matrix between them using sparse feature point matching [26].

Once we have the camera poses of the first two frames, the depth maps and remaining camera poses can be initialized by minimizing the following equation sequentially frame-by-frame in a coarse-to-fine manner [57]:

$$\begin{aligned}
E_t^{init}=&\sum_{\mathbf{x}}\left\|\left(\Psi_t\circ B_s\right)\left(W^{t\to s}\left(\mathbf{x}\right)\right)-\Psi_s\circ B_t\left(\mathbf{x}\right)\right\|_1\\
&+\lambda_d\sum_{\mathbf{x}}\left\|\nabla D_t\left(\mathbf{x}\right)\right\|_2,
\end{aligned}\tag{6.10}$$

where s is used to represent the time of the previous frame. The estimated depth maps have low resolution. Therefore, we need to upsample it to match the resolution of the target latent images and then begin the main loop of the optimization. For upsampling, simple bicubic interpolation method is used.

6.5.4 Occlusion Handling

Although the use of L1-norm for the matching term in Equation (6.4) makes it robust to existence of occlusion, modeling the visible area in Ω_{ts} helps the recover better depth values around the discontinuities. To that end, we update the visible area Ω_{ts} whenever depth maps and camera poses are updated. Given updated depth maps and camera poses, Ω_{ts} is updated as follows:

$$\Omega_{ts} = \{\mathbf{x} \,|D_t(\mathbf{x}) > D_t(\mathbf{y}), \forall \mathbf{y} \in \Theta_{ts}(\mathbf{x})\}, \tag{6.11}$$

where Θ_{ts} represents the set of pixels in camera at time t that fall in the same area after warping.

$$\Theta_{ts}(\mathbf{x}) = \left\{\mathbf{y} \,\middle|\middle| W^{t\to s}(\mathbf{y}) - W^{t\to s}(\mathbf{x})\right| \leq 0.5\}. \tag{6.12}$$

6.6 Experimental Results

The validity of the proposed method is tested on both synthetic and real datasets. For the real datasets, we used the approach proposed in [74] for camera calibration. The additional information about the input dataset is shutter time and frames per second (FPS), which together are needed for interpolating the camera path and simulating blurs for each frame. This information can be obtained as metadata when we take an image sequence by using commercial cameras. When unavailable, we manually set the values to produce plausible results.

For comparison, we set the baseline as the simple variational matching method. It solves the same optimization problem as the proposed method, except that the images are fixed to upscaled version of blurry images by using bicubic interpolation and the capturing operations are missed in the energy terms.

The values of some parameters are empirically found. The proposed algorithm converges to good solutions when *max_iter* is 3 and M is 50. Also, we fixed the value of λ_s to be large (10) for all datasets because it should give strong constraints on the solutions. Only the value of λ_r is tuned on the basis of the dataset in the range of 2.0 to 5.0. The upscale factor of the method is set to 2.

One of possible limitations of the proposed framework is its computational complexity. It takes about five hours to process ten frames of 320×240 images in our Matlab implementation with quad-core 3.2GHz CPU. It can be improved, however, because many parts of the algorithm could be run in parallel on GPU.

6.6.1 Synthetic datasets

No public datasets provide blurry LR images with the corresponding ground-truth latent images and depth maps. Therefore, we simulate them by using the approaches in Chapter 4. While we have used depth-based image rendering technique to generate synthetic datasets using Middelbury datasets [28], the Mesa dataset is generated by using a full 3D model and smooth camera motion based on the Blender software [1].

The quantitative results are summarized in Table 6.1 with comparisons to the ground truths, results of the baseline, and results of the method in [41]. The results of [41] are obtained by using the implementation of the authors. Furthermore, we have tested the proposed method without super resolution by setting upscale factor to one to evaluate the effect of super resolution in the proposed framework.

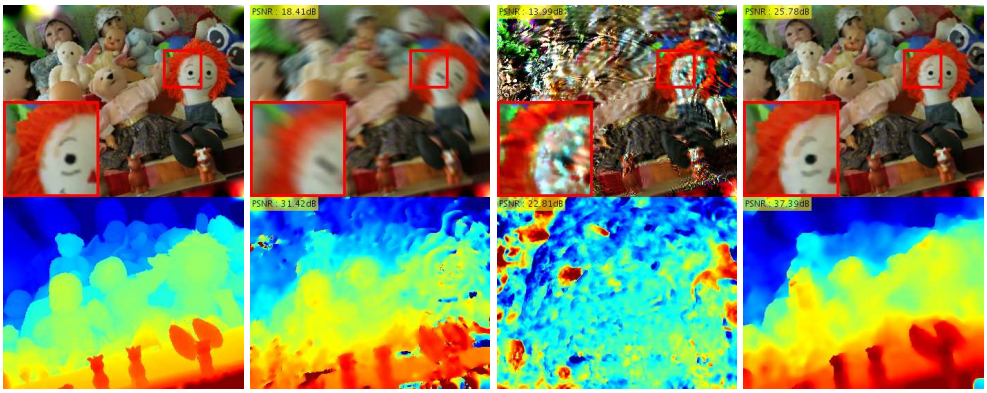
For most cases, the proposed method outperforms the other approaches. The performance of the proposed method is most impressive for the **Dolls** [28] dataset, because the camera motion includes rapid rotational moves. On the other hand, improvement on the **Mesa** [2] dataset is relatively small. This is because the amount of blur is small and the Blender's built-in blur model is different from the proposed one in the definition of the instant for latent images during shutter time. It is noteworthy that the use of super resolution clearly results in better accuracy in both camera pose estimation and depth estimation as well as the latent image estimation.

To evaluate competing methods quantitatively, we compute peak-signal-to-noise-ratio (PSNR) for the estimated latent images and depth maps with comparison to corresponding ground truths. Note that the images and depth maps can have different values according to the estimation of camera poses. Therefore, we first align the results to the ground-truth camera poses before computing PSNR. The images are warped using the ground-truth depth and camera poses and depth maps are warped and resampled.

The Dolls dataset

The camera rotates largely in this synthetic dataset. Since the method proposed by Lee *et al*. [41] cannot effectively handle the rotational motion of camera, the results of [41] are worse than the baseline for this dataset. On the contrary, the proposed method works well with general camera motions as it can be shown in the estimated latent images and depth maps. Qualtative results are shown in Figure 6.5 and quantitative results are summarized in Table 6.2.

(a) Part of input images centered on the target frame



(b) Ground truth (c) Baseline (d) Lee *et al*. [41] (e) Proposed

Figure 6.3: Results for the Dolls dataset. Some part of the input images are shown in (a) with the target frame red-boxed. In (b)∼(e), the results of each method are presented. For each result, the latent image and estimated depth map are shown from up to down.

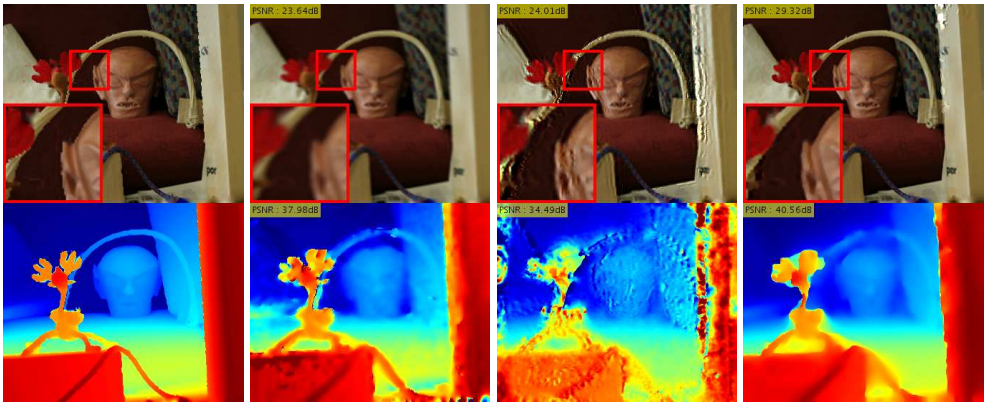
Table 6.2: The performance comparison for the Dolls dataset. The PSNR values are averaged for whole frames in the sequence.

Method	**PSNR**(images)	**PSNR**(depth maps)
Baseline	18.72dB	31.63dB
Lee *et al*. [41]	13.72dB	18.37dB
Proposed	26.18dB	38.76dB

The Reindeer dataset

On the other hand, the motion of camera is almost linear for this dataset. As a result, the method in [41] works better than in the case of the Dolls dataset. Still, the slight rotational motion makes it unreliable. Qualtative results are shown in Figure 6.4 and quantitative results are summarized in Table 6.3.

(a) Part of input images centered on the target frame



(b) Ground truth (c) Baseline (d) Lee *et al*. [41] (e) Proposed

Figure 6.4: Results for the Reindeer dataset. Some part of the input images are shown in (a) with the target frame red-boxed. In (b)∼(e), the results of each method are presented. For each result, the latent image and estimated depth map are shown from up to down.

Table 6.3: The performance comparison for the Reindeer dataset. The PSNR values are averaged for whole frames in the sequence.

Method	**PSNR**(images)	**PSNR**(depth maps)
Baseline	23.76dB	37.41dB
Lee *et al*. [41]	24.78dB	34.47dB
Proposed	30.64dB	39.36dB

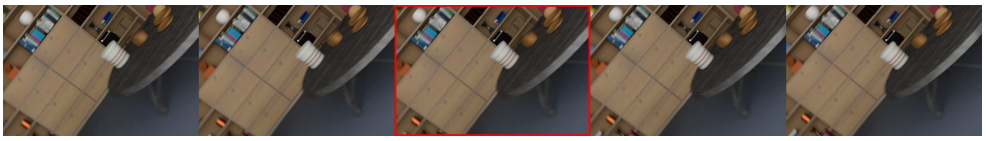
The Mesa dataset

The motion of camera for this dataset is more realistic than the first two datasets. It includes rotation, translation, and forward motion. Also, the baselines of the cameras between frames are relatively small compared to the complexity of the motion. As a result, the method in [41] completely fails for some frames. The proposed method successfully reconstructs the images and depth maps even for this difficult dataset.

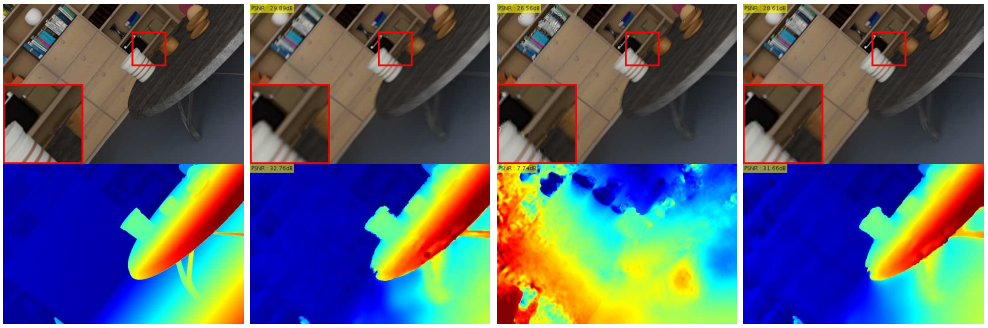
The performance of the proposed method for the Mesa dataset is slightly worse than the baseline in terms of the PSNR. Unfortunately, the blur model used in Blender software [1] is different from the proposed one. It assumes the midpoint of the camera path during the shutter time is the viewpoint for the latent image. That is why the latent images and depth maps of the proposed method is slightly misaligned from the ground truths. Although we compensate this misalignment by warping them into the ground-truth camera poses, these warping operations make additional errors. Also, the blur effect in the synthesized images is not so severe that the baseline itself gives satisfying results. Note that, still, the proposed method gives more visually pleasing results in terms of both images and depth maps.

Qualtative results are shown in Figure 6.5 and quantitative results are summarized

in Table 6.4.



(a) Part of input images centered on the target frame



(b) Ground truth (c) Baseline (d) Lee *et al*. [41] (e) Proposed

Figure 6.5: Results for the Mesa dataset. Some part of the input images are shown in (a) with the target frame red-boxed. In (b)∼(e), the results of each method are presented. For each result, the latent image and estimated depth map are shown from up to down.

Table 6.4: The performance comparison for the Mesa dataset. The PSNR values are averaged for whole frames in the sequence.

Method	**PSNR**(images)	**PSNR**(depth maps)
Baseline	29.96dB	32.87dB
Lee *et al*. [41]	26.15dB	7.99dB
Proposed	28.51dB	31.72dB

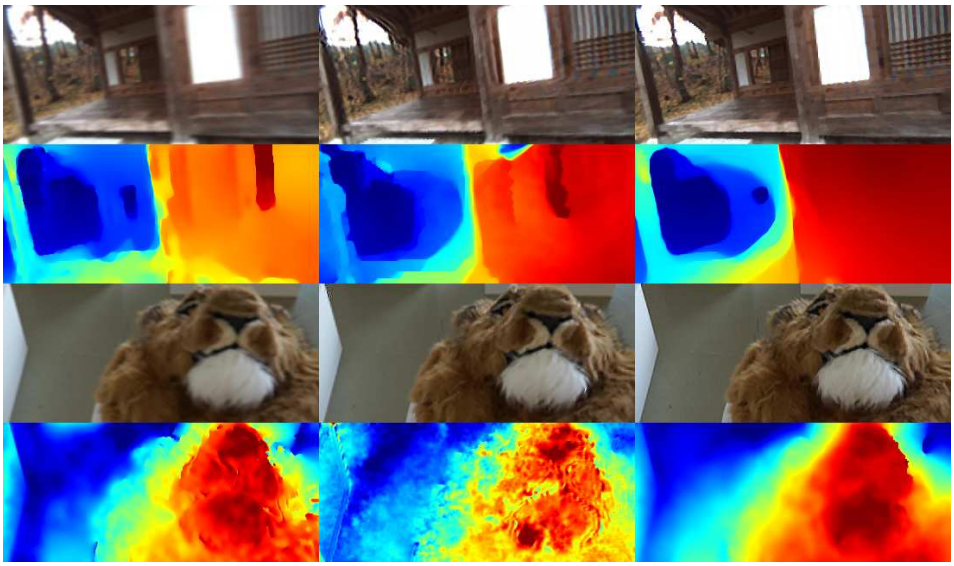


Figure 6.6: The comparison of depth maps and latent images for real datasets. From left to right, the results are from the (a) baseline, (b) method in [41], and (c) proposed method. Each pair of rows shows results for one dataset. The top two rows are from the dataset in [41] (mostly linear camera motions), while the bottom rows are from one of our datasets (with the rotating and forward-moving camera).

6.6.2 Real datasets

We received a dataset, **house**, used in [41] from the authors for comparison. All the other real datasets are made by capturing image sequences with a hand-held camera, Sony Nex6. Note that the exact information about shutter time or FPS is known for these datasets. The results are shown in Figure 6.6. The results of the proposed method are more plausible in terms of both depth maps and latent images.

Figure 6.7 shows the comparison to other image restoration methods [13, 61, 72]. While the images are blurred by real camera motions, we generate LR images by downsampling them manually to compare the super resolution performances. The pro-

Figure 6.7: Comparison of the estimated latent images. From left to right, results are from the (a) bicubic interpolation, (b) use of [61] after [72], (c) use of [13] on original HR images, (d) latent images of the proposed method, and (e) corresponding depth maps.

posed method clearly outperforms the others, even if the results of [13] are obtained using the original HR images.

More specifically, the magnified views of each dataset in Figure 6.7 shows that originally unrecognizable characters become recognizable only in the results of the proposed method. The first row shows that fur-like structures are well reconstructed in deblurred image and the proposed algorithm can reconstruct the depth maps with many depth discontinuities. The third row shows that even when the surfaces have weak texture, the depth of the surfaces can be reliably estimated. This results in less ringing artifacts around the seams of baseball, showing a clearly better deblurring result than the others. The last rows show that the proposed algorithm is scalable to some degree;

it can deal with larger and bigger spaces than the workspaces or desk environments.

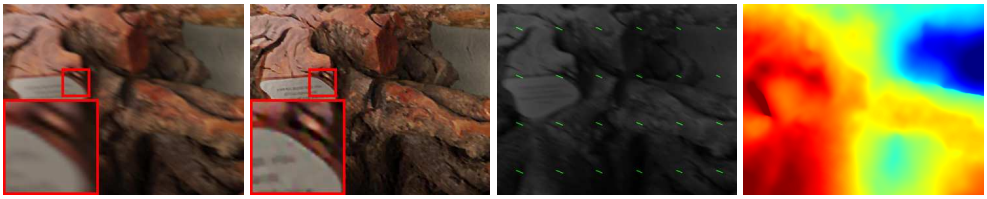
More results on real datasets are shown in Figure 6.8, Figure 6.9, Figure 6.10, and Figure 6.11. In these figures, the estimated blur kernels are also visualized, showing that the estimated blur kernels coincide with the blur patterns in the input images. These real datasets are taken from various environments ranging from an indoor scene with a book shelf (Figure 6.8) to outdoor scene with bench (Figure 6.11), and even including a scene from an art gallery (Figure 6.10). The results indicate that the proposed method is versatile and can be applied to many real-world scenarios.

(a) Part of input images centered on the target frame

(b) Bicubic interpolation (c) Estimated latent image (d) Visualized blur kernels (e) Estimated depth map

Figure 6.8: Additional experimental results on real images. Some part of the input images are shown in (a) and the bicubic interpolation result of the target frame is in (b). The estimated latent image of the proposed method is shown in (c) with the corresponding blur kernels in (d) and the depth maps in (e).

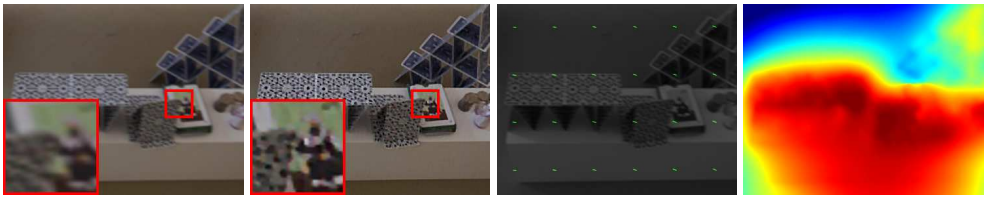
(a) Part of input images centered on the target frame



(b) Bicubic interpolation (c) Estimated latent image (d) Visualized blur kernels (e) Estimated depth map

Figure 6.9: Additional experimental results on real images. The configuration of the subfigures are same as that of Figure 6.7

(a) Part of input images centered on the target frame

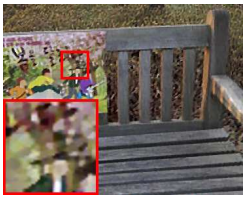


(b) Bicubic interpolation (c) Estimated latent image (d) Visualized blur kernels (e) Estimated depth map

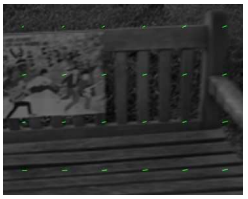
Figure 6.10: Additional experimental results on real images. The configuration of the subfigures are same as that of Figure 6.7

(a) Part of input images centered on the target frame

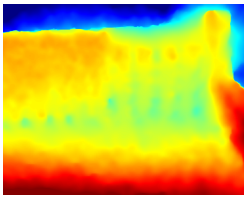
(b) Bicubic interpolation



(c) Estimated latent image



(d) Visualized blur kernels



(e) Estimated depth ma

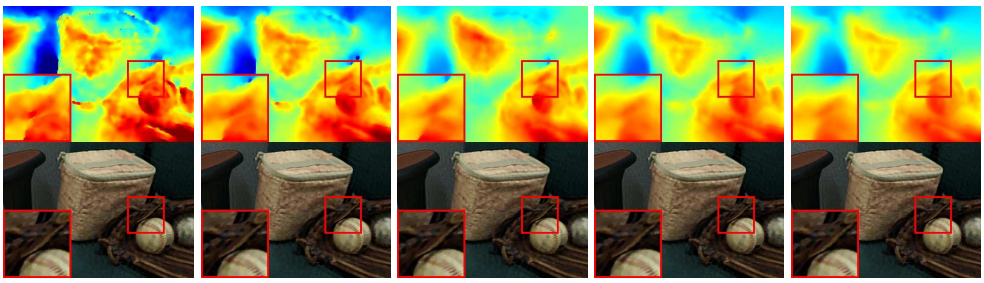
Figure 6.11: Additional experimental results on real images. The configuration of the subfigures are same as that of Figure 6.7

6.6.3 The effect of parameters

The parameters for the proposed method are empirically chosen. They include the number of iterations for optimization, *max_iter*, the degree of regularization in depth maps, λ_d, the degree of regularization in latent images, λ_h. In this section, we show the effect of these parameters on solutions.

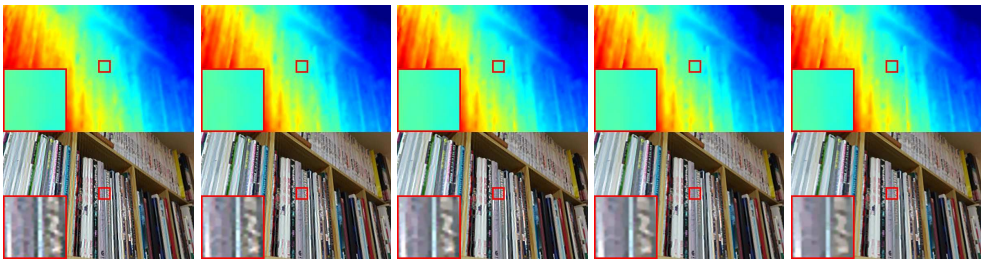
Figure 6.12 shows the effect of λ_d. As expected, the small value of λ_d tends to give noisy depth maps while large λ_d makes the solution overly smooth. The value of λ_d seems to have less effect on the latent images. The chosen value for most dataset is 5. The top row shows the estimated depth maps and the bottom row shows the estimated latent images.

Figure 6.13 shows the effect of λ_h. Again, the value of λ_h seems to have less effect on the depth maps. The chosen value for most dataset is 0.3. The top row shows the



(a) $\lambda_d = 1.25$ (b) $\lambda_d = 2.5$ (c) $\lambda_d = \mathbf{5}$ (d) $\lambda_d = 10$ (e) $\lambda_d = 15$

Figure 6.12: The effect of varying λ_d.



(a) $\lambda_h = 0.1$ (b) $\lambda_h = 0.2$ (c) $\lambda_h = \mathbf{0.3}$ (d) $\lambda_h = 0.4$ (e) $\lambda_h = 0.5$

Figure 6.13: The effect of varying λ_h.

estimated depth maps and the bottom row shows the estimated latent images.

6.7 Conclusion

We proposed a pioneering framework to jointly solve four inter-related computer vision problems: dense depth reconstruction, camera pose estimation, super resolution, and deblurring. The energy function that models these combined problems is derived by revisiting the blurry image formulation, in which more general camera motions and nonlinear blur kernels are allowed than the previous blur-aware matching methods. The experiments show that the proposed method outperforms other related methods

that only deal with one or two of the target problems, in terms of both depth maps and latent images.

Table 6.1: The performance comparison for synthetic datasets. The inputs are upsampled for '**Baseline**', '**Lee *et al.* [39, 41]**', and '**Proposed (w/o SR)**' method. The results of '**Baseline (w/ [72] + [61])**' are obtained by using the images sequentially processed by the methods in [72] and [61]. The depth map errors are measured by using PSNR and relative depth errors (rel.), and the image errors are measured using PSNR. The camera positions are firstly scaled to match the scale of ground truths and then errors are measured in terms of translation error (trans.) and rotation error (rot.) according to the metric in [51]. The translation errors are normalized by using the ground-truth distance between the first two cameras. All the errors are averaged for the whole frames in each sequence.

Datasets	**Methods**	**Depth errors**		**Image errors**	**Pose errors**	
		PSNR(dB)	**rel.**	**PSNR(dB)**	**trans.**	**rot.(°)**
Dolls [28]	Baseline	31.87	0.14	18.72	0.37	0.18
	Baseline (w/ [72] + [61])	29.66	0.17	15.48	5.12	0.25
	Lee *et al.* [39, 41]	26.51	0.34	13.72	3.81	2.62
	Proposed (w/o SR)	34.71	0.11	15.95	0.91	0.35
	Proposed	**39.51**	**0.05**	**27.35**	**0.09**	**0.07**
Reindeer [28]	Baseline	37.63	0.08	23.76	0.61	0.01
	Baseline (w/ [72] + [61])	31.19	0.11	17.31	5.81	0.03
	Lee *et al.* [39, 41]	34.52	0.12	24.78	0.74	4.50
	Proposed (w/o SR)	37.01	0.10	23.69	1.34	0.04
	Proposed	**39.69**	**0.07**	**32.10**	**0.50**	**0.01**
Mesa [2]	Baseline	37.55	0.04	33.38	0.48	0.02
	Baseline (w/ [72] + [61])	25.24	0.37	20.10	16.61	2.34
	Lee *et al.* [39, 41]	34.49	0.08	26.45	6.91	1.24
	Proposed (w/o SR)	37.74	0.04	33.97	**0.17**	**0.02**
	Proposed	**38.50**	**0.04**	**36.47**	0.31	0.03

Chapter 7

A Unified Framework for SLAM

7.1 Motivation

Visual simultaneous localization and mapping (SLAM) is a computer vision problem where the camera pose and 3D scene information is jointly updated from the incoming image sequence. The differences from the structure from motion (SfM) methods include its on-line updating scheme compared to the batch joint optimization. Additionally, the runtime of the algorithm is also an important issue in SLAM since it is commonly used for near real-time applications such as robot navigation or augmented reality.

The accuracy and efficiency of the SLAM algorithms has been improved constantly during the past decades [36, 47, 17, 46, 16]. Still, even the most recent algorithms [46, 16] do not explicitly consider the degradation of the input like motion blur. Since the presence of severe motion blur has influence both on feature detection and matching, conventional method for camera pose estimation or map building scheme would fail for such input data. This prevents the existing algorithms from being versatile and reliable for various real-world scenarios; low-light condition or fast moving

cameras.

There are a few related works [37, 39] to handle blurry image sequence in a SLAM framework. The method proposed in [39] uses blur-aware matching first proposed in [33] to track the feature points in tracking phase. And the deblurring of the incoming frame is done for better feature detection. The limitation of [39] is that the deblurring procedure and the blur aware matching is separated each other. Strictly speaking, the algorithm is blur-robust SLAM and deblurring is performed outside of the main SLAM pipeline. In terms of SLAM framework, it is based on sparse feature point detection and tracking, which is not robust to the presence of large weakly textured areas.

In this chapter, we propose a SLAM framework, in which the deblurring of key frame images, camera pose update, and map update are jointly estimated. It also uses blur-aware matching similar to the one used in [39] during the tracking phase, but instead of double blurring both the key frame image and new image, we only blur the key frame image. This saves the amount of computation, while giving better localization due to less amount of blur in textures.

It is important to minimize the increase of computational computation due to runtime issue. For this reason, we select the recent SLAM method proposed in [16]. It uses direct image registration based SLAM formulation, which we can apply the generalized imaging process equations in Section 3.4 directly, while minimizing the computational complexity by sparsely sampling the scene points.

7.2 Baseline

In this section, we briefly summarize the pipeline of the baseline method. It is based on a two-track framework like the one firstly proposed in [36]. One thread process every

newly obtained frame using a simple visual odometry to track the pose of the moving camera while the other thread process joint estimation of map and poses of key frames using a more complicated energy optimization. The main differences compared to the conventional SLAM methods are that 1) it uses direct image registration for visual odometry and joint optimization instead of feature point detection and matching and 2) the image registration is performed only for sparsely sampled pixel points rather than entire pixels. The advantage of this approach is that it can utilize the accuracy of direct image registration without much increasing the computational complexity compared to the feature-based SLAM.

The important equations for this SLAM system are the ones used for the energy formulation for joint optimization. Following to the notations in this dissertation, it can be represented as follows:

$$E_{photo} = \sum_{i \in F} \sum_{\mathbf{p} \in P_i} \sum_{j \in \text{obs}(\mathbf{p})} E_{\mathbf{p}j}, \tag{7.1}$$

where F is the set of all activated key frames and P_i is the set of sparsely sampled pixel points for the frame i. The set obs $(\mathbf{p})$ is a set of frames where $\mathbf{p}$ is visible, and the energy for individual pixel matching $E_{\mathbf{p}j}$ is defined as follows:

$$E_{\mathbf{p}j} = \sum_{\mathbf{p} \in N_{\mathbf{p}}} w_{\mathbf{p}} \left\| \left(I_j \left(\mathbf{p}' \right) - b_j \right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i \left(\mathbf{p} \right) - b_i \right) \right\|_{\gamma}. \tag{7.2}$$

Note that this energy is almost same as the one used for the method proposed in Chapter 6, except that the two global brightness between two frames are calibrated by introducing additional parameters a_i, b_i, and t_i, and the Huber norm is used instead of magnitude of L2 norm. The weighting value $w_{\mathbf{p}}$ is used for compensation of small geometric errors. The pixel position $\mathbf{p}'$ is actually computed by using the warping equation, which is the function of the depth and camera poses, as in Equation (3.5). In this

dissertation, we only remark the most important equations which are to be modified in the proposed method. A more comprehensive explanation regarding the baseline algorithm can be found in the original paper [16].

7.3 Proposed Method

We follow the pipeline of the baseline method for SLAM with two main modifications.

First, every key frame image is deblurred by using the camera poses and depth information when it is newly registered. We can deblur the entire key-frame image by interpolating the sparsely sampled depth map to make a dense depth map and, thus, dense pixel-wise blur kernels. However, it is inefficient since only the image patches around the sparsely sampled key points are used for joint optimization and tracking. Thus, we deblur the patches around the sparsely sampled key points independently, where the size of each patch is determined the corresponding blur kernel. We represent a k-th key point in a key frame i as $\mathbf{x}_i^k$ and a set of pixel coordinates around it as $P\left(\mathbf{x}_i^k\right)$. The optimization equation for each patch is defined as follows:

$$\sum_{\mathbf{x}\in P\left(\mathbf{x}_i^k\right)} \left\| B_i^k\left(\mathbf{x}\right) - \Psi_i^k \circ I_i^k\left(\mathbf{x}\right) \right\|_2 + \lambda_h \sum_{\mathbf{x}\in P\left(\mathbf{x}_i^k\right)} \left\| \nabla I_i^k\left(\mathbf{x}\right) \right\|_2, \tag{7.3}$$

where the input blurred image patch around $\mathbf{x}_i^k$ and the corresponding output deblurred image patch are represented as B_i^k and I_i^k, respectively. The blur operator Ψ_i^k in the above equation acts as a convolution kernel where the blur coefficient is computed on $\mathbf{x}_i^k$ and set to be the same for all pixels in the patch.

After all patches are deblurred, the deblurred image I_i is updated by copying the pixel values of the deblurred patches. Then by using the deblurred key frame images in the joint estimation of Equation (7.1), we can perform the blur-robust key frame refinement.

Second, we modify the registration energy term in Equation (7.2) during the tracking phase to make capable of handling blurry images. Given that the most recent key frame image I_i is already deblurred, and we want to align the newly obtained image B_j to it, the equation becomes as follows:

$$E_{\mathbf{p}j} = \sum_{\mathbf{p} \in N_{\mathbf{p}}} w_{\mathbf{p}} \left\| \left(B_j\left(\mathbf{p}'\right) - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(\left(\Psi_j \circ I_i\right)(\mathbf{p}) - b_i\right) \right\|_\gamma . \tag{7.4}$$

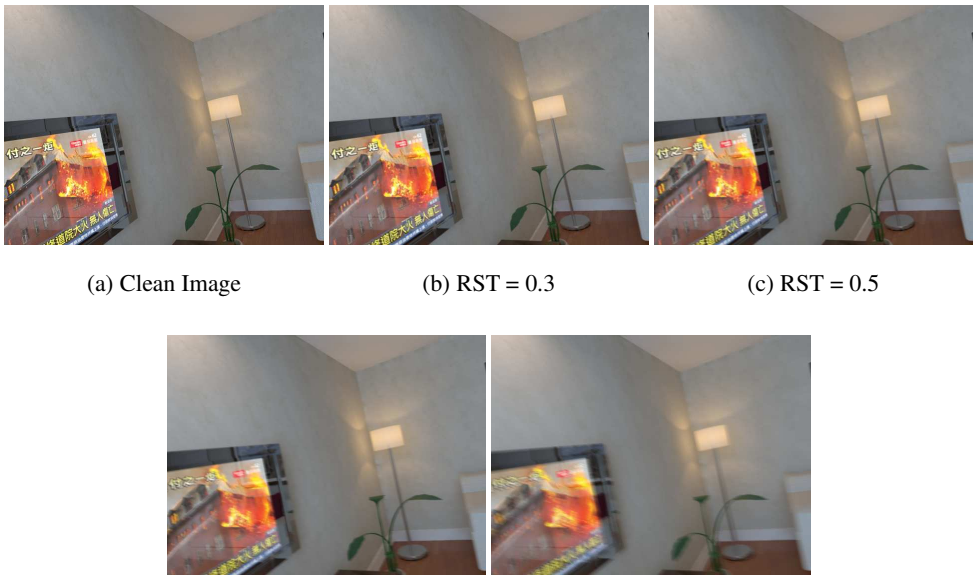
7.4 Experimental Results

In this section, we compare the performance of the proposed method to the baseline to verify its effectiveness under the presence of motion blur in input images.

7.4.1 Quantitative comparison

Although there are publc datasets [25, 8, 18] to evaluate the monocular SLAM algorihtms, the explicit blur handling is not considered in these dataset. While the sequences in the **EuRoC MAV dataset** [8] contain some blurry frames, the dataset does not provide the shutter time information necessary for the proposed method. The **TUM monoVO dataset** [18] provides frame-wise shutter time information along with frame rates, but the degree of motion blur in this dataset is too weak. The **ICL-NUIM dataset** [25] is synthetic dataset and the images in this dataset are ideally clean.

To test the performance of the proposed method, we first synthesized the blurry image sequences based on the **ICL-NUIM dataset**, using the approache in Section 4.1. Note that this is only possible for **ICL-NUIM dataset** since the ground-truth depth maps and camera poses are fully known only for synthetic dataset. To adjust the degree of blur, we have defined a term relative shutter time (RST). The RST is simply multiplication of the actual shutter time and the framerate. Thus, if RST is equal to 1



(a) Clean Image (b) RST = 0.3 (c) RST = 0.5

(d) RST = 1.0 (e) RST = 2.0

Figure 7.1: Visualization of synthesized motion blur images with varying RST values. The **lr kt0** sequence of **ICL-NUIM dataset** is used.

then, the imaginary intermediate frames between two frames fully contribute to synthesize a blurry image. Figure 7.1 visualizes the synthesized blurry images with varying RSTs.

The performance of the proposed method is compared to the baseline using the blurry **lr kt0** sequences with varying RSTs. The methods are evaluated in terms of absolute trajectory error (e_{ate}), and the results are summarized in the Table 7.1. The errors are measured by averaging the individual errors of five independent forward runs for each sequence and method pair, considering the randomness of the algorithms. Since the motion between consecutive frames is very small for this dataset, the proposed method is ineffective for the sequences with small RST values. Still, the presence of motion blur indeed decrease the performance of the baseline method, and when the

Table 7.1: The performance comparison of the baseline and proposed method on the synthesized blurry image sequences with varying RST values. The image sequences are synthesized by using the **lr kt0** sequence of **ICL-NUIM dataset**.

RST	**baseline**	**proposed**
0.3	0.0016	0.0016
0.5	0.0016	0.0019
1.0	0.0029	0.0029
2.0	0.0074	0.0044

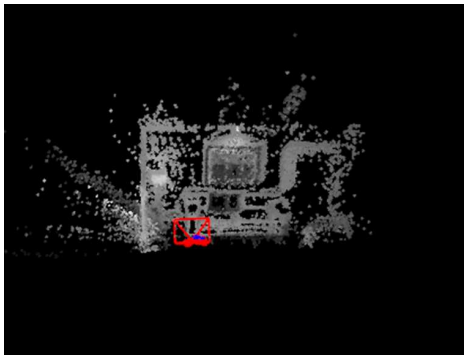
blur is severe (RST = 2.0), the proposed method outperforms the baseline method as expected. The reconstructed 3D points maps of each method in Figure 7.2 indicate that the motion blur caused noise in 3D points of the baseline due to the matching ambiguity, while the result of proposed method shows a cleaner 3D map.

Theoretically, whether the RST of a sequence is small or large, the proposed method estimates the blur kernels with proper sizes and the performance should not be lower than the baseline. However, the table shows that when the degree of blur is small, the performance of the proposed method is worse than the baseline. The main causes are thought to be; i) The use of TV prior will smooth out the details when the size of the blur kernels are small, ii) patch-wise deblurring makes unsatisfactory artifacts due to boundary pixels, iii) when camera pose tracking has severe error, it will propagate to the subsequent frames and have more effect than in the case of baseline since deblurring and blur-aware matching both are affected by the error.

We also compared the performance using the **TUM monoVO dataset** [18]. From the sequences in the **TUM monoVO dataset**, we only selected the sequences that

(a) An example of input image

(b) Baseline



(c) Proposed

Figure 7.2: Visualization of reconstructed 3D points maps of the (b) baseline and (c) proposed method for synthesized blurry image sequence with RST = 2.0 (the **lr kt0** sequence of **ICL-NUIM dataset**). Considering that the most structure should be rectangular as shown in (a), the reconstruction result of the baseline contains more noise than that of the proposed method, due to failures of matching key points.

have maximum shutter time larger than 30ms. Still the most of the frames have very short (3ms) shutter time and contain no motion blur, making the proposed method less effective. Table 7.1 summarizes the comparison results for each sequence. Note that we only can measure loop error for this dataset, and again we use average error of five

Table 7.2: The performance comparison for the **TUM monoVO dataset**. The errors are averaged alignment error for five forward runs (e_{align}) [18].

Sequence number	baseline	proposed
13	0.4591	0.4256
14	0.2221	0.1916
15	0.2521	0.2882
26	0.5239	0.4261
28	0.2529	0.1990
35	0.1280	0.1690
36	0.4320	0.4546
37	0.2055	0.1800

forward runs using the alignment error (e_{align}) as proposed in [18].

7.4.2 Qualitative results

We have tested the proposed SLAM system on three more real image sequences, **cafe**, **statue**, and **flowers**. All the datasets are made by using a hand-held camera, Sony Nex6 and the exact information about shutter time or FPS is known. The camera motions are not restricted to be translational and the images are severely blurred as shown in Figure 7.3. The number of frames of image sequences are different for each dataset, ranging from 150 to 200.

The performance of the proposed system is compared to the baseline qualitatively. Figure 7.4, Figure 7.5, and Figure 7.6 shows the reconstructed camera paths along with sparse 3D maps for each dataset. The 3D maps are denser in the results of baseline

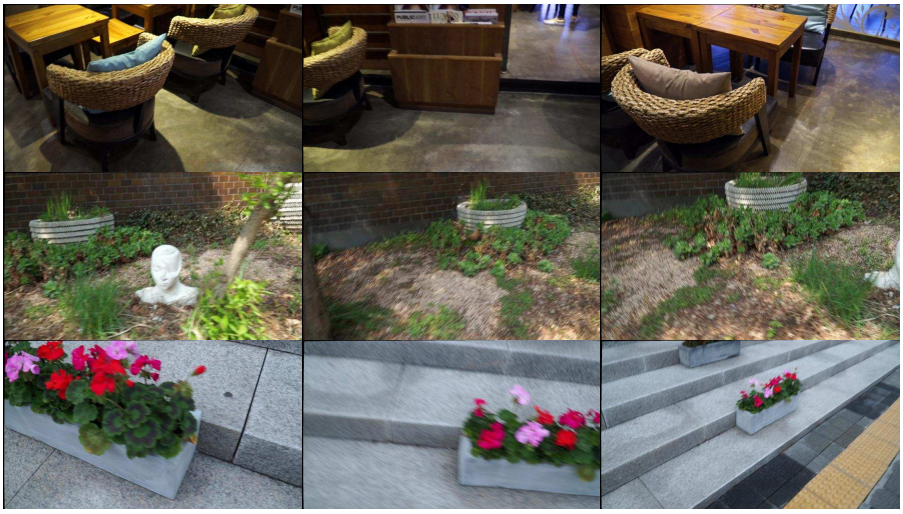


Figure 7.3: Sample images from each data sequence are shown. From top to row, the images are from **cafe**, **statue**, and **flowers** sequence.

system. This is because the proposed system treats each patch around the key point as if it is fronto-parallel to the imaging plane during the deblurring procedure, which causes inaccurate deblurring results with undesired artifacts. Especially when there are large depth discontinuities inside a patch, the reconstructed image patch becomes too inaccurate and the corresponding 3D key point is classified outlier in the following optimization process. The result of deblurring is shown in Figure 7.7.

Although the number of points in the reconstructed 3D map of the proposed system is less, the proposed system shows more accurate camera paths with much less noise in the reconstructed sparse 3D maps. This is because the presence of severe motion blur makes the matching result ambiguous, resulting in much more noise in the result of baseline system. This can be seen from the Figure 7.4, Figure 7.5, and Figure 7.6. For example, the stair structure in the **flowers** dataset should look almost like a thin line when the viewing ray is aligned to be on one of the stair edges. The result of the

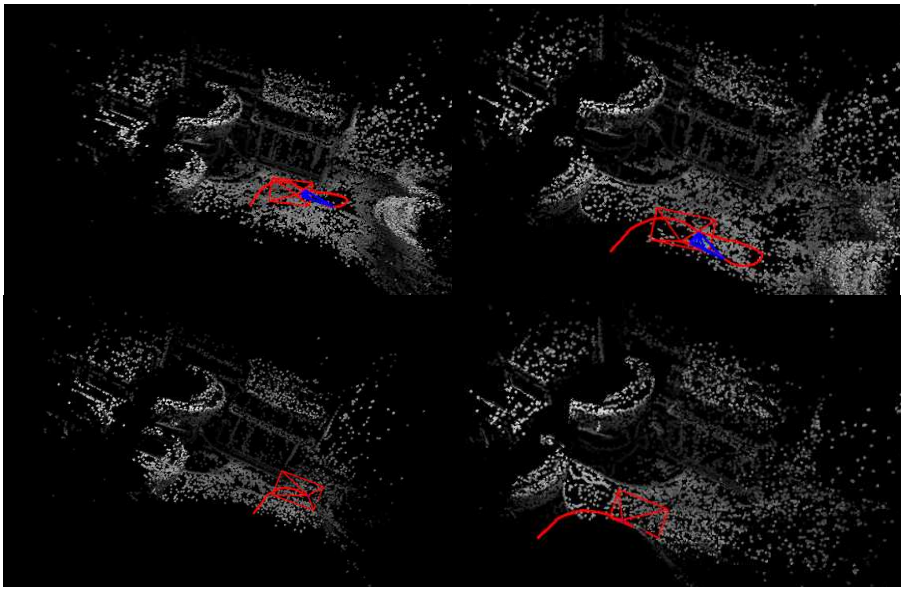


Figure 7.4: Reconstruction results of the baseline in the top row and reconstruction results of the proposed system in the bottom row. Each column shows the scene structure from a different viewpoint. The results are for **cafe** dataset.

baseline system, however, shows a much thicker line than that of the proposed system due to the noise in the reconstruction. As a result, the reconstructed camera path is more natural in the result of the proposed system. Also, in the **cafe** dataset the chairs appearing from right are severely blurred that makes the camera motion estimation of the baseline system inaccurate. The motion for the last few frames seems purel rotational, but the results from baseline system show translational movement.

7.4.3 Runtime

The runtime of the proposed system become much slower than the baseline system due to the computational complexity of deblurring. Under the computing environment with Intel i5-6600K (3.50GHz), the runtime of the proposed system drops to 0.05$\sim$0.1

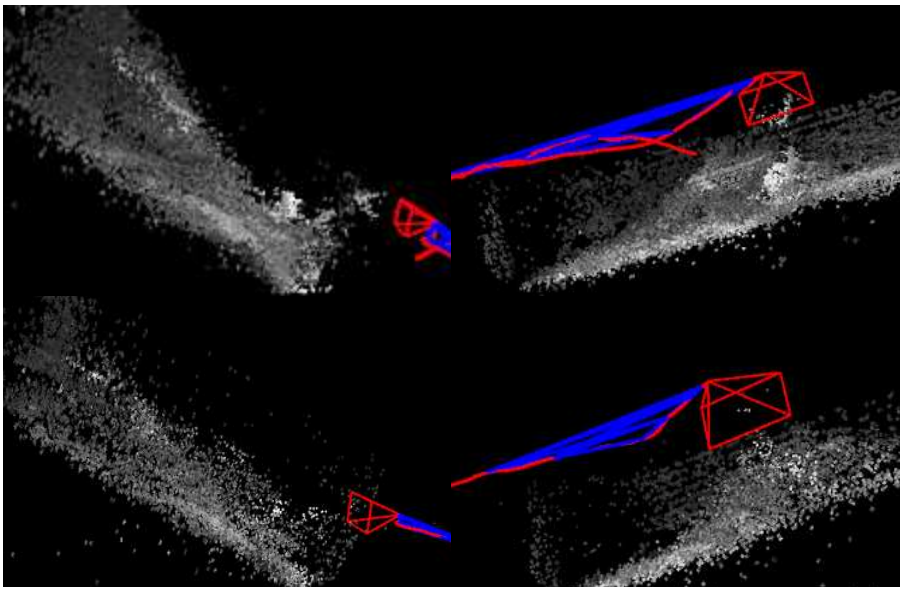


Figure 7.5: Reconstruction results of the baseline in the top row and reconstruction results of the proposed system in the bottom row. Each column shows the scene structure from a different viewpoint. The results are for **statue** dataset.

fps where the runtime of the baseline system is 7$\sim$8 fps, where the image resolution is 768×432. Still, it is much more faster than the batch system in Chapter 6, and it has scalability to handle a long sequence. We need further algorithm optimization to apply this system in realtime SLAM system.

7.5 Conclusion

The accuracy of conventional SLAM algorithms drops when the input image sequence contains severe motion blur. We extend the use of joint estimation framework in Chapter 6 to solve SLAM problem, where the image deblurring, camera pose estimation, and depth estimation are solved in a unified SLAM framework. The experimental results show that the proposed system is more reliable than the baseline system for the

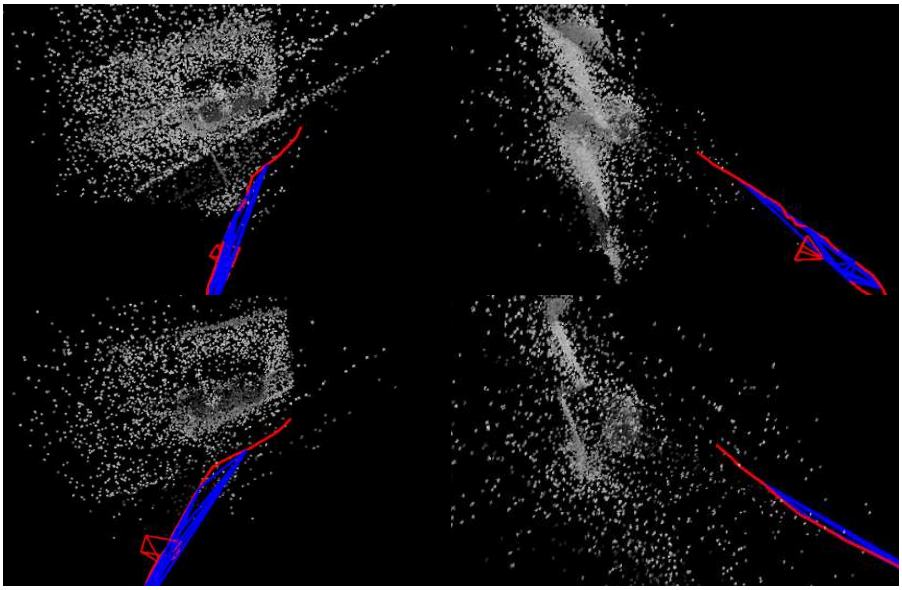


Figure 7.6: Reconstruction results of the baseline in the top row and reconstruction results of the proposed system in the bottom row. Each column shows the scene structure from a different viewpoint. The results are for **flowers** dataset.

image sequences with severe motion blur. Still, when the degree of motion blur is small the performance of the proposed method is often worse than the baseline due to poor deblurring algorithm used in the pipeline. The improvement of deblurring algorithm and reducing the computational complexity of the entire pipeline would be the remaining works.

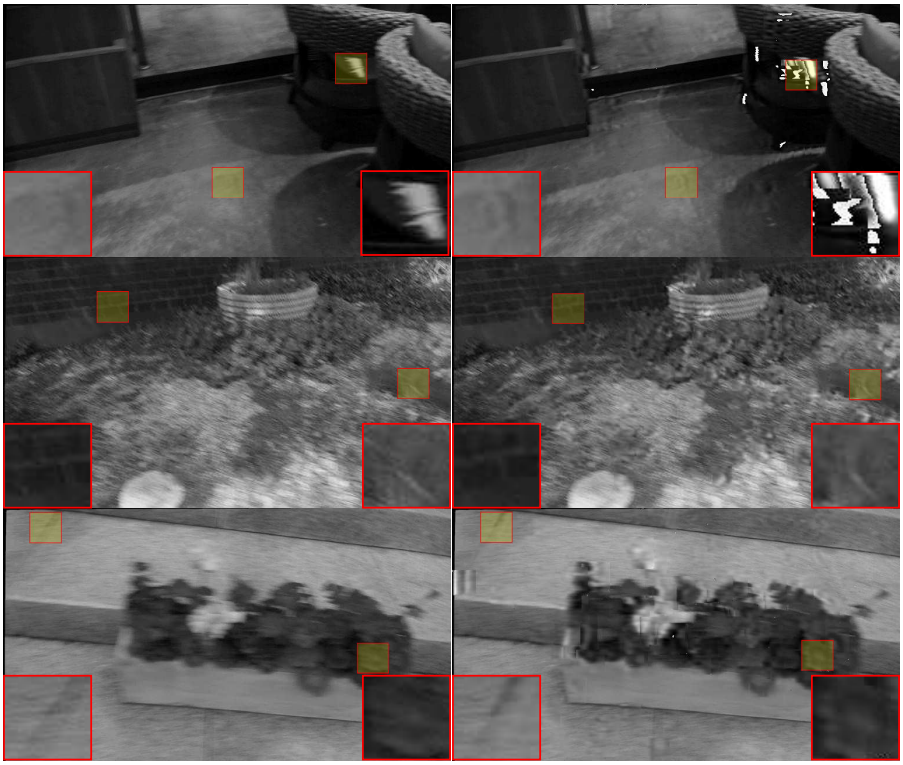


Figure 7.7: Deblurring results are visualized for each dataset. From top to row, the images are from **cafe**, **statue**, and **flowers** sequence. From left to right, the original images and the corresponding deblurred images are shown. The red boxes with yellow masks are magnified in the bottom corners of each image. Note that the right red box of the **cafe** dataset shows the failure case where the presence of disparity discontinuities made some undesired visual artifacts.

Chapter 8

Conclusion

8.1 Summary and Contribution of the Dissertation

Solving image reconstruction, such as deblurring and super resolution, is one of the fundamental problems of computer vision research. While one of the main goals is in reconstructing visually pleasing images from the degraded input, it is also important to get physically meaningful reconstruction results that can be used for other computer vision problem. Especially when the interested application is multi-view stereo or SLAM, the accuracy and consistency of reconstruction should be in sub-pixel units. Unfortunately, reconstructing a latent image from single input image is inherently inverse problem, and using the conventional single-view approaches as preprocessing fails worse than the use of original degraded images.

On the other hand, the dense, pixel-wise correspondence and multi-view geometry constraints among multiple images can give a leverage to solve image reconstruction problem. While it is commonly known for super resolution problem, as the multi-frame super resolution frameworks are based on image registration, this fact is not fully utilized in deblurring methods.

In this thesis, we analyze the image capturing process and approximate it using the equations based on multi-view geometry theories. We applied this assumption to jointly solve the multi-view stereo problems and image reconstruction problems for three different scenarios; deblurring and depth estimation for one-shot multi-view images, batch joint estimation of super-resolved and deblurred latent images with corresponding camera poses and depth maps from a single-view image sequence, and online update of camera poses and depth maps from blurry image sequence in a SLAM environment. Experimental results show that the application of the proposed methods outperforms the conventional baseline methods where each problem is solved separately. We suggest that solving the two most fundamental computer vision problems, multi-view stereo and image reconstruction, jointly in a unified problem is effective and practical.

8.2 Future Works

We have proposed three different methods that can be selectively used for different circumstances. While the first two batch methods are targeting off-line application, still the computational complexity makes these method less practically useful. This is mainly due to the optimization process that involves many images or depth maps in one equation, which results in an ill-conditioned and large linear system at each iteration. Developing a better optimization technique which fully utilizes the divide-and-conquer strategy would be one way to make the systems more applicable. Designing a dedicated approximated linear solver or an initialization scheme for faster convergence would also be possible directions.

Applying deep learning to the systems whether partially or fully would also be

interesting. All the proposed methods are based on conventional energy minimization framework where the energies are derived by using multi-view geometry constraints and commonly used TV-prior. Reflecting the fact that the methods based on deep learning approaches are the most successful for each single problem, The application of deep learning would be promising way to improve the performance of the joint estimation systems, providing many benefits, not even for accuracy but also for computational efficiency.

Bibliography

[1] http://www.blender.org/.

[2] http://www.blendswap.com/blends/view/72340/.

[3] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[4] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009.

[5] A. Bhavsar and A. Rajagopalan. Resolution enhancement in multi-image stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1721–1728, September 2010.

[6] J.-L. Blanco. A tutorial on se(3) transformation parameterizations and on-manifold optimization. Technical report, University of Malaga, 2010.

[7] Y. Bok, H.-G. Jeon, and I. S. Kweon. Geometric calibration of micro-lens-based light-field cameras using line features. In *Proceedings of the European Conference on Computer Vision*. Springer, 2014.

[8] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[9] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision*, 2012.

[10] P. Chandramouli, P. Favaro, and D. Perrone. Motion deblurring for plenoptic images. *arXiv preprint arXiv:1408.3686*, 2014.

[11] C. Chen, H. Lin, Z. Yu, S. Bing Kang, and J. Yu. Light field stereo matching using bilateral statistics of surface cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[12] S. Cho and S. Lee. Fast motion deblurring. 28(5):145, 2009.

[13] S. Cho, J. Wang, and S. Lee. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):64, 2012.

[14] D. G. Dansereau, A. Eriksson, and J. Leitner. Motion deblurring for light fields. *arXiv preprint arXiv:1606.04308*, 2016.

[15] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[16] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. In *arXiv:1607.02565*, July 2016.

[17] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision*, September 2014.

[18] J. Engel, V. Usenko, and D. Cremers. A photometrically calibrated benchmark for monocular visual odometry. *arXiv preprint arXiv:1607.02555*, 2016.

[19] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, October 2004.

[20] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, December 2015.

[21] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[22] J. Fiss, B. Curless, and R. Szeliski. Light field layer matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[23] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proceedings of the European Conference on Computer Vision*, 2010.

[24] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Hong Kong, China, May 2014.

[25] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, 2014.

[26] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[27] Y. S. Heo, K. M. Lee, and S. U. Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[28] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[29] Z. Hu, L. Xu, and M.-H. Yang. Joint depth estimation and camera shake removal from single blurry image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014.

[30] Z. Hu and M.-H. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In *Proceedings of the British Machine Vision Conference*, 2012.

[31] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. S. Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015.

[32] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2005.

[33] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2005.

[34] M. Jin, P. Chandramouli, and P. Favaro. Bilayer blind deconvolution with the light field camera. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015.

[35] T. H. Kim and K. M. Lee. Segmentation-free dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014.

[36] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2007.

[37] G. Klein and D. Murray. Improving the agility of keyframe-based slam. In *Proceedings of the European Conference on Computer Vision*. Springer, 2008.

[38] J. Kwon and K. M. Lee. Monocular slam with locally planar landmarks via geometric rao-blackwellized particle filtering on lie groups. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[39] H. S. Lee, J. Kwon, and K. M. Lee. Simultaneous localization, mapping and deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, November 2011.

[40] H. S. Lee and K. M. Lee. Multiswarm particle filter for vision based slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[41] H. S. Lee and K. M. Lee. Dense 3d reconstruction from severely blurred images using a single moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2013.

[42] H. S. Lee and K. M. Lee. Simultaneous super-resolution of depth and images using a single camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2013.

[43] H. Lin, C. Chen, S. Bing Kang, and J. Yu. Depth recovery from light field using focal stack symmetry. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[44] K. Maeno, H. Nagahara, A. Shimada, and R.-i. Taniguchi. Light field distortion feature for transparent object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[45] C. Mei and I. Reid. Modeling and generating complex motion blur for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.

[46] R. Mur-Artal, J. Montiel, and J. D. Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[47] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[48] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[49] T. Portz, L. Zhang, and H. Jiang. Optical flow in the presence of spatially-varying motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2012.

[50] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the limits of stereo using variational stereo estimation. In *IEEE Intelligent Vehicles Symposium*, June 2012.

[51] V. Rodehorst, M. Heinrichs, and O. Hellwich. Evaluation of relative pose estimation methods for multi-camera setups. *International Archives of Photogrammetry and Remote Sensing*, pages 135–140, 2008.

[52] J. A. Scales and A. Gersztenkorn. Robust methods in inverse theory. *Inverse problems*, 4(4):1071, 1988.

[53] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[54] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *Proceedings of the SIGGRAPH Conference*, New York, NY, USA, 2006.

[55] A. Snoswell and S. Singh. Light field de-blurring for robotics applications. In *Australasian Conference on Robotics and Automation*, 2014.

[56] H. Strasdat, J. Montiel, and A. J. Davison. Real-time monocular slam: Why filter? In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, 2010.

[57] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2010.

[58] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015.

[59] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[60] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015.

[61] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *IEEE Asian Conference on Computer Vision*, 2014.

[62] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[63] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):889–901, May 2012.

[64] T.-C. Wang, A. A. Efros, and R. Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[65] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4d light fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012.

[66] A. Wedel, T. Pock, C. Zach, D. Cremers, and H. Bischof. An improved algorithm for TV-L1 optical flow. In *Proc. of the Dagstuhl Motion Workshop*. Springer, September 2008.

[67] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International journal of computer vision*, 98(2):168–186, 2012.

[68] W. Williem and I. Kyu Park. Robust light field depth estimation for noisy scene with occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[69] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of the European Conference on Computer Vision*. Springer, 2010.

[70] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of the European Conference on Computer Vision*, Berlin, Heidelberg, 2010.

[71] L. Xu and J. Jia. Depth-aware motion deblurring. In *Computational Photography (ICCP), 2012 IEEE International Conference on*. IEEE, 2012.

[72] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2013.

[73] Y. Xu, H. Nagahara, A. Shimada, and R.-i. Taniguchi. Transcut: Transparent object segmentation from a light-field image. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2015.

[74] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the IEEE International Conference on Computer Vision*, 1999.

초 록

저품질의 영상들로부터 장면의 3D 구조나 카메라 포즈를 추정하는 것은 어려운 문제이다. 대부분의 기존 다시점 스테레오 알고리즘들은 고화질, 고품질의 입력 영상을 가정하고 있어 블러, 저해상도 등으로 인한 화질 저화가 존재하는 입력에 대해서는 안정적으로 동작하지 않는다. 본 논문에 제시된 실험들은 기존의 영상 복원 기법을 독립적으로 적용한 후, 기존의 다시점 스테레오를 수행하는 접근이 대개의 경우 효과적이지 않으며 때로 오히려 더 좋지 않은 성능으로 이어진다는 것을 보여준다. 이는 단순하게 각 프레임별로 독립적인 영상 복원 기법을 적용하는 것이 시각적으로 보기에 좋은 결과를 낼 수는 있으나, 다시점 스테레오 입장에서는 오히려 서로 연관성을 갖는 전체 영상들 사이의 일관성을 해치기 때문이다.

본 학위 논문에서는 서로 연관된 문제인 영상 복원과 다시점 스테레오를 하나의 통합된 시스템으로 공동으로 해결하는 기법들을 다룬다. 이러한 가정의 유효성은 여러 연관된 문제들을 동시에 해결하는 새로운 기법들을 제안, 구현하고 실험적으로 검증함으로써 확인된다. 연관된 문제들은 전체 영상에 대해 세밀한 깊이 지도 복원, 카메라 포즈 추정, 초해상도, 그리고 디블러링으로, 하나의 움직이는 카메라로 촬영된 연속된 영상들 혹은 여러 개의 카메라로 동시에 촬영된 한 프레임의 다시점 영상들을 대상으로 한다. 제안된 기법들은 물리적인 영상 취득 과정을 추정하고자 하는 변수들로 모델화하여, 해당 문제들을 모두 다룰 수 있는 하나의 비용 함수를

정의하고 이를 반복적인 최적화 기법으로 해결한다. 합성 영상 및 실제 영상을 대상으로 한 다양한 실험 영상들을 이용한 실험에서는, 기존의 다시점 스테레오 기법들이 실패하는 저품질 영상들에 대해서도 제안된 기법들은 고품질의 깊이 지도 복원 및 카메라 포즈 추정이 가능하다는 것을 보여준다. 또한, 제안된 기법들의 영상 복원 결과는 기존의 단일 영상 디블러링, 초해상도 기법, 혹은 기존의 동영상 디블러링, 초해상도 기법들이나 그 조합에 비해 시각적으로 더 훌륭한 결과를 보여준다.

본 학위 논문의 의의는 전체적인 관점에서 기존의 컴퓨터 비젼 문제들을 조망함으로써 각 문제들을 해결하는데 있어 새로운 관점을 제시하고 있다는 것이다. 특히, 저품질의 입력 영상들에 대해 다시즘 스테레오 및 영상 복원을 수행함에 있어 연관된 문제들을 동시에 하나의 시스템으로 해결하는 것이 더 나은 시각적 결과 뿐만 아니라, 물리적으로 보다 설득력 있는 결과를 얻을 수 있음을 보였다. 제안된 최적화 알고리즘은 계산 복잡도 측면에서 제안된 기법들을 보다 실용적으로 만들어줄 수 있다.

주요어: 다시점 스테레오, 영상 복원, 디블러링, 초해상도, 공동 추정

학번: 2011-30234