Ph.D DISSERTATION

Simultaneous 3D Reconstruction, Deblurring, and Super-resolution using a Single Moving Camera

움직이는 단일 카메라를 이용한 3차원 복원과
디블러링, 초해상도 복원의 동시적 수행 기법

BY

HEE SEOK LEE

August 2013

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

Abstract

Vision-based 3D reconstruction is one of the fundamental problems in computer vision, and it has been researched intensively significantly in the last decades. In particular, 3D reconstruction using a single camera, which has a wide range of applications such as autonomous robot navigation and augmented reality, shows great possibilities in its reconstruction accuracy, scale of reconstruction coverage, and computational efficiency. However, until recently, the performances of most algorithms have been tested only with carefully recorded, high quality input sequences. In practical situations, input images for 3D reconstruction can be severely distorted due to various factors such as pixel noise and motion blur, and the resolution of images may not be high enough to achieve accurate camera localization and scene reconstruction results. Although various high-performance image enhancement methods have been proposed in many studies, the high computational costs of those methods prevent applying them to the 3D reconstruction systems where the real-time capability is an important issue.

In this dissertation, novel single camera-based 3D reconstruction methods that are combined with image enhancement methods is studied to improve the accuracy and reliability of 3D reconstruction. To this end, two critical image degradations, motion blur and low image resolution, are addressed for both sparse reconstruction and dense 3D reconstruction systems, and novel integrated enhancement methods

for those degradations are presented. Using the relationship between the observed images and 3D geometry of the camera and scenes, the image formation process including image degradations is modeled by the camera and scene geometry. Then, by taking the image degradation factors in consideration, accurate 3D reconstruction then is achieved. Furthermore, the information required for image enhancement, such as blur kernels for deblurring and pixel correspondences for super-resolution, is simultaneously obtained while reconstructing 3D scene, and this makes the image enhancement much simpler and faster. The proposed methods have an advantage that the results of 3D reconstruction and image enhancement are improved by each other with the simultaneous solution of these problems. Experimental evaluations demonstrate the effectiveness of the proposed 3D reconstruction and image enhancement methods.

Keywords: Vision-based 3D reconstruction, Visual SLAM, Image enhancement, Image deblurring, Image super-resolution.

Student Number: 2006-21271

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 3D Reconstruction using a single camera

Understanding the three dimensional (3D) structure of a scene and an object has an important role in the human visual perception system as well as the machine vision system. Visual tasks of human, such as navigation, recognition of object and people, are greatly aided by 3D visual information. Thus, people prefer digital contents made by 3D videos or graphics rather than 2D original images, and producing 3D digital contents receives great attention in the last decades. To generate 3D contents from the real world, methods for acquiring and analyzing 3D geometry information from images or videos are crucial.

In Augmented Reality (AR), which overlays related objects or contents of a scene onto images, 3D geometry information composed of a structure of environment and a camera pose is very useful to improve the reality of AR systems. For example, overlaid objects in the AR view should change its appearance according to camera view changes. Only estimation of both 3D scene structure and camera pose can realize

such AR system. 3D geometry estimation is also important for a self-localization of autonomous robots or vehicles. A mobile robot that moves automatically should use a Simultaneous Localization and Mapping (SLAM) system which estimate a map of environment and a trajectory of mobile robot simultaneously, and estimating 3D geometry is a key part of the SLAM system.

Obtaining 3D geometry can be achieved by various sensors. In particular, obtaining 3D geometry from images is referred to as *image-based 3D reconstruction*, and it is a fundamental problem in computer vision research. Various types of image-based 3D reconstruction approaches, such as reconstruction using stereopsis, multiple static camera, and single moving camera, have been developed for the last decades. In particular, the single camera-based reconstruction is widely applied to systems for AR and autonomous robot systems where the cameras for reconstruction move dynamically.

The single camera-based 3D reconstruction has two primary objectives. The first one is to generate a 3D structural model of environment or objects as a form of sparse point cloud, dense depth map, or surface using mesh model. The second objective is to estimate a pose of camera which is generally represented by 3D translation with 3D rotation. Given that the two objectives have to be solved simultaneously, the problem does not have a closed form solution, and thus it is difficult to solve. Two different approaches have been developed to solve this problem: filtering-based approach [8–11] and optimization-based approach [12–14]. Filtering-based 3D reconstruction, which uses Extended Kalman Filter (EKF) or Particle Filter (PF) to simultaneously estimate a map and a camera motion, were initially studied in SLAM literature for autonomous mobile robots, and then it is applied to vision-based reconstruction systems where camera is used as an observation sensor. On the other

hand, optimization-based 3D reconstruction, which solves the geometry problems of scene and camera using optimization method such as bundle adjustment [15], was addressed by computer vision researchers to reconstruct a model of object, and then it is extended to the large scale reconstruction for a navigation purpose.

In the last decade, 3D Reconstruction using a single camera has been received much attention and studied extensively both in robotics research and computer vision research. For the intelligent robotics, 3D Reconstruction with a single moving camera which is often referred as visual SLAM, plays a key role for the self-localization of mobile robots. Visual SLAM has many advantages over other range sensor-based SLAM such as SLAM using laser scanner or sonar. A camera used in visual SLAM is much smaller and cheaper than other range sensors, and it can provide more information on surrounding environment such as colors and textures. One drawback of visual SLAM is relatively higher computational cost than other SLAM systems due to its bearing-only property, but with the development of both algorithms and computation hardware, great improvement has been conducted in the visual SLAM performances, and attempts to utilize visual SLAM for practical applications are now begun by the industries. Augmented reality is another representative application of 3D Reconstruction using a single camera. With the widespread of portable computing devices equipped with a camera, such as smart phones, tablet PCs, and wearable PCs, augmented reality shows its potentials for various applications.

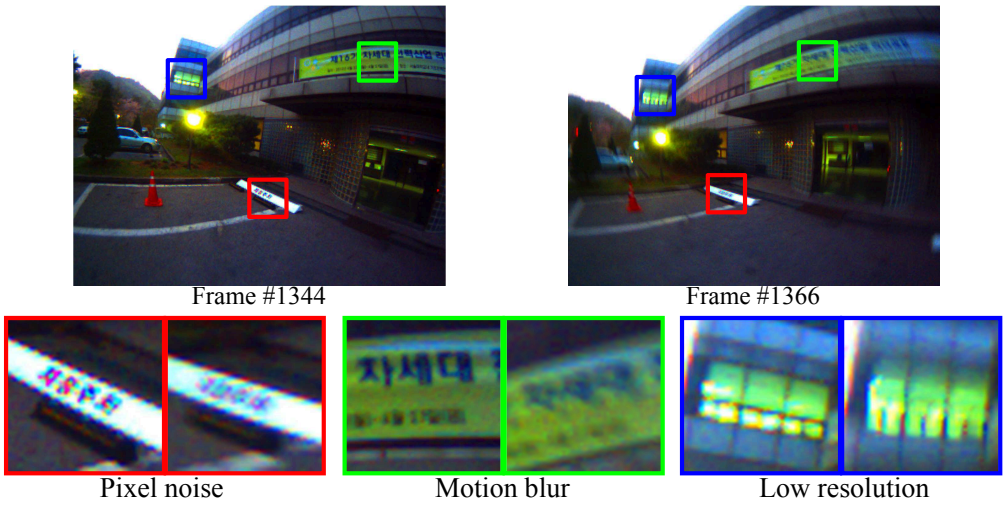


Figure 1.1: Various image quality degradations that cause difficulties in correspondence between two images for 3D reconstruction.

1.2 Image Enhancement for 3D Reconstruction

1.2.1 Image quality problem in 3D reconstruction

Although 3D Reconstruction using a single camera shows its great feasibility for challenging scenarios such as large scale operation [10,13] and frequent camera pose drift [9,16] which should be addressed for practical applications, still there are many remaining practical issues on this method. One of the most important issues is various distortions of input images which make it difficult to match pixels or features across the input images that is essential step for the scene point reconstruction and the camera localization. In laboratory experiments, high-performance cameras which can provide high-resolution and high frame rate images are used and the cameras are carefully controlled by skilled researcher to capture proper input images. However, in a practical situation, the image quality can be degraded by various fac-

tors including pixel noise, camera motion blur, and resolution decrement. Figure 1.1 shows examples of those three image degradations which make matching problems for 3D reconstruction difficult. Therefore, methods for securing the image quality are essential to utilize 3D Reconstruction using a single camera in practical purposes.

There have been various image enhancement algorithms for removing those image degradation factors. However, most algorithms solve the image enhancement problem independently from the geometry of camera and scene, instead they rely on the input images only and use a prior knowledge on image properties. Recent image-only-based image enhancement algorithms can recover high quality images without any geometric information, but they require high computational cost to solve the problem. Therefore, those methods are inadequate to 3D reconstruction using a single camera where the real-time capability is important.

Few studies have been conducted on the image enhancement methods specialized for improving the single camera-based 3D reconstruction. Most 3D reconstruction methods robust to image quality degradation focus on handling of degradation factors in a reconstruction step instead of enhancing the input images. For example, [17] and [16] presented blur-robust methods for 3D reconstruction. In [17], the point spread function (PSF) is estimated for segmented image regions, and the estimated PSF is utilized to minimize an undesired effect of the motion blur in extracting interest points and building image descriptors. Although this method does not deblur input images, the computation time is not adequate for real-time operation. [16] tried to solve blurring effect in visual SLAM using detected edge features. They utilized the blur-invariance of edge features to obtain correspondences between images. In this method, however, edge features are not registered to a map when the motion blur exists. It can be a problem when motion blur continues for many

frames in unmapped region, since no point or edge for localization will be available in the map. On the other hand, if we recover the input images in advance of the reconstruction step, then mapping scene point can be always performed and motion blur can be handled more effectively. The drawback of explicit image enhancement is high computational time, but it can be solved by utilizing geometric information from 3D reconstruction process.

For the resolution problem of 3D reconstruction, the relationship between image super-resolution and 3D scene structure is studied in several works [18–20]. In [18], the super-resolution is formulated with the calibrated 3D geometry and solved using MAP-MRF framework. Occlusions are effectively handled in their super resolution method using depth information, but super-resolution does not contribute to depth map estimation in this method. In [20], a method for increasing the accuracy of 3D video reconstruction is present. The 3D video is composed of texture images and 3D shapes, and increasing their accuracy is achieved by simultaneous super-resolution using MRF formulation and its optimization. This work has differences with the proposed reconstruction-combined method in that this work uses multiple static cameras to reconstruct moving object, and does not perform full frame super-resolution. Recently, the authors of [19] formulate a full frame super resolution problem combined with a depth map estimation problem, and attempt to enhancing results of both problems. However, their solution is not fully simultaneous but follows EM-style method, *i.e.*, they fix the current high resolution image and estimate the depth map, and vice versa. For each iteration, MRF optimization is applied to depth estimation, and iterated conditional modes (ICM) is used for image estimation, thus the computation cost is inevitably large in this method. More related works on 3D reconstruction and image enhancement will be discussed in the later

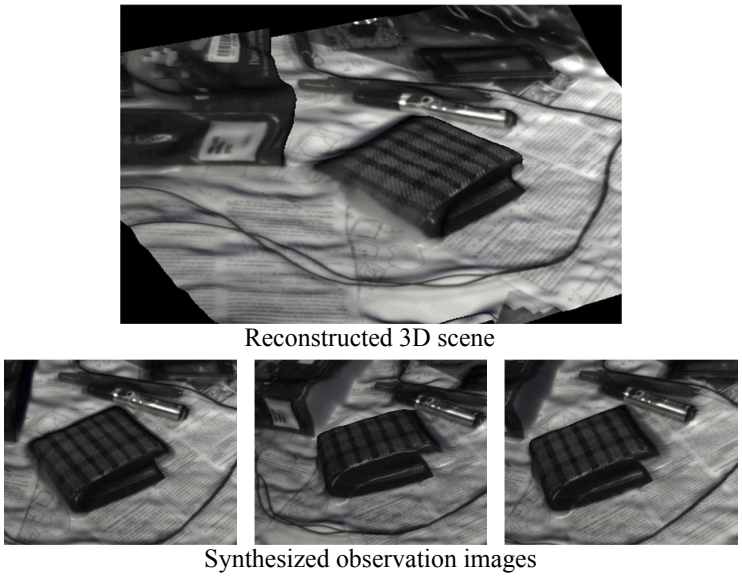


Figure 1.2: Example of 3D reconstruction of a scene and its synthesized observation images for different virtual viewpoints.

sections.

1.2.2 Proposed approach: Simultaneous 3D Reconstruction and Image Enhancement

The image formation process can be interpreted as a camera projection from a scene point to an image plane. If we have a geometry information composed of 3D scene structure and camera motion, then we can model the image formation process with the geometry information. In other words, given with a 3D structure of scene and a camera motion as shown in Figure 1.2, the observed images for any camera pose can be synthesized. The geometric information can be effectively utilized for image enhancement, for example, a blurred appearance of a pixel can be predicted from the camera motion and 3D position of the pixel. Many image enhancement

problem is ill-posed, but ambiguities of image enhancement can be resolved by the geometry information. Reversely, enhancing the quality of input image enables enhancing the result of 3D reconstruction. The key of accurate 3D reconstruction is to found correspondences of pixels or feature points across input images as accurate as possible. It is obvious that applying image enhancement to input images of 3D reconstruction improves the reconstruction qualities.

The results of 3D reconstruction and image enhancement can improve each other since the two problems are closely correlated. However, solving two problems separately takes much computational cost and may produce worse results than the original results of each algorithm. Therefore, the simultaneous solutions for two problems are addressed in this dissertation. By utilizing the dependency of images and 3D geometry, these two problems are combined and solved by a single optimization framework. As a result, the computational cost for image enhancement can be saved by using of geometric information from 3D Reconstruction. Furthermore, we can overcome the limitation of other geometry-free algorithms, *i.e.*, image-only-based algorithms, and can expect better enhancement result than those algorithms. These are the main motivations of this study, and this approach is referred as *geometric image enhancement*. In particular, two types of image enhancement, motion deblurring and super-resolution are addressed in this dissertation.

1.3 Dissertation Goal and Contributions

The 3D reconstruction system developed for this study is equipped with a single camera and controlled by human hand, *i.e.*, the camera has 6 degrees of freedom (DOF) motion. Two types of single camera-based 3D reconstruction approach that have

different purposes are addressed. The first type is sparse point-based reconstruction, and the second type is dense depth reconstruction. The sparse point-based reconstruction requires a relatively small computational cost than the dense reconstruction because only selected feature points are reconstructed with its 3D positions. On the other hand, dense reconstruction that estimates depth values of all pixels in an image requires more computation cost than the sparse point-based reconstruction, but it provides more complete reconstruction results. Thus, the sparse point-based reconstruction is regarded as localization oriented, whereas dense reconstruction is regarded as reconstruction oriented. The goal of the dissertation is to apply motion deblurring and image super-resolution to both reconstruction methods. The contribution of the dissertation is summarized as follows:

- *A 3D reconstruction method robust to image degradation* is proposed. Combining the deblurring method with 3D reconstruction using a single camera allows fast camera motion, which should be addressed for a practical use especially under low light condition, by robustly performing a data association for a blurred image. The super-resolution improves the quality of 3D reconstruction with low-resolution images and helps robust data association for severe scale changes of observed scene.

- *An efficient image enhancement* is proposed. The information required for image enhancement can be easily obtained from 3D reconstruction. Blur kernels for deblurring and pixel correspondences for super-resolution are directly related to camera motion and scene depth, thus those can be simultaneously estimated with the estimation of 3D geometry. This geometry-aware image enhancement has advantages in the computational speed and the robustness

to severe scene structure variation.

- *An analysis of the relationship between geometry and image enhancement* is presented. There are few studies on the 3D geometries for the image enhancement. In this dissertation the theoretical and experimental analysis of 3D geometries in the image deblurring and super-resolution is discussed. Their relationship can be utilized not only in the proposed single camera-based 3D reconstruction, but also in other sensor-based 3D reconstruction systems where 3D geometry is estimated.

1.4 Organization of Dissertation

The main body of this dissertation is composed of five chapters. The first two chapters deal with the methods for sparse point-based 3D reconstruction, combined with deblurring and image super-resolution, respectively. Chapter 4 and 5 address the dense reconstruction method combined with deblurring and image super-resolution, respectively, and chapter 6 presents unified deblurring and image super-resolution method for dense reconstruction. Table 5.1 summarizes the chapter organization of the dissertation.

Table 1.1: Chapter organization for the dissertation.

	Deblurring	Super-resolution
Sparse point-based reconstruction	Chapter 2	Chapter 3
Dense reconstruction	Chapter 4	Chapter 5
	Chapter 6	

- *Chapter 2 and 3:* Among two 3D reconstruction approaches for sparse point-based reconstruction, optimization-based reconstruction method is combined with image deblurring and filtering-based reconstruction is combined with image super-resolution because each reconstruction approach has its own advantages in each image enhancement method. More 3D scene points are reconstructed in the optimization-based method compared with the filtering-based method, and then we can obtain more blur kernels for non-uniform deblurring. On the other hand, the proposed image super-resolution is achieved via Kalman filter, thus we can effectively combined the super-resolution method with 3D reconstruction in a single filtering framework.

  The motion deblurring method for sparse point-based reconstruction is studied in Chapter 2. The proposed algorithm achieves fast blur kernel estimation using camera geometry. The blur kernel is modeled by a trajectory of projected pixel in image during exposure time, and this trajectory can be easily calculated from the the reconstructed 3D scene point and the camera motion and exposure time. Image super-resolution for sparse point-based reconstruction using the Rao-Blackwellized particle filter-based formulation [21] is then studied in Chapter 3. During the update of 3D landmark poses and a camera pose through the particle filter iteration [22], a high resolution template of each landmark is also updated by Kalman filter simultaneously.

- Chapter 4, 5, and 6: The basic idea of image enhancement applied to sparse point-based reconstruction is extended to the dense depth reconstruction algorithms in Chapter 4, 5, and 6. The dense reconstruction, which is extended from the sparse point-based reconstruction, also suffers from various image

distortions, and thus image enhancement is needed. The goal is to estimate depth of all pixels in an image as well as their deblurred and high resolution pixel values.

Chapter 4 presents the deblurring combined reconstruction, where a blur kernel for each pixel is parameterized with its depth and solved by variational method [23–25] using camera motion. In Chapter 5, the super-resolution combined reconstruction is presented. The image correspondences for input image sequence is parameterized with pixel depth and also solved by variational method using camera motion. Since the proposed approaches are not an alternating method but a simultaneous method of depth estimation and image enhancement, it does not require huge amount of computational cost. The final goal of this study is to incorporate those two enhancement methods in a unified framework, and this is discussed in Chapter 6.

Chapter 2

Sparse 3D Reconstruction and Image Deblurring

2.1 Introduction

To make a more practical 3D reconstruction system for mobile robots or augmented reality, handling rapid camera motion is very essential problem. If the camera motion is fast and unpredictable, then observed landmark positions in image also change fast and unpredictably, and sometimes the appearances of landmarks can change, which caused by *motion blur*.

Motion blur is usually regarded as an undesired phenomenon in recoding images or videos. Especially in SLAM, where a camera keeps moving by human hands or autonomous robots, failure of localization or reconstruction is often caused by severe motion blur. The motion blur makes it difficult to perform data association for reconstructed landmarks, as well as reconstruction of new landmarks for detected features. We can reduce motion blur by recoding images with high frame rate,

equivalently shorten the exposure time, but enough exposure time is inevitable under low light condition and motion blur occurs.

Many recent vision-based SLAM systems can handle localization failures caused by motion blur by applying the relocalization (global localization) algorithms [9, 26] after localization failure. However, when the camera explores through a region that the camera has not visited and reconstruction has not been done, the relocalization becomes useless since no landmarks to be matched is available in that region. Therefore, the motion blur in unmapped region can be handled only if the system can continuously run the normal map reconstruction processes, including data association as well as mapping new landmark under motion blur.

In sparse point-based reconstruction, many tasks on images are performed with detected point features, such as registering a new feature as a landmark, or finding matching features for reconstructed landmarks. General point feature detectors used in sparse point-based reconstruction, however, cannot give enough features from a blurred image. With a blurred image, moreover, feature matching between frames is difficult and the accuracy of matching decreases. To solve these problems, deblurring an image can improve the performance of visual SLAM by giving enough interest points detected and images that are easy to match. High-quality methods to remove the motion blur have been developed in recent decades [2, 3, 27, 28], but most require a large computational budget. Thus it is hard to use those methods to recover images for image-based reconstruction.

In this chapter, the sparse point-based reconstruction algorithm, especially visual SLAM framework combined with fast image deblurring is proposed. By considering motion blur, data association of visual SLAM can be enhanced, and camera localization can be performed robustly even a scene is blurred. The information

obtained from visual SLAM are used to estimate a motion blur kernels, then the estimated kernels are used in deblurring input image. With the restored image, it is possible to extract more *good features to track* and register them as new landmarks, which is difficult with a blurred image. As a result, localization and mapping can be performed successfully under motion blur.

2.2 Related Work

Although motion blur is an important factor for the visual SLAM performance, there have been few studies on the methods for handling motion blur. In [17], the point spread function (PSF) is estimated for a number of segmented image regions, and the estimated PSF is used to minimize an undesired effect of the motion blur in both extracting interest points and building image descriptors based on SIFT [29]. Although their method does not require explicit deblurring, the computation time is not adequate for real-time operation (one second per frame). They pointed out that deblurring based on deconvolution might worsen the image quality, and is not an adequate solution for handling motion blur in visual SLAM, because the quality of the restored image strongly depends on the accuracy of the estimated PSF. In the proposed approaches, however, we can see that small errors in blur kernel estimation due to image measurement noise are acceptable in the proposed deblurring method.

In [16], the motion blur problem in visual SLAM is solved by using *edgelets*. Edgelet means "a very short, locally straight segment of what may be a longer, possibly curved, line". Their observation is that the edgelet may remain intact even in a heavily blurred image if the directions of edge and motion blur are parallel. Motivated by this observation, they presented a tracking method using edgelets,

and made their visual SLAM system to be robust to motion blur. However, this edge-preserving property can not be applied to motion blur by in-plane camera rotation. In [16], edgelets are not registered to a map while the motion blur exists. It can be a problem when motion blur continues for many frames in unmapped region, since no point feature or edgelet for localization will be available in the map, while the proposed explicit deblurring enables continuous mapping of landmarks in unmapped region.

For the 3D reconstruction purpose, there have been some studies on relationship between motion blur and scene depth. In [30] and [31], the dependency of scene depth on the blur kernel estimation is pointed out, and simultaneous estimation of scene depth and blur kernel is proposed. When a camera motion is pure translation and parallel to the image plane, the magnitude of blur kernel is inverse proportional to the scene depth, and a pixel-wise blur kernel can be parameterized by depth and camera motion. In the proposed method, a full camera calibration is available from visual SLAM, thus the blur kernel estimation for arbitrary camera motion including rotation is possible.

2.3 Motion Blur and 3D Geometry

In this section, the motion blur and its relationship with the camera motion and 3D structure of the scene is discussed. In an ideal case where the exposure time is infinitesimal, the image projection is a one-to-one function between 3D real point and 2D pixel position. In real cases, however, lights from 3D point are projected to the image plane as a 'line' according to a motion of camera or object during the camera exposure. This results in a blurred image. Thus, to capture a sharp image,

(a) Negligible camera motion (b) Fast camera motion with frame rate 30Hz, 15Hz, and 7.5Hz, respectively.

Figure 2.1: Motion blurs and detected Harris corner points [1] with different frame rates. (a) Negligible camera motion with frame rate of 15Hz for a comparison. (b) Fast camera motion with frame rate of 30Hz, 15Hz, and 7.5Hz, respectively. As motion blur becomes severer by decreasing frame rate, the number of detected corner points rapidly decreases.

at least one of following two conditions should be satisfied: exposure time is almost infinitesimal, or no camera motion and no object motion exists in the scene. Motion blur is generated if both conditions are violated.

2.3.1 Motion blur in visual SLAM

We can reduce motion blur in image by using a high frame rate camera. However, if the light source is not enough, then exposure time of the camera should increase to obtain images for SLAM and motion blur is inevitably generated. Figure 2.1 shows the motion blurs with different frame rate (exposure time) and extracted corner points using same feature extractor and same parameters. The results in Figure 2.1 show that although enough frame rate is secured, motion blur occurs inevitavely and image quality is degraded. In [32], studies on the influence of motion blur in feature detection and tracking were conducted, and it shows that most feature detectors and descriptors have severe performance degradation under motion blur. Since most feature detector algorithms rely on the cornerness or edgeness responses based on the

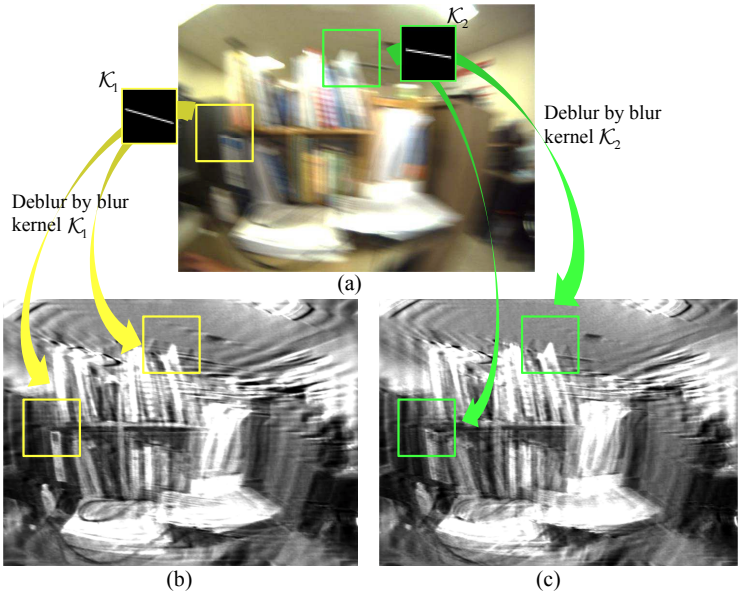


Figure 2.2: Estimated blur kernels at different 3D positions and deblurred regions by estimated kernels. (a) Original blurred image and kernels from two 3D points, $\mathcal{K}_1$ and $\mathcal{K}_2$. (b, c) Deblurring results by kernels $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively. The kernel from other point gives poor deblurring result.

pixel intensities or derivatives, the number of detected features rapidly decreases as the blur effect increases. In feature matching, motion blur changes feature appearance severely or removes high frequency component of feature appearance, which results in low matching performance.

2.3.2 Motion deblurring

Blurred image I_b can be modeled by convolution of blur kernel $\mathcal{K}$ and sharp image I_s, as $I_b = I_s * \mathcal{K}$, where $*$ is convolution operator. Deblurring is inverse process of blurring, thus we can remove this blur by estimating a blur kernel and recover the image by deconvolution. Once the blur kernel is accurately estimated, conventional

deconvolution algorithms such as Weiner filter or Lucy Richardson filter [33] can recover the image with satisfactory quality. Therefore, the most important problem is to estimate an accurate blur kernel. A simple but efficient approach to estimate a blur kernel is to assume a spatially uniform kernel for the entire image [2,27]. However, this assumption is valid only if the scene has a planar structure and the camera has no rotational motion. When a blur is non-uniform, we have to estimate blur kernels for divided image parts and deblur each part, but it is very computationally expensive.

To handle non-uniform blur, some methods for a single image deblurring based on image properties (e.g., α-channel [34], transparency [35]) are proposed, but those methods can be applied only if foreground object and background scene can be distinguished. On the other hand, using the 3D geometry information such as camera motion and scene structure can improve the accuracy and efficiency of deblurring. Some studies have been performed on deblurring by considering the camera motion for a single image deblurring [3,28]. However, they do not deal with full (6-D) camera motion; they simplify the camera motion as three degrees of freedom (DOF). This can be a problem when the objects are not sufficiently distant from the camera. Also, the 3D structure of the scene is not considered in [3, 28], while the depth of scene point is highly correlated to blur kernel.

The dependency of pixel motion which is closely related to blur kernel to the 3D structure of the scene is clearly noted in the well known homography equation, $\mathbf{H} = \mathbf{K}\left(\mathbf{R} - \frac{\mathbf{T}\mathbf{n}^\top}{d}\right)\mathbf{K}^{-1}$. The matrix $\mathbf{R}$ is the camera rotation matrix, $\mathbf{T}$ is the camera translation vector, $\mathbf{n}$ and d are the normal vector from the camera to the 3D plane and the distance to the plane respectively. In Figure 2.2, blur kernels are estimated at different 3D landmark positions and the image is deblurred using the

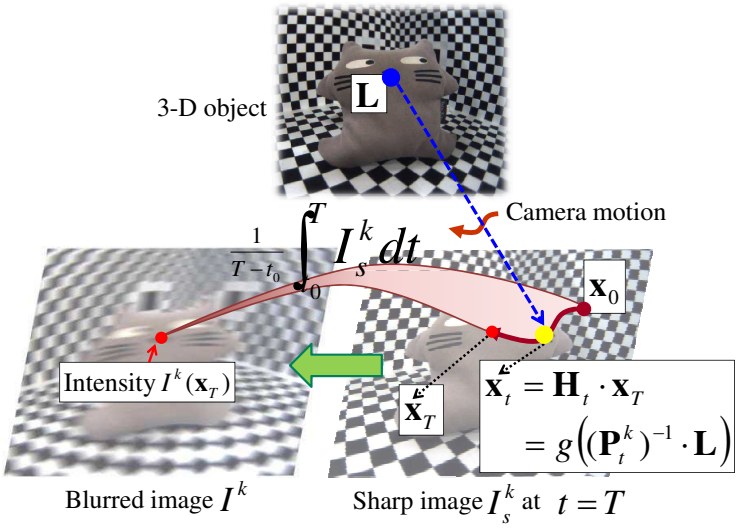


Figure 2.3: Movement of the projected point by camera motion.

estimated kernel. Although there is no moving object and rotation of camera with z-axis, the blur is non-uniform and a different kernel at different landmark gives the wrong deblurring result. In visual SLAM, a camera motion and 3D point structure of the scene are continuously estimated, then we can easily calculate the blur kernel for each individual scene point using those estimates and get good deblur results.

2.3.3 Motion blur and 3D geometry

If the exposure time of camera is not infinitesimal and the camera moves fast, then lights from an object are not projected to one image point. Rather, they make a 'line', and motion blur is generated. Figure 2.3 illustrates the projection process of a 3D point by a moving camera. A superscript k is used for a frame index, and a subscript t is used for time in capturing one image. Let L and $\mathbf{x}_t$ be a 3D scene position and its projected point in image, respectively, with homogeneous representation. During exposure time $t \in [t_0, T]$, the projection $\mathbf{x}_t$ moves from

the initial position $\mathbf{x}_{t_0}$ to the final position $\mathbf{x}_T$, making a trajectory on the image. The movements of pixels in an image can be represented by homography $\mathbf{H}_t$, as $\mathbf{x}_t = \mathbf{H}_t \cdot \mathbf{x}_T$. Since homography is non-uniform for general non-planar scene, $\mathbf{H}_t$ is represented as a function of pixel position $\mathbf{x}_T$. In the image I^k at the frame index k, the intensity of pixel $\mathbf{x}_T$ can be represented as

$$\begin{aligned} I^k(\mathbf{x}_T) &= \int_{t_0}^{T} \Delta I^k(\mathbf{x}_T, t) dt \\ &= \int_{t_0}^{T} \frac{1}{T - t_0} I_s^k(\mathbf{H}_t \cdot \mathbf{x}_T) dt, \end{aligned} \tag{2.1}$$

where $\Delta I^k(\mathbf{x}_T, t)$ is a pixel intensity generated in an infinitesimal time dt, and I_s^k is an intensity of the sharp image at $t = T$.

The relationship between the 3D scene point L and its projected point is given by the equation $\mathbf{x}_t = g((\mathbf{P}_t^k)^{-1} \cdot L)$, where $\mathbf{P}_t^k$ is the camera pose at frame index k defined on the Special Euclidean group $SE(3)$, which represents the rigid transformation of camera composed of 3D translation and 3D rotation [36]. The function $g(\cdot)$ is a perspective camera projection function with the camera intrinsic parameters. Then, we can rewrite Equation (2.1) using the 3D geometry as

$$\begin{aligned} I^k(\mathbf{x}_T) &= \int_{t_0}^{T} \frac{1}{T - t_0} I_s^k(\mathbf{H}_t \cdot \mathbf{x}_T) dt \\ &= \int_{t_0}^{T} \frac{1}{T - t_0} I_s^k(g((\mathbf{P}_t^k)^{-1} \cdot L)) dt. \end{aligned} \tag{2.2}$$

In the motion deblurring algorithm based on the convolution model, the blur kernel $\mathcal{K}$ is inferred from a sequence of homography $\mathbf{H}$, then $\mathcal{K}$ is used to deconvolve the blurred image. In a general situation of blur, the move of a pixel, equivalently $\mathbf{H}$ or $\mathcal{K}$, is not given (called blind deconvolution), and the problem is highly ill-posed. To solve blind deconvolution, complicated methods based on the regularization such as natural image statistics are used to estimate both the blur kernel $\mathcal{K}$ and the sharp

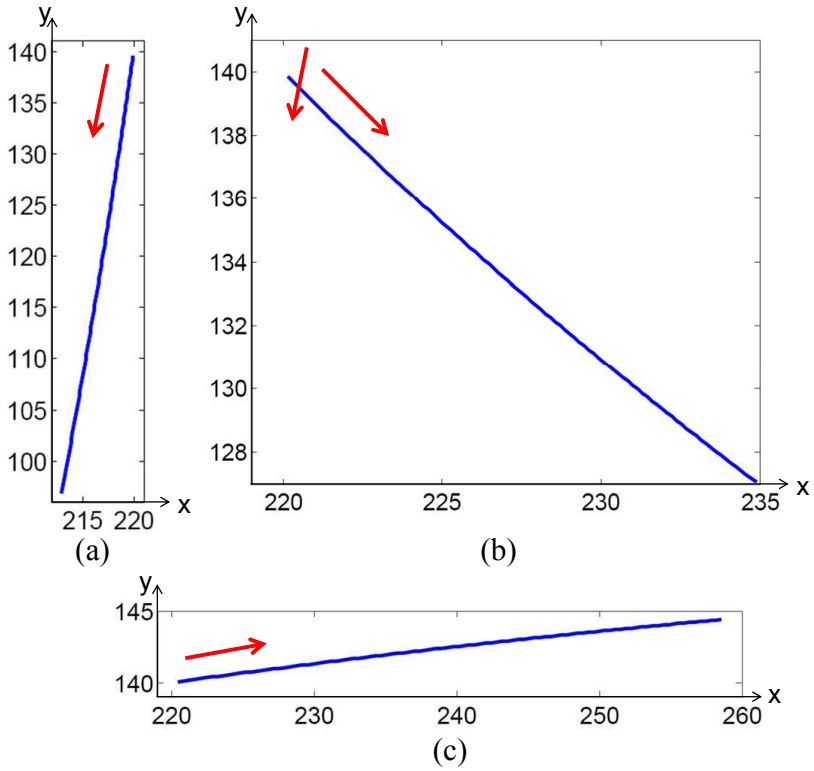


Figure 2.4: Trajectories of projected point by camera rotation with different axes. (a) x-axis rotation (pitching). (b) z-axis rotation (rolling). (c) y-axis rotation (yawing).

image I_s. On the other hand, if we have the estimates of $\mathbf{P}_t$ and L, then we can easily make the kernel $\mathcal{K}$, and the problem becomes the non-blind deconvolution, which is simpler and faster to solve than the blind deconvolution.

2.3.4 Blur kernel from 3D geometry

A 2D blur kernel represents the averaged trajectory of pixels in blurred image during exposure time. If there is no object motion, only the camera motion makes this trajectory, and it can be easily found from the 3D camera geometry. Various shapes

of projected landmarks' trajectories can be generated from various camera motions composed of translation and rotation. Then the 2D blur kernel $\mathcal{K}(\mathbf{u})$ with the domain $\mathbf{u} \in \mathbb{R}^2$ can be represented by the projected trajectory $\tilde{\mathbf{x}}_t = h(g((\mathbf{P}_t^k)^{-1} \cdot L))$ during the exposure time $t \in [t_0, T]$, where the function $h(\cdot)$ is dehomogenization such that $\tilde{\mathbf{x}}_t = h(\mathbf{x}_t)$. With the indicator function $\delta(\mathbf{u}) = \begin{cases} 0, & \text{if } \mathbf{u} = (0,0), \\ 1, & \text{if else.} \end{cases}$, the resulting 2D blur kernel is

$$\mathcal{K}(\mathbf{u}) = \frac{1}{T - t_0} \int_{t_0}^{T} \delta(\mathbf{u} - \tilde{\mathbf{x}}_t) dt. \tag{2.3}$$

It is assumed that the velocity of pixel is constant during short exposure time, thus all positions in blur kernel have constant weight value.

When the camera exposure time is very short, the blur kernel can be approximated as a straight line since the trajectory of projected pixel is short. In [37], the approximation is applied and the blur kernel is easily computed from an inter-frame difference of pixel position $\tilde{\mathbf{x}}_T^k - \tilde{\mathbf{x}}_T^{k-1}$. The blur kernels in [37] is simply parameterized with the kernel direction and magnitude, and they are estimated for subdivided small patches in image and used for patch-wise deconvolution.

However, this linear (straight line) approximation has some limitations. First, a camera motion which has rotational component in camera principal axis (z-axis) can bring significant errors in blur kernel. A pixel motion from small camera rotation with pitch or yaw axes (x or y axes) is almost not curved, but camera rotation with roll axis (z-axis) can make significant curvature in blur kernel. Figure 2.4 shows the examples of projected point's trajectory as a result of 8 degrees camera rotation with different axis. The image size is 640 by 480 pixels^2 and the starting point of projection is (100, 100) pixels away from the image center. The average curvature

value κ ($\kappa = \lim_{\Delta\tilde{\mathbf{x}}\to 0}\left|\frac{\Delta\varphi}{\Delta\tilde{\mathbf{x}}}\right|$, φ is instant direction of line) of each trajectory is 0.0589, 0.0712, 0.4011 degree/pixel for x-axis, y-axis, and z-axis rotation of the camera, respectively. This indicates that the curvature of motion blur from camera rolling is not negligible in blur kernel approximation.

The second limitation of linear approximation is that the effect of radial distortion cannot be handled. Although a camera motion is pure translation, a pixel motion in an image can be curved due to the camera lens distortion. To apply approximated linear kernel, the radial distortion should be removed first.

To handle more general shape of blur kernel, therefore, a nonparametric representation of blur kernel is used. The nonparametric blur kernel is calculated directly from Equation (2.3), not using the linear approximation. Additionally, pixel-wise blur kernels are estimated instead of patch-wise blur kernels, to handle in-plane camera rotation where the length of blur kernel is varied from the distance to the rotation center. In the practical implementation, the blur kernel is represented in discrete form with N elements,

$$\mathcal{K}_{i,j} = \frac{1}{N}\sum_{n=1}^{N}\delta_d((i,j) - \tilde{\mathbf{x}}_{t_0+n\frac{T-t_0}{N}}), \tag{2.4}$$

where $\delta_d(i,j) = \begin{cases} 1, & \text{if } (i^2+j^2)^{1/2} < 0.5, \\ 0, & \text{if else.} \end{cases}$ is the discrete indicator function combined with respect to kernel coordinate (i,j). The intermediate pixel position $\tilde{\mathbf{x}}_{t_0+n\frac{T-t_0}{N}}$ is calculated from the intermediate camera pose $\mathbf{P}_{t_0+n\frac{T-t_0}{N}}$ as $\tilde{\mathbf{x}}_{t_0+n\frac{T-t_0}{N}} = h(g((\mathbf{P}_{t_0+n\frac{T-t_0}{N}})^{-1}\cdot L))$.

The intermediate camera pose $\mathbf{P}_{t_0+n\frac{T-t_0}{N}}$ can be calculated using the exponential map $\exp(\cdot)$ and log map $\log(\cdot)$ between $SE(3)$ and its Lie algebra $se(3)$. The incremental difference between $\mathbf{P}_{t_0}$ and $\mathbf{P}_T$ can be expressed in $se(3)$ as $\mathbf{\Delta p} =$

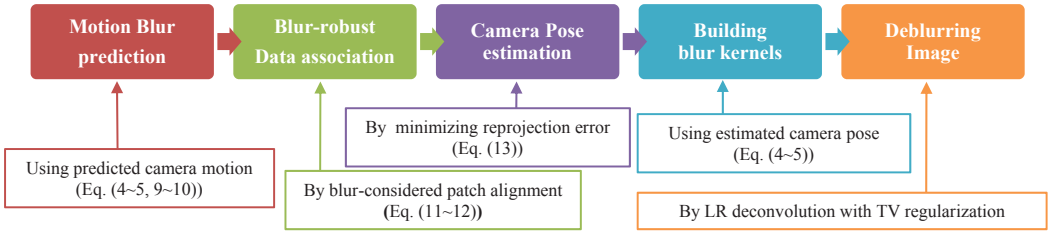


Figure 2.5: Overall procedure of the proposed algorithm.

$\log(\mathbf{P}_{t_0}^{-1}\mathbf{P}_T)/N$. Then, the intermediate camera pose $\mathbf{P}_{t_0+n\frac{T-t_0}{N}}$ can be calculated as

$$\mathbf{P}_{t_0+n\frac{T-t_0}{N}} = \mathbf{P}_{t_0} \cdot \exp(n\Delta\mathbf{p}). \tag{2.5}$$

Note that in deriving (2.5) we apply the approximation of the Baker-Campbell-Hausdorff (BCH) formula [38] with the first two terms, which says that z satisfying $\exp(z) = \exp(x)\exp(y)$ is given by

$$z = x + y + \frac{1}{2}[x, y] + \frac{1}{12}[x, [x, y]] - \frac{1}{12}[y, [x, y]] + \ldots, \tag{2.6}$$

where $[\cdot, \cdot]$ is the matrix commutator given by $[A, B] = AB - BA$.

Since the visual SLAM applied in this study is sparse point feature based, most pixels in image do not have its own 3D position information. For non-reconstructed pixels, a depth value from nearest reconstructed landmark to it is given and the pixel is reprojected to 3D space to get the rough 3D position of the pixel. The proposed blur kernel estimation method can be more effectively used in dense reconstruction system, where every pixel in image has its reconstructed 3D position.

2.3.5 Reconstruction error and blur kernel error

In the proposed blur kernel estimation, the accuracy of 3D reconstruction and camera pose is directly related to the accuracy of blur kernel. Since 3D position error of

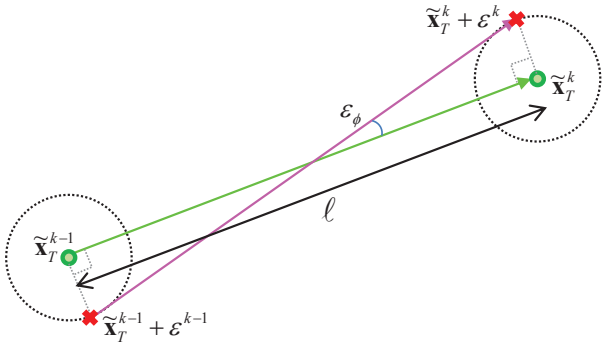


Figure 2.6: The worst case that estimated blur kernel has maximum error with given reprojection errors.

landmark and camera pose error are highly correlated, it is difficult to analyze the effect those errors separately. However, those two errors are reflected in the reprojection error of landmark, and we can investigate the effect of SLAM errors in blur kernel estimation using the reprojection error. The relationship between the reprojection error and the blur kernel error is explained below.

To simplify the error analysis, the blur kernel is assumed to be linear. As described in [37], the length ℓ and the direction ϕ of linear blur kernel can be calculated as

$$\begin{aligned} \ell &= \left| h(g((\mathbf{P}_T^k)^{-1} \cdot L)) - h(g((\mathbf{P}_T^{k-1})^{-1} \cdot L)) \right| \cdot \frac{(T - t_0)}{T}, \\ \phi &= \arctan(\frac{v}{u}), [u, v]^\top = \frac{h(g((\mathbf{P}_T^k)^{-1} \cdot L)) - h(g((\mathbf{P}_T^{k-1})^{-1} \cdot L))}{T}. \end{aligned} \tag{2.7}$$

Let ϵ_{re}^{k-1} and ϵ_{re}^k be the reprojection error of landmark at frame $k-1$ and k respectively. Then the upper bound of magnitude error ϵ_ℓ of blur kernel is derived using

Equation (2.7) as,

$$\begin{aligned}
|\epsilon_\ell| &= \left|\ell - \ell'\right| \\
&= \left|\ell - \left|(h(g((\mathbf{P}_T^k)^{-1} \cdot L)) + \epsilon_{re}^{k-1}) - (h(g((\mathbf{P}_T^{k-1})^{-1} \cdot L)) + \epsilon_{re}^{k})\right| \cdot \frac{(T - t_0)}{T}\right| \\
&\leq \left|\epsilon_{re}^{k-1} - \epsilon_{re}^{k}\right| \cdot \frac{(T - t_0)}{T}.
\end{aligned} \tag{2.8}$$

Where ℓ' is the error included kernel magnitude. The upper bound of direction error ϵ_ϕ can be easily derived from the figure 2.6, as

$$\begin{aligned}
\left|\epsilon_\phi\right| &= \left|\phi - \phi'\right| \\
&\leq \arctan\left(\frac{\left|\epsilon_{re}^{k-1} - \epsilon_{re}^{k}\right|}{\ell}\right).
\end{aligned} \tag{2.9}$$

There are two error sources in 3D reconstruction and camera localization. First one is wrong data association, and second one is measurement noise in feature position. Landmarks with a large reconstruction error, usually come from the wrong data associations, can be handled by the outlier rejection. However, landmarks with a small error due to measurement noise may not be filtered out by the outlier rejection, and this might affect the accuracy of the blur kernel estimation.

When a landmark is reconstructed from N number of observations (images) with measurement noise $\epsilon^n \quad (n = 1, ..., N)$, the 3D position L of the landmark is determined by minimizing the reprojection errors for all measurements. Then L will be projected into next frames with the expected error $\sqrt{\frac{1}{N}\Sigma_{n=1}^{N}\epsilon^{n2}}$. For example, let the average measurement error of landmark position in image be 2 pixels, the length of blur kernel be 20 pixles, and the camera exposure time be 50 percent of frame interval. Then the expected reprojection error is also 2 pixels, and the upper bound of magnitude and angle errors of linear blur kernel calculated from

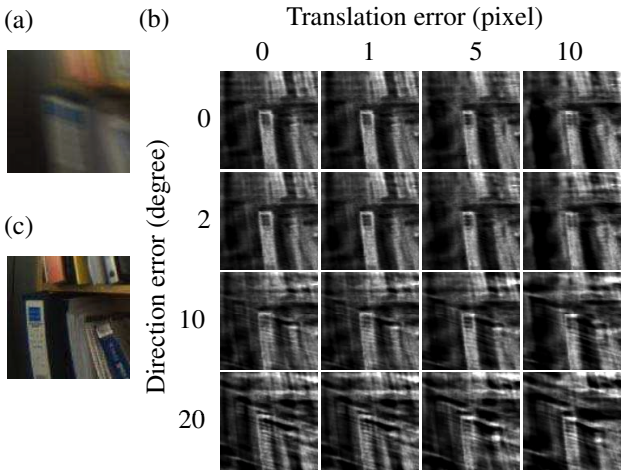


Figure 2.7: Results of deblurring in a presence of kernel error. (a) A blurred image. (b) Deblurred images by kernels with translation and direction error. (c) Sharp (unblurred) image taken at different moment for comparison.

Equation (2.8,2.9) is 2 pixels and 11.3 degrees. Although a blur kernel has errors of those upper bound values, the deblurring result is not too degraded as shown in fig 2.7 which shows the results of various kernel errors.

2.4 Visual SLAM and Deblurring

Figure 2.5 summarizes the overall procedure of the proposed visual SLAM algorithm. First, a motion blur is predicted and a blurred version of landmark's template is approximated to perform the blur-robust data association. After data association the camera pose is refined. Finally, the blur kernels for each landmark are built and the blurred image is recovered using the obtained kernel to conduct the remaining tasks for visual SLAM.

The proposed visual SLAM system is implemented based on [12] which uses a

parallel processing of localization and mapping. The initial reconstruction is done using two images with a user specified baseline, then the result is bundle adjusted to obtain a more accurate map. As the camera moves, the camera pose is calculated by minimizing reprojection errors of reconstructed landmarks, and new 3D landmarks are registered with their appearances in a form of small patches. To handle the viewpoint changes for landmarks, the landmark patches are updated by affine warping calculated from the camera pose. In the proposed blur-robust data association, the patches are additionally blur adjusted using Equation (2.12).

An image pyramid is used to extract point features because high-level (low-resolution) images are less sensitive to motion blur than low-level images. The image pyramid has four levels and point features are detected using FAST-10 [39] corner detector. Many successful data associations are from high-level images in blurred images, and those are useful for calculating camera pose and estimating blur kernels.

2.4.1 Blur-robust data association

Since the data association in visual SLAM can be regarded as a tracking of a small patch, a tracking algorithm robust to motion blur can be a solution for handling the motion blur for visual SLAM. In [40], the image region tracking with blurred images is performed by blurring the template image, rather than deblurring the current blurred image. [41] extended the blur model of [40] from the translational blur to any complex blur, and [42] improved the efficiency of [40] by approximating a blurred image using image derivatives. Those tracking methods are performed in the 2D image space. On the other hand, using a 3D structure, we can easily predict a motion blur using that information and give the predicted value as an initial value

for the tracking to boost the tracking performances.

With a help of bundle adjustment, high accuracy of reconstruction and localization can be achieved with unblurred scene although a monocular camera is used. However, it is hard to estimate accurate $\mathbf{P}^k$ when the image is blurred, since point features are used for calculating $\mathbf{P}^k$, which are not robust to motion blur. The camera pose $\mathbf{P}^k$ has to be estimated from detected feature points, but not enough points are extracted in the blurred image and data association becomes difficult. To solve this problem, a blur-robust data association method is proposed as follows.

First, the pose of camera is predicted for a new frame. The auto-regressive process on $\mathbf{P}$ is used for camera pose prediction by assuming smooth camera motion. The auto-regressive (AR) state dynamics $\mathbf{a}^k$ is updated as

$$\mathbf{a}^k = a \log((\mathbf{P}_T^{k-1})^{-1} \cdot \mathbf{P}_T^k), \tag{2.10}$$

where a is the first-order AR process parameter. Then the new camera pose at frame k can be predicted as

$$\widehat{\mathbf{P}_T^k} = \mathbf{P}_T^{k-1} \cdot \exp(\mathbf{a}^{k-1}). \tag{2.11}$$

The function $\log(\cdot)$ and $\exp(\cdot)$ are log maps and exponential maps, respectively, as described in the previous section.

The predicted camera pose $\widehat{\mathbf{P}_T^k}$ does not consider the observation of the current image I^k, thus the value is not accurate and needs to be refined. In conventional visual SLAM, point features are extracted from the current image and they are matched with their stored appearances, for example 8×8 patches, of reconstructed landmarks. For successful matches, subpixel refinement using patch alignment algorithm such as inverse compositional algorithm [43] is performed to find an accurate

position of the landmark. In a blurred image, however, the patch alignment is hard to be achieved and this results in no or few features are successfully associated with the landmarks.

To handle appearance differences between the stored landmark patches and the blurred patches in the current image, a blurred version of the landmark patch is generated using the estimated blur kernel. We can easily generate a blurred patch by convolution with kernel from Equation (2.4). Using the predicted camera pose from Equation (2.11), first a blur kernel $\widehat{\mathcal{K}}$ is predicted and an initially blurred patch $\mathcal{T}_b$ for the template patch $\mathcal{T}_p$ is synthesized using the obtained kernel as $\mathcal{T}_b = \mathcal{T}_p * \widehat{\mathcal{K}}$. Using the patch $\mathcal{T}_b$, the sliding window template matching is performed around the projected position of each landmark to find the feature's observed position roughly.

Since the initially blurred patch $\mathcal{T}_b$ is generated by predicted blur kernel $\widehat{\mathcal{K}}$ which is not exact, refinement of blur kernel as well as blurred patch using current observation needs be performed to achieve more accurate data association. The objective of blur kernel refinement is to find a blur kernel which satisfies the equation $\mathcal{T}_{obs} = \mathcal{K} * \mathcal{T}_b$, where $\mathcal{T}_{obs}$ is the observed appearance of patch. This can be achieved by iteratively minimizing the difference between $\mathcal{T}_{obs}$ and $\mathcal{K} * \mathcal{T}_b$, but the convolution operation is replaced by the approximation method in [42] to avoid the large computation time of convolution.

For a small patch transformation $\Delta\Theta$ composed of x and y translation, the deformed appearance $\mathcal{T}(\Delta\Theta)$ of initially blurred patch $\mathcal{T}_b$ including blur effect can be approximated by the second-order Taylor expansion as [37]

$$\mathcal{T}(\Delta\Theta) \approx \mathcal{T}_b + a\mathcal{J}_{\mathcal{T}_b}\Delta\Theta + b\Delta\Theta^\top \mathcal{H}_{\mathcal{T}_b}\Delta\Theta, \tag{2.12}$$

The constants a and b are related to the exposure time,

$$a = \frac{t_0 + T}{2T}, \quad b = \frac{T^2 + Tt_0 + t_0^2}{3T^2}. \tag{2.13}$$

and the matrices $\mathcal{J}_{\mathcal{T}_b}$ and $\mathcal{H}_{\mathcal{T}_b}$ are the Jacobian and the Hessian of the patch $\mathcal{T}_b$. Based on this approximation, the transformation vector $\Delta\Theta$ is estimated by the blur-robust version [42] of Efficient Second-order Minimization (ESM) [44] tracking algorithm. The landmark patch's position is then refined by ESM iteration, and successfully matched and refined landmarks with sub-pixel accuracy are obtained and will be used to estimate the accurate camera pose.

2.4.2 Deblurring for SLAM

After the blur-robust data association described in previous section, the data association outliers have to be filtered because ESM does not guarantee the result to be global optimum. Any types of outlier filtering methods such as RANSAC can be used, but simple threshold filtering based on the reprojection error is sufficient in this study.

After the outlier rejection, the new camera pose $\mathbf{P}_T^k$ is calculated by minimizing the sum of reprojection errors for inlier matches. The objective function is represented as

$$\mathbf{P}_T^k = \arg\min_{\mathbf{P}} \sum_{m=1}^{M} \left\| h(g(\mathbf{P}^{-1} \cdot L^m)) - \tilde{\mathbf{x}}_{obs}^m \right\|, \tag{2.14}$$

where m_{obs} is the observed landmark position from data association. Then using the kernel estimation method described in Section III, the blur kernel for all pixels in the image can be obtained and image deblurring can be easily done using those kernels. Figure 2.8 shows the example of estimated kernels at different landmarks and corresponding deblurring result. We can deblur every input frame for further

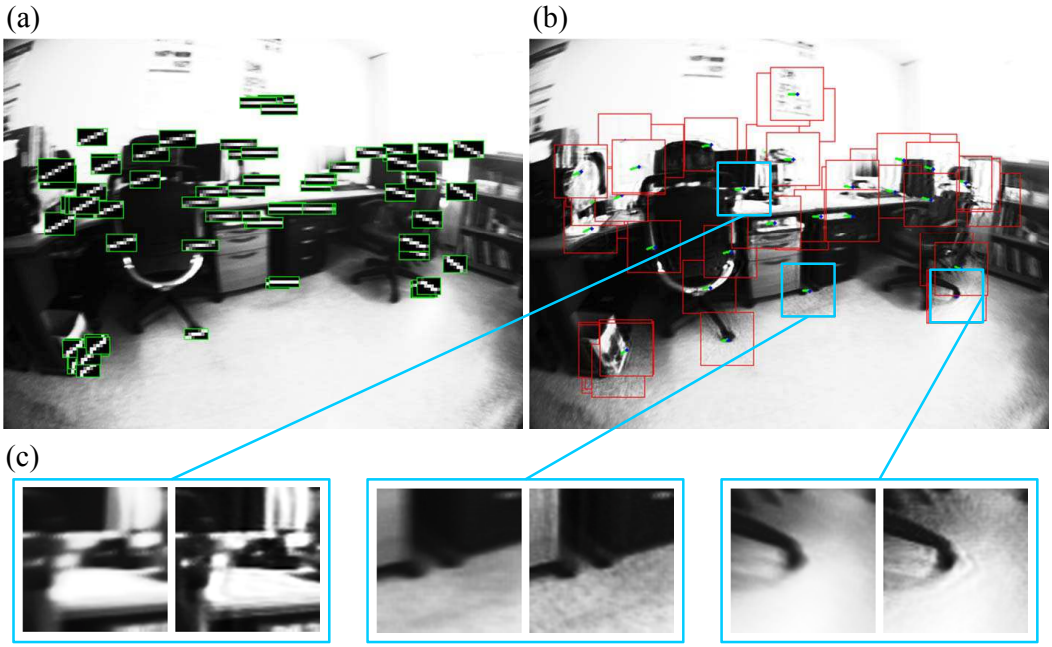


Figure 2.8: Example of deblurred patches. (a) Blur kernels at each landmark. (b) Partially deblurred image. (c) Close-up of patches (left: input, right: deblurred)

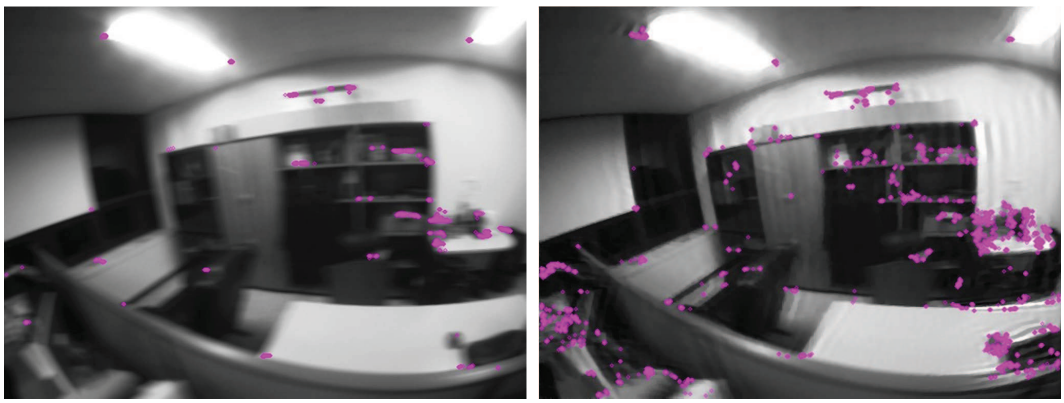


Figure 2.9: Extracted FAST-10 corners from the blurred (left) and deblurred (right) image.

vision tasks such as recognition, or only deblur the keyframes when new keyframe is added to register new landmarks. In this study, the second option is chosen because we focus on the SLAM performance and have to save the computational cost.

For the deblurring, Lucy-Richardson (LR) deconvolution algorithm [33] is applied with total variational (TV) regularization. Let $I(\mathbf{x})$ be the observed (blurred) pixel intensity, and $I^n(\mathbf{x})$ be the intermediate image during LR iteration. Then the LR iteration is composed of estimating delta image $D^n(\mathbf{x}) = I(\mathbf{x}) / \left(\mathcal{K} * I^n(\mathbf{x})\right)$ and updating the deblurred (sharp) image $I^{n+1}(\mathbf{x}) = I^n(\mathbf{x}) \left(\mathcal{K}^* * D^n(\mathbf{x})\right)$, with the initial solution $I^0(\mathbf{x}) = I(\mathbf{x})$, where $\mathcal{K}^*$ is adjoint kernel of $\mathcal{K}$ i.e., $\mathcal{K}^*_{i,j} = \mathcal{K}_{-i,-j}$. Since the estimated blur kernels have simple shape and short trajectory, a small number of LR iteration is sufficient. In the experiment, the number of LR iteration is set to 50, which takes about 300 millisecond in the proposed GPGPU implementation. Different from the patch-wise deblurring in [37], the pixel-wise deblurring does not suffer from the boundary effect which comes from discontinuities of blur kernel across patches and FFT (Fast Fourier Transform) operation.

On the resulting deblurred image, the feature detector runs again and obtains interest points for new landmark registration. Compared with the blurred image, the restored (deblurred) image provides more good features for mapping. Figure 2.9 shows an example of a deblurred image and detected features using various feature detectors. Several widely used feature detectors in visual SLAM systems are tested, including Fast-10 detector, Harris corner detector and SURF detector.

FAST-10 corners with high cornerness values measured by the Shi and Tomasi (ST) cornerness measure [45]. In the deblurred image, 2436 corners are extracted and their average ST measure is 134.8 for 7×7 window. While, in blurred image, only 557 corners are detected and their average ST measure is 81.2. This means that

by deblurring images, we can obtain more *good features to track*. This is critical when the camera moves fast for a number of frames. Without deblurring, it is hard to obtain enough features for localization, then the accuracy of visual SLAM decreases and sometimes the camera pose can be lost.

2.5 Experiments

In the experiment section, we will focus on two performance factors of the proposed algorithm. One is the improvement of visual SLAM performance, and the other is the image-deblurring quality. Point grey research's Dragonfly 2 is used with the fish-eye lens of 160° field of view for image capturing. The size of the input image is 640 × 480, and all tasks are processed with gray scale images. The experiments are done on a 3.3GHz quad core PC and two threads (mapping thread and localization thread) run on each core at the same time. For the GPU-based deconvolution, NVIDIA's GeForce GTX570 with 480 stream processors and 1280MB video memory is used.

When the blur-robust data association is activated, the average processing time for all localization processes is about 30ms per frame, while it takes 12ms with no blur handling. Thus the system ensures the frame rate of at least 30 fps. The processing time for image deblurring is about 300ms, which is acceptable because the keyframes are added infrequently, and adding keyframe is done at the background thread. The main bottleneck is the LR iteration, thus advanced deconvolution algorithm and GPGPU implementation may improve the computation speed. When the length of blur kernel is less than 2 pixels, the deblurring is skipped and original input image is used for mapping.

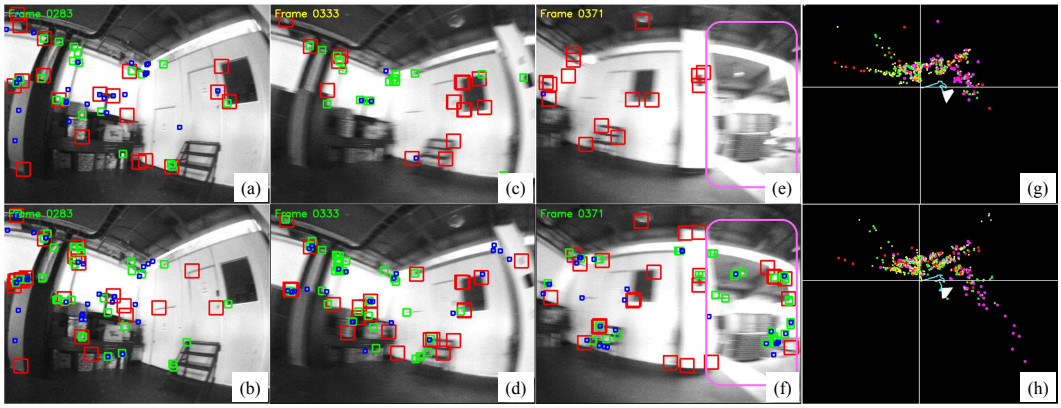


Figure 2.10: Data association and mapping of SLAM systems with (bottom row) and without (top row) the blur handling for translation-dominant camera motion. Data association results of selected frames (a $\sim$ f). Results of mapping by each system (g, h). The colors of landmarks in the scenes and the map represent the different levels of image pyramids where the landmarks are extracted.

2.5.1 Performances of visual SLAM

The performance of proposed blur-handled visual SLAM algorithm is tested by comparing it with conventional keyframe-based SLAM [12]. First, the experiment on the blur-robust data association is conducted for translation-dominant camera motion. After the initial reconstruction and mapping for some frames, the camera moves rapidly to make motion blur. With a smooth camera motion, both systems show good data association results as shown in Figure 2.10-(a, b). When a motion blur occurs, the number of tracked landmarks decreases without the blur-robust data association (Figure 2.10-(c, d)). When the camera observes unmapped region (magenta rectangles in Figure 2.10-(e, f)) where a motion blur exists, no new landmark is registered to the map with the conventional system (Figure 2.10-(e)), while the

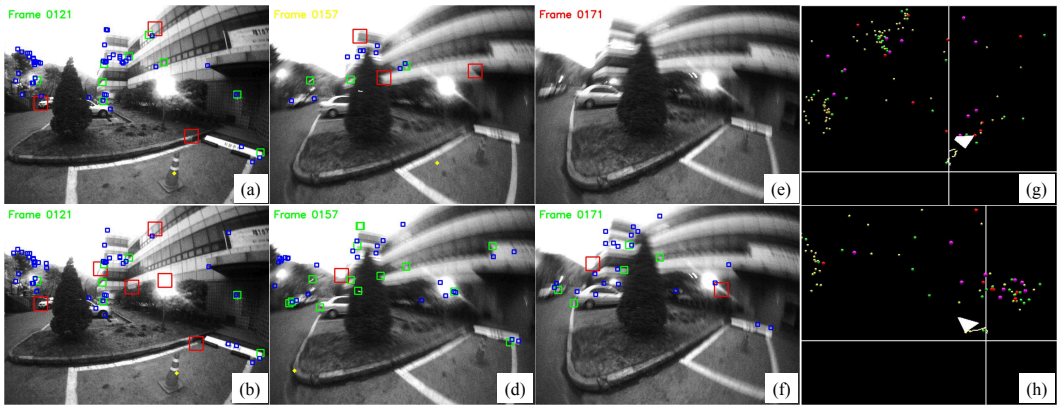


Figure 2.11: Data association and mapping of SLAM systems with (bottom row) and without (top row) the blur handling for rotation-dominant camera motion. Data association results of selected frames (a $\sim$ f). Results of mapping by each system (g, h). The colors of landmarks in the scenes and the map represent the different levels of image pyramids where the landmarks are extracted.

proposed system deblurs the image and extracts and registers new landmarks (Figure 2.10-(f)). As a result, conventional SLAM system fails to continue mapping, and the resulting map is incomplete (Figure 2.10-(g)). On the other hand, the proposed blur-handling system reconstructs the map of entire visited region (Figure 2.10-(h)).

The robustness of blur handling data association is also tested for rotation-dominant motion, as shown in Figure 2.11. The camera moves left and then suddenly rotates with z-axis. Without the blur handling data association, the number of matched landmark decreases rapidly and finally the camera loses its pose and drifts, while blur-handled system maintain its camera pose correctly.

The number of reconstructed landmarks and the number of tracked landmarks are compared for the conventional SLAM system and the proposed blur-handled system, respectively. The number of reconstructed landmarks demonstrates the

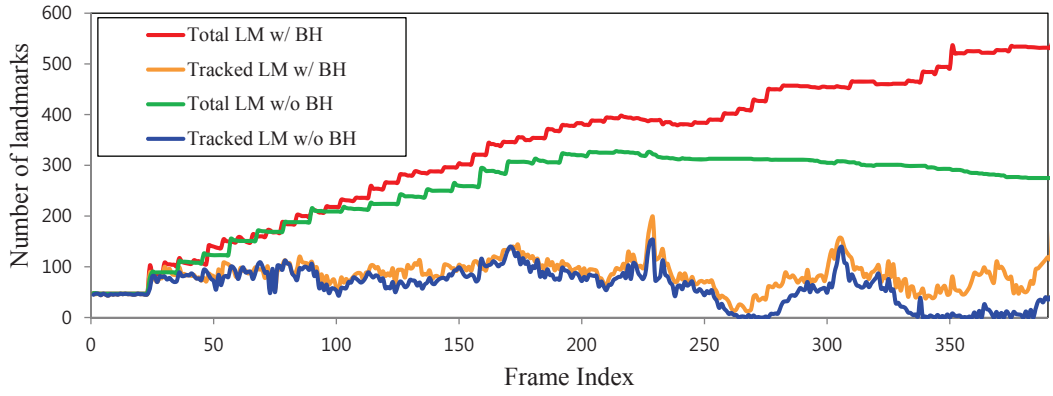


Figure 2.12: Comparison of the numbers of total landmarks in maps, and the numbers of currently tracked landmarks in each frame.

contribution of the proposed deblurring for mapping, and the number of tracked landmarks shows how the blur-robust data association improve the tracking quality. Since the system frequently lose the camera pose without blur-robust data association, the keyframe-based relocalization [12] is used to recover the camera poses to continuously compare the number of landmarks. Figure 2.12 shows the plots of those values versus the frame index for the input sequence used in Figure 2.10. Before the severe motion blur occurs (about the 250th frame), the numbers of landmarks are similar for both systems. Under the motion blur, however, the number of tracked landmarks rapidly decreases and the number of reconstructed landmarks rarely increases in the system with no blur handling.

For real scene data, it is difficult to test the accuracy of localization and mapping of SLAM since it is hard to obtain ground truth data. Instead, the SLAM performance is measured indirectly by measuring the reprojection errors for recon-

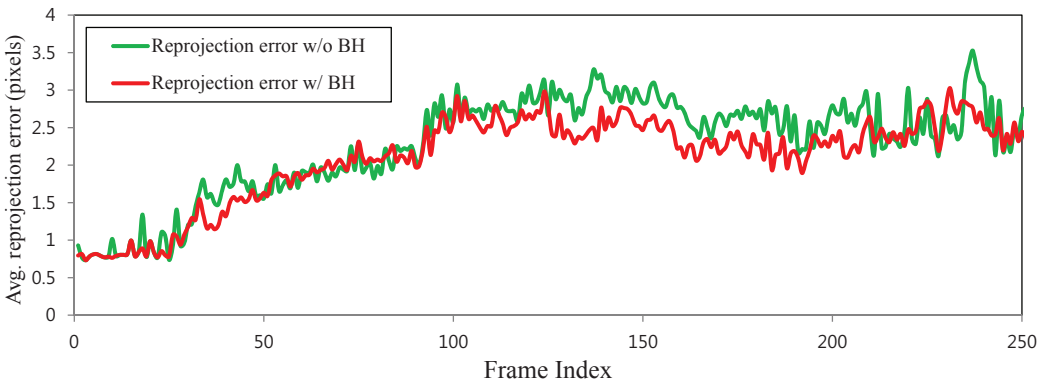


Figure 2.13: Comparison of the reprojection error.

Table 2.1: Comparison of conventional and blur-handled system.

	Total reconstructed LM	Average # of Tracked LM	Average Reproj. Err
Conventional	273	60.1	2.27
Blur-handled	542	83.8	2.09

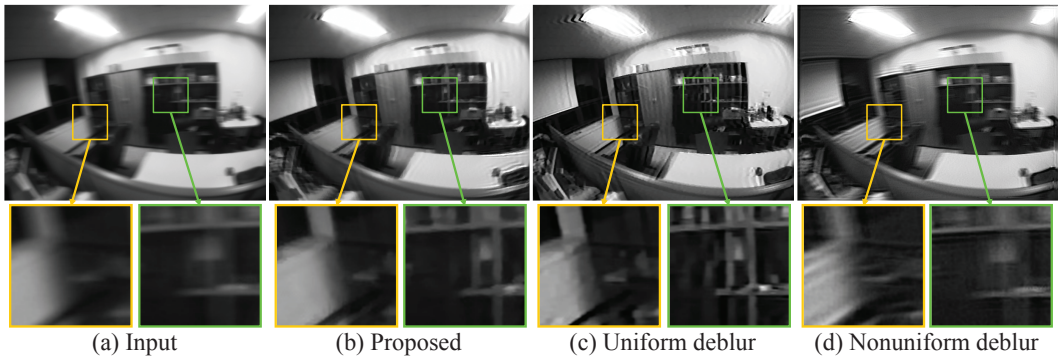


Figure 2.14: Comparison of image deblurring results for fast camera translation. (a) Blurred input image. (b) Deblurred by the proposed method. (c) Uniform deblurring [2]. (d) Non-uniform deblurring [3].

structed landmarks. The reprojection errors of the conventional SLAM system and the proposed blur-handled system are compared until the conventional SLAM system lost the camera pose. Figure 2.13 shows the results for the same sequence used in Figure 2.10.

The performance comparison of the conventional system and the blur-handled system by presenting the average values of above measured values is summarized in Table 2.1. The total number of reconstructed landmarks is from the last frame (720th frame), and the average number of tracked landmarks is calculated for all frames. The average reprojection errors are calculated for first 250th frames, because after the 250th frame the conventional SLAM system frequently loses the camera pose and relies on the relocation.

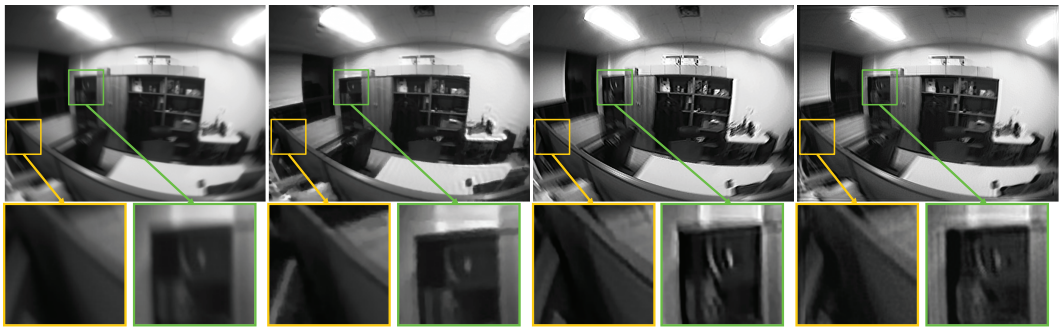


Figure 2.15: Comparison of image deblurring results for fast camera rotation. (a) Blurred input image. (b) Deblurred by the proposed method. (c) Uniform deblurring [2]. (d) Non-uniform deblurring [3].

2.5.2 Deblurring qualities

To compare the performance of the SLAM-combined deblurring to other existing deblurring algorithms, two types of motion blur with fast camera translation and fast camera rotation are generated. Since the camera is moved by human hand, the translation scene has some rotational component, and the rotation has also some translational component, which are more general situation than pure translation and pure rotation.

Figure 2.14 shows the deblurring result of the proposed algorithm for fast camera translation, compared with other single image deblurring methods - the uniform deblurring [2] and non-uniform deblurring [3]. The results of [2] and [3] are obtained using the public software from each author. Although the results of the proposed deblurring method suffer from some ringing artifacts, we can see that edges in objects are recovered well in the results. The uniform deblurring method of [2] recovers sharp images well for some regions since the camera translation makes almost uniform

motion blur, but has more ringing artifacts than the result of the proposed method since it does not consider the effect of scene depth. The non-uniform deblurring of [3], which approximate a 6 DOF camera motion as a x-y-z rotation, does not recover this motion blur from translational camera motion.

For a severe rotational motion blur as shown in Figure 2.15, the proposed method also gives better deblurring results than others. The rotational motion blur is highly non-uniform, thus the result of [2] is worse than its result for translational motion blur. The non-uniform deblurring of [3] gives correct deblurring for some regions, but overall quality is not satisfactory. The average computation time of [2] is 36 seconds by their C++ implementation with no hardware acceleration, and the average computation time of [2] is 30 minutes by their MATLAB implementation. On the other hand, the SLAM-combined method requires about 300 ms per frame with GPGPU hardware acceleration, which enables real time operation.

2.6 Summary

In this chapter, a new approach for handling motion blur in visual SLAM is proposed. From a camera pose and a reconstructed 3D point structure, a motion blur for each landmark can be easily predicted without any complicated image processing algorithm. Then using the predicted motion blur and the blur-robust patch alignment methods, the data association of visual SLAM could be robust to motion blur, thus estimating an accurate camera pose with a blurred scene is possible. A blur kernel from the accurate camera pose is used to deblur the input image, and more good features to track are obtained and the system can continue the SLAM process for the next frames.

Chapter 3

Sparse 3D Reconstruction and Image Super-Resolution

3.1 Introduction

The resolution of image is one of important factors in various computer vision algorithms. Especially in 3D reconstruction with a single camera, the accuracy of camera pose and scene structure estimation is highly affected by image resolution. Image resolution is an important factor for achieving sufficient accuracy of various geometry-related computer vision algorithms including 3D reconstruction, since it influence the feature detection, localization and matching. 3D reconstruction and camera pose estimation requires high accuracy of pixel correspondence, *i.e.*, sub-pixel accuracy, and the higher resolution of image provides the more accurate estimation results. Even in an image of a scene, the resolutions of objects vary according to their sizes and depths. A small measurement error of pixel position does not bring large errors in object position and camera pose when an object is close to the cam-

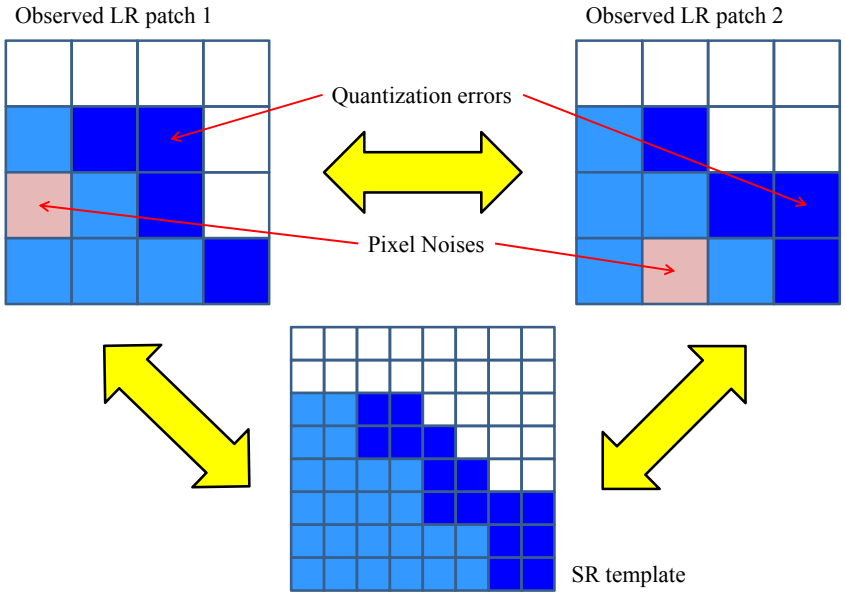


Figure 3.1: Illustration of similarity between landmark patches. Using the high-resolution template can provide higher similarity than using low-resolution patches by reducing the sensitivity of pixel noise and quantization error.

era, while, it does significantly when the object is far from the camera. Therefore, it is necessary to enhance the image resolution to reduce the sensitivity to the image measurement error caused by pixel noise or quantization error and achieve reliable and accurate 3D reconstruction as illustrated in Figure 3.1. Furthermore, high-resolution image helps the finding scene point correspondences in the case of large scale difference between a stored template and an observed patch.

Image super-resolution, the method for enhancing image resolution, has two different approaches: reconstruction-based approach and learning-based approach. The reconstruction-based approach, which is related to the proposed approach, infers the high-resolution pixel by merging multiple observations of a target pixel. Multiple ob-

servations are obtained by finding corresponding pixels through an image sequence. Therefore, finding accurate pixel-wise correspondences is the key for the success of the reconstruction-based super-resolution. For general scenes, these correspondences can be obtained up to sub-pixel accuracy using optical flow algorithms. However, optical flow in low-resolution images usually do not provide enough accuracy in correspondences, producing unsatisfactory results. Several iterative methods [46, 47] alternately estimate a high-resolution image and pixel correspondences, and show better results. However, these methods usually take a very large amount of computation time, and thus they are not appropriate for real-time applications such as visual odometry or SLAM.

In this chapter, a method for image patch super-resolution that specialized for 3D reconstruction is proposed. Estimated camera motion and scene position from the 3D reconstruction algorithm helps robust and accurate patch registration for super-resolution. 6 degrees of freedom patch pose including 3D position and 3D rotation is estimated in geometric particle filtering visual SLAM framework [22], and this pose estimate is combined with the estimated camera pose to predict a patch registration in image sequence. High resolution pixel estimation is performed by back-projection [48] model, and implemented using extended Kalman filter (EKF), which enables simultaneous estimation of high resolution images as well as accurate patch pose.

There are several methods that have better performances than the Kalman filter-based method for image super-resolution from image sequence, but Kalman filter-based estimation is employed in this study because incremental estimation of high-resolution patch with respect to currently observed images is important in 3D reconstruction system. With the Kalman filter-based super-resolution, the 3D re-

construction system can update the resolution of landmark patches for every time step and use them for better data association, camera pose and landmark pose estimation.

At the experiment section, the improvement of 3D reconstruction accuracy by super-resolution is presented first, and then the results of image super-resolution is compared with other multiple image super-resolution methods.

3.2 Patch-based Image Super-Resolution

Patch-based image super resolution using multiple images is one of the classical problems in image processing and computer vision research. The basic theory of this problem is formulated as *back-projection method* [48] where a latent high resolution image is found by minimizing the reconstruction error between an observed images and the simulated low resolution image from the estimated high resolution image. To simulate an observed low resolution images from the estimated high resolution image, accurate image registration is required. Let $\mathcal{T}^H$ be the currently estimated high resolution image represented by 1D vector, and let $\mathcal{T}_k^L$ be the sequence of simulated low resolution images, where k is the index of image sequence. For notational simplicity, the subscript k is omitted when there is no ambiguity. The relationship between $\mathcal{T}^H$ an $\mathcal{T}_k^L$ can be modeled by the combination of geometric image warping $\mathcal{W}_k$, spatial blurring $\mathbf{B}_k$, and pixel down-sampling $\mathbf{D}_k$ as follows:

$$\mathcal{T}_k^L = \mathbf{D}_k(\mathbf{B}_k(\mathcal{W}_k(\mathcal{T}^H))). \tag{3.1}$$

It is assumed that the spatial blurring and down-sampling functions are invariant to image index i, thus $\mathbf{B}_k = \mathbf{B}$ and $\mathbf{D}_k = \mathbf{D}$ for all k. The reconstruction error R_k

for the estimation of $\mathcal{T}^H$ can be defined from the observed low resolution sequence $\mathcal{T}_k^O$ as

$$R_k = \|\mathcal{T}_k^L - \mathcal{T}_k^O\|_2 = \|\mathbf{D}_k(\mathbf{B}_k(\mathcal{W}_k(\mathcal{T}^H))) - \mathcal{T}_k^O\|_2. \tag{3.2}$$

Before the minimization of the error function (3.2) with respect to high-resolution patch $\mathcal{T}^H$, the image warping $\mathcal{W}_k$ has to be estimated first. Generally, this is achieved by applying image registration methods such as the inverse compositional method (IC) [43] or the efficient second order minimization (ESM) [44] to image pair. These registration methods, however, requires good initial solutions to converge because the methods are based on the image derivative which can handle only small update. Therefore, when a baseline of image pair is large, those registration methods may fail and the super-resolution cannot be achieved correctly. Wide baseline of image pair can be handled if we initialize the registration by wide baseline matching methods, such as feature matching using feature descriptors. In the proposed 3D reconstruction-combined method, this wide baseline matching can be more easily achieved by utilizing 3D geometry. Data association of visual SLAM is performed by projecting mapped landmarks into current view and finding corresponding feature point in an observation image. This reduces search regions for feature matching, and matched features can be found with higher speed and accuracy than image only-based feature matching.

The minimization of reconstruction error given by the Equation (3.2) can achieved by various method such as gradient-based local optimization and MRF-based global optimization. Given that the observation images $\mathcal{T}_k^O$ are sequentially obtained for every frame, using filtering-based method such as Kalman filter is an effective solution. Several works successfully apply Kalman filter to image super-resolution meth-

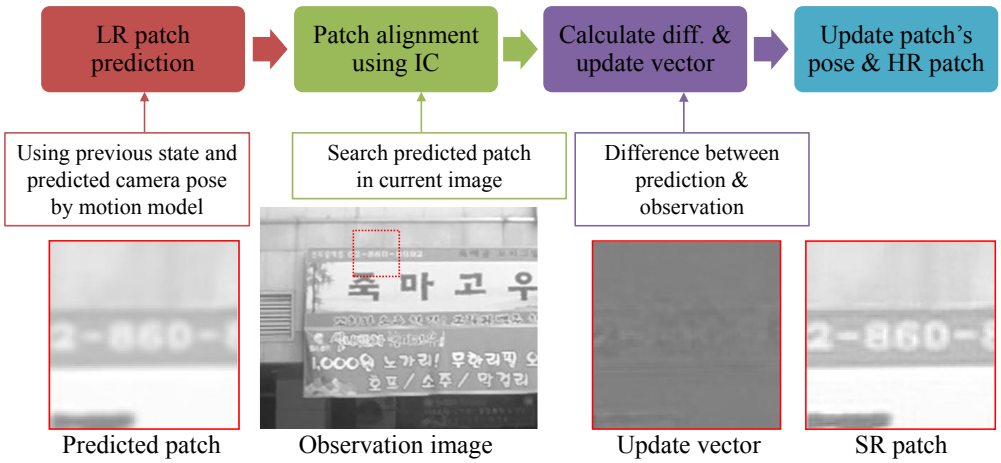


Figure 3.2: EKF steps for super-resolution of landmark template and example images for each step.

ods [49–52]. In Kalman filter-based image super resolution, a latent high-resolution patch is initially set to upscaled reference patch. The latent high-resolution patch is then gradually updated by minimizing the difference between a predicted low-resolution patch and an observed low-resolution patch. The image warping $\mathcal{W}$ to predict low-resolution patch can be given, or can be estimated simultaneously with the high-resolution patch. In the proposed method, the image warping $\mathcal{W}$ is simultaneously estimated with respect to camera motion and landmark pose in a unified filtering framework.

3.3 Simultaneous Landmark Pose and High-Resolution Patch Estimation

Based on the particle filtering-based SLAM using locally planar landmarks [22], an EKF-based 6-DOF landmark pose estimation with the back-projection [48] is pro-

posed to estimate a high resolution image for landmark templates. The state vector of EKF is composed of 6-DOF landmark poses and high resolution templates for registered landmarks, and they are updated using incoming input images. Figure 3.2 illustrates steps for EKF-based image super-resolution and its example images. The state vector of high resolution landmark template, which is initially constructed by upscaling an input image, is updated in the EKF formulation after the data association of locally planar landmarks using the warping-based image registration method. The 6-DOF landmark pose is simultaneously updated by EKF, and it helps the super-resolution of template be more accurate. The image registration is critical part of reconstruction-based super-resolution. The 3D geometry from SLAM provides good initial solution for the 2D image-based fine registration of landmark template which can be trapped in local optima with bad initial solution.

3.3.1 Particle filtering framework for simultaneous landmark pose and high-resolution patch estimation

The proposed EKF-based image super-resolution is combined with the camera pose estimation via Rao-Blackwellized particle filter (RBPF) framework [21,53,54]. There are three unknowns that have to be estimated using this framework: the camera trajectory $\mathbf{P}_{1:t}$ composed of 6-DOF camera pose $\mathbf{P}_k$ at each time step t, the set of 6-DOF landmark poses $\mathbf{L} = \{L^1, ..., L^m, ..., L^M\}$, and the high-resolution patches $\mathbf{T}^H = \{\mathcal{T}^{H,1}, ..., \mathcal{T}^{H,m}, ..., \mathcal{T}^{H,M}\}$ for each landmark, where m denotes the landmark index and M is the number of registered landmarks in the map. The posterior probability for these variables $p(\mathbf{P}_k)$ given with a set of observation $\mathbf{y}_{1:k}$ can be

factorized as follows:

$$\begin{aligned} & p(\mathbf{P}_{1:k}, \mathbf{L}, \mathbf{T}^H | \mathbf{y}_{1:k}) \\ =& p(\mathbf{L}, \mathbf{T}^H | \mathbf{P}_{1:k}, \mathbf{y}_{1:k}) p(\mathbf{P}_{1:k} | \mathbf{y}_{1:k}) \\ =& p(\mathbf{T}^H | \mathbf{L}, \mathbf{P}_{1:k}, \mathbf{y}_{1:k}) p(\mathbf{L} | \mathbf{P}_{1:k}, \mathbf{y}_{1:k}) p(\mathbf{P}_{1:k} | \mathbf{y}_{1:k}) \\ =& \prod_M p(\mathcal{T}^H | L, \mathbf{P}_{1:k}, \mathbf{y}_{1:k}) \prod_M p(L_m | \mathbf{P}_{1:k}, \mathbf{y}_{1:k}) p(\mathbf{P}_{1:k} | \mathbf{y}_{1:k}). \end{aligned} \tag{3.3}$$

In RBPF, the probability distribution of the camera trajectory $\mathbf{P}_{1:k}$ and the landmark positions $\mathbf{L}$ are factorized and the factorized two distributions are estimated by particle filter and kalman filter, respectively. Similarly, each probability distribution in Equation (3.3) is estimated by different method depending on its characteristics. The probability distribution of camera trajectory $p(\mathbf{P}_{1:k}|\mathbf{y}_{1:k})$ is approximated by samples and their weights from importance sampling and resampling of particle filter on the manifold of Lie group, because direct calculation of the distribution is intractable. The probability distribution of landmark poses $\mathbf{L}$ are individually estimated by unscented Kalman filter [55] with respect to each samples from particle filter. Except high-resolution patch estimation, the importance sampling of camera pose and unscented Kalman filter estimation for landmark poses follow that of [22].

3.3.2 Kalman filter-based high-resolution patch estimation

For the high-resolution patch estimation by Kalman filter, the state vector is formed by representing the high-resolution patch $\mathbf{T}^H$ as a vector form. The initial value of the state vector is obtained by upscaling the landmark patch using bicubic interpolation from the image where the landmark is registered. The covariance matrix $\mathbf{C}$ of Kalman filter is set to be diagonal, which means that each pixel in the high-resolution patch is estimated independently, to save computational cost. The initial

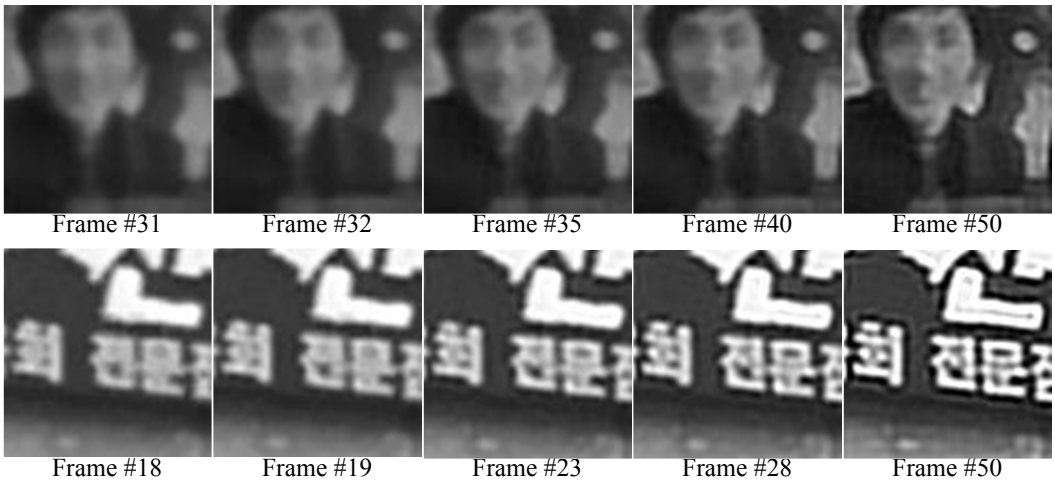


Figure 3.3: High-resolution updates of landmark templates for selected frames. The leftmost templates correspond to initial states obtained by upscaling the original template with bicubic interpolation.

values of covariance matrix is equally given by constant σ_0^2.

After sampling camera pose $\mathbf{P}_k$ and updating landmark poses $\mathbf{L}$, a single update for probability distribution of each high-resolution patch estimation $p(\mathcal{T}^H|L, \mathbf{P}_{1:k}, \mathbf{y}_{1:k})$ is performed. The prediction of low-resolution patch $\hat{\mathcal{T}}_k^L$ is generated from the current estimate of high-resolution patch $\mathcal{T}^H$ using Equation (3.1). Similarly to the approximation in [56], the Jacobian matrix for Kalman filter update can be simplified by the resampling weight $w(\mathcal{T}^H, \mathcal{T}^L)$ which can be obtained from the mapping function by Equation (3.1). The jth element of state vector and covariance matrix are then updated using the measurement noise covariance $\mathbf{C}_n$ which is a diagonal

matrix having the constant elements $\mathbf{C}_n(j,j) = \sigma_n^2$ for all j as follows:

$$\begin{aligned} K(j) &= \mathbf{C}(j,j)w_j/(\mathbf{C}_n(j,j) + \sum_j \mathbf{C}(j,j)w_j^2) \\ \mathcal{T}_j^H &\leftarrow \mathcal{T}_j^H + K(j)(R_j) \\ \mathbf{C}(j,j) &\leftarrow \mathbf{C}(j,j)(1 - w_j K(j)), \end{aligned} \tag{3.4}$$

where K_j is the jth element of Kalman gain matrix, and R_j is the reconstruction error of the jth element of the state vector for the kth observation patch, given by Equation (3.2). For each time step k, a single Kalman filter update using Equation (3.4) is performed, and the latent high-resolution patch is gradually obtained. The covariance matrix is regarded as converged when the mean of its diagonal elements is below a predefined threshold, and then no more Kalman filter update is performed. Figure 3.3 shows examples of gradually updated high-resolution templates.

3.4 Experiments

In the experiment, both the SLAM performance and the super-resolution performance of the proposed method are evaluated. The improvement of SLAM performance is evaluated by enabling and disabling the super-resolution part, and the super-resolution performance is compare with other image-based method.

3.4.1 Improvement of SLAM performance

The SLAM performance can be evaluated by measuring the accuracy of camera pose and landmark pose estimation, but it is difficult to obtain the ground truth of both estimation. Therefore, the average error of landmark projection is measured

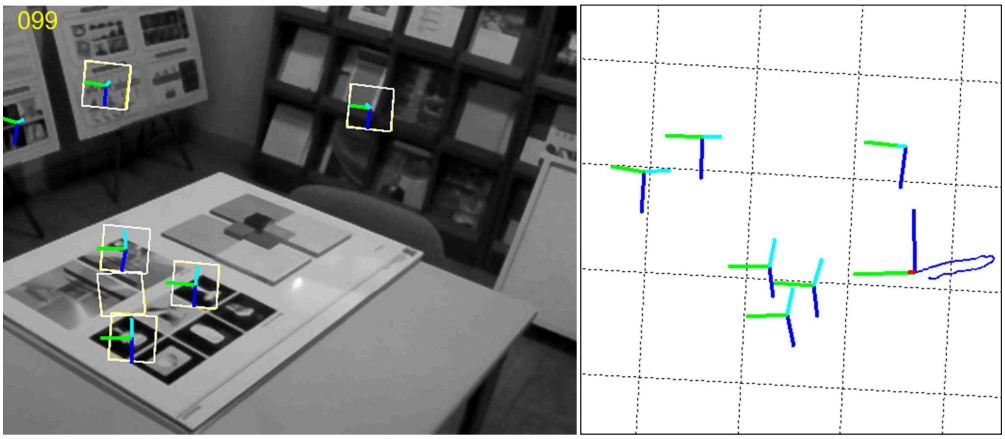


Figure 3.4: 3D reconstruction result by the proposed method. Left: Input images shown with estimated landmark poses. Right: Estimated camera trajectory and landmark poses in 3D map.

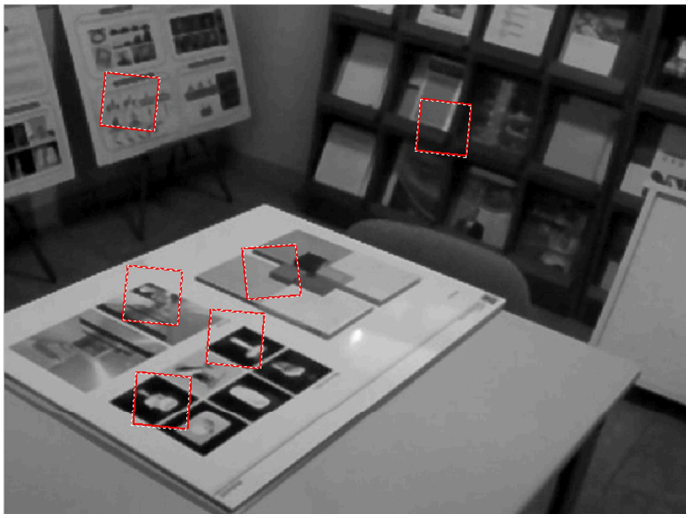


Figure 3.5: Projected landmarks after camera pose estimation (white dotted lines) and observed landmarks (red solid rectangles), which indicate the accuracy of SLAM results indirectly.

to evaluate the SLAM performance of the proposed method indirectly. Different from the reprojection error of landmarks for point-based 3D reconstruction, the projection error of planar landmark can be measured more precisely by employing 2D Homography. Given with four corner points of projected landmark and observed patch in the image, 2D Homography corresponds to those points is calculated and the measurement error can be defined by distance between the calculated Homography and identity Homography in the manifold of the special linear group $SL(3)$ [22]. If the camera pose and landmark poses are accurately estimated, then the error of landmark projection should be small. Figure 3.5 shows the example of projected landmarks (white solid rectangles) and their observed patches (red dotted rectangles) with and without the proposed super-resolution method, and their corresponding measurement error values. To project the landmarks onto images, the sample mean of camera pose and landmark poses are used in the particle filter framework.

The average landmark projection errors for an example sequence with and without the landmark patch super-resolution are plot in Figure 3.6. Initially, the error values of two methods are almost same because the resolution of landmark patches is not sufficiently enhanced. However, as more observations are incorporated for super-resolution, more accurate estimation of camera and landmark poses is possible and the average landmark projection error is reduced with the proposed method.

3.4.2 Super-resolution quality

The super-resolution results by the proposed method for indoor and outdoor image sequence are presented in Figure 3.7 and Figure 3.8, respectively. It is shown that details of target landmarks are recovered, and the sharpness of landmark textures are improved. The sharpness of landmark texture is important in data associa-

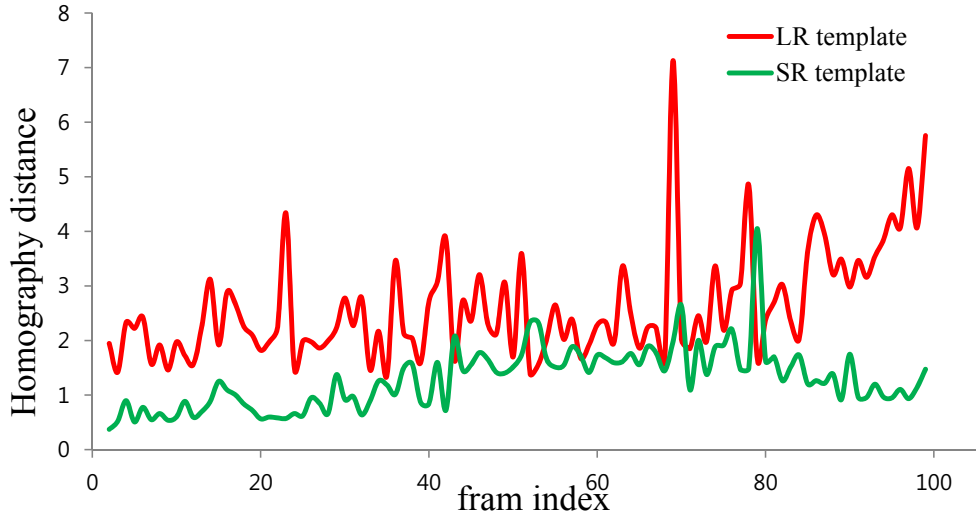


Figure 3.6: Plot of the average landmark projection error with and without the proposed super-resolution.

tion process because sharp edges in texture contribute to accurate registration of landmark patches.

The similarity between patches in terms of normalized cross correlation (NCC) is compared in Figure 3.9. For real image data, the accuracy of data association of landmark is difficult to measure because its ground truth is not available. Therefore, the accuracy is indirectly measured by using the similarity between a landmark template and observed patches. The super-resolution template is the most probable appearance of real landmark, thus the effect of pixel noise and quantization is reduced in data association with super-resolution template.

Figure 3.7: Super-resolution results for *building* sequence. Left: Low-resolution patches tracked in input images. Right: Super-resolution patches (×3) by the proposed method.

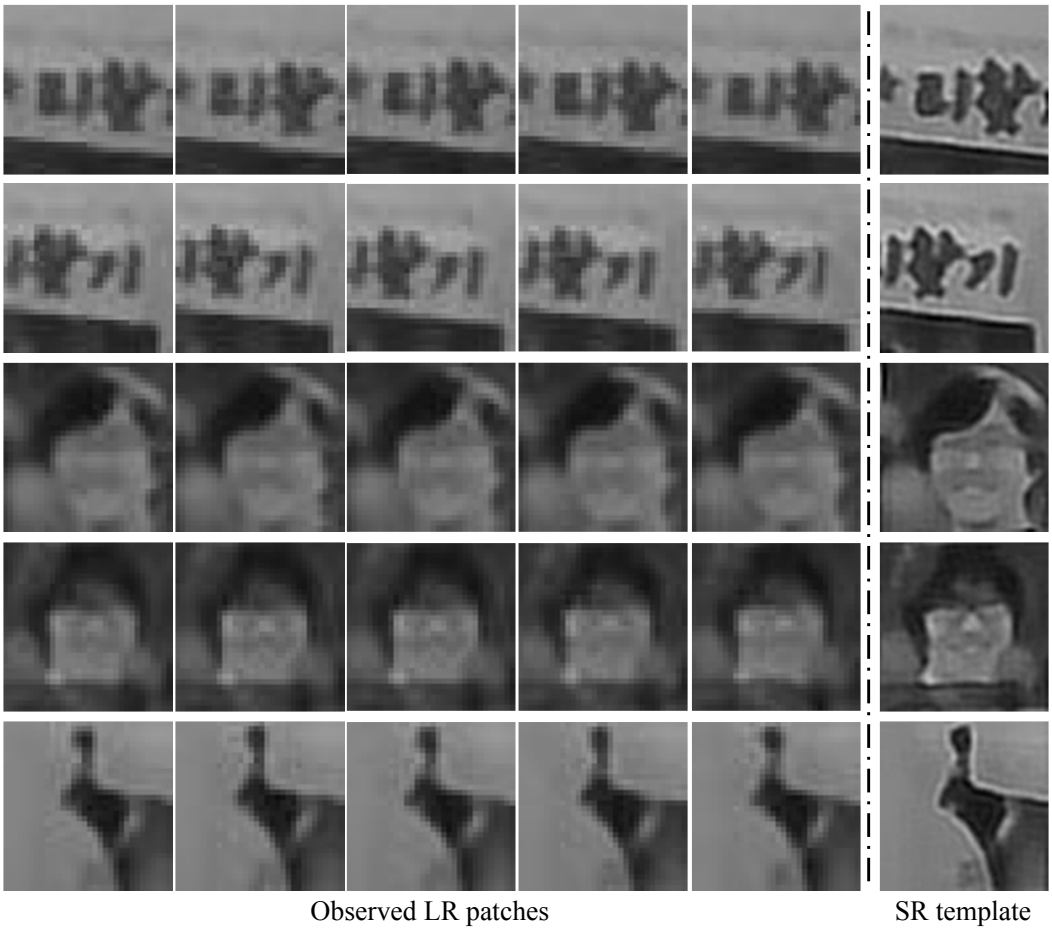


Figure 3.8: Super-resolution results for *poster* sequence. Left: Low-resolution patches tracked in input images. Right: Super-resolution patches ($\times 3$) by the proposed method.

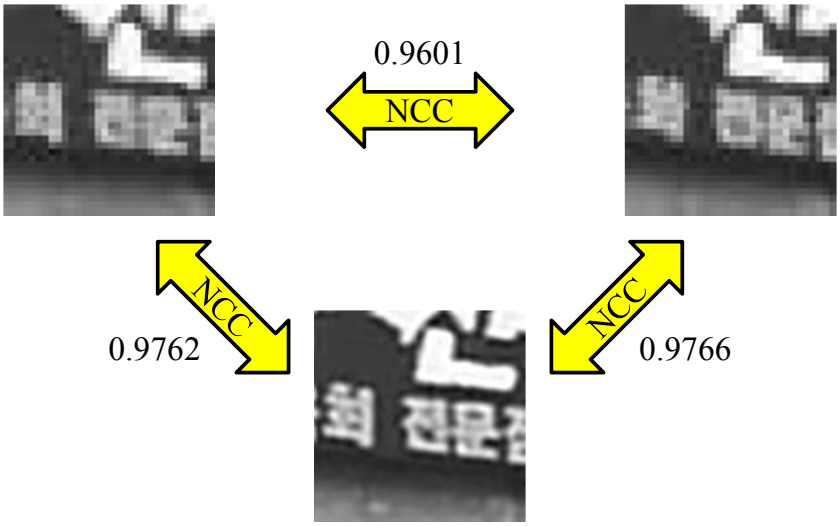


Figure 3.9: Similarity between landmark patches. The super-resolution patches provide higher NCC measures than low-resolution patches.

3.5 Summary

In this chapter, the sparse point-based 3D reconstruction is combined with super-resolution of landmark patch. The problem of low-resolution of landmark patches in the sparse point-based 3D reconstruction is overcome by explicitly enhancing the resolution of landmark patches in the unified particle filtering framework. With the help of 3D geometry the super-resolution landmark templates are easily obtained, and reversely the super-resolution templates contribute to accurate 3D reconstruction results.

Chapter 4

Dense 3D Reconstruction and Image Deblurring

4.1 Introduction

Motion blur in images is an undesirable effect in various computer vision algorithms. In particular, motion blur is a critical issue in the correspondence problem because motion blur destroys the structure details of images. Consequently, numerous algorithms that rely on pixel correspondence, such as optical flow, are severely affected by motion blur.

The pixel correspondence is also important problem in the image-based 3D reconstruction algorithms, *e.g.*, stereo reconstruction and structure from motion. Among these reconstruction algorithms, dense reconstruction algorithms [57–60], which reconstruct dense 3D structures from a single moving camera, frequently suffer from severe motion blur due to camera shakes because the camera keeps moving by human hands or mobile robots. To estimate primitive depth maps for full surface recon-

struction, pixel correspondences for two or more images have to be estimated with high accuracy. However, motion blur degrades the resolution of input images in a blurred direction, and classical dense correspondence algorithms based on brightness or gradient constancy fail to obtain correct pixel correspondences.

To handle motion blur for 3D reconstruction, deblurring methods, particularly video deblurring [6, 61–63], can be used by recovering input images. However, most high-quality deblurring methods are inadequate for fast dense reconstruction systems, because these methods typically entail high computational cost but cannot handle scene-depth variation in blur kernel estimation. Therefore, a blur-handling method for 3D reconstruction is proposed, in which blur kernel and depth of pixel are simultaneously estimated by adopting their dependency on each other.

A blur kernel from camera shake can be interpreted as a trajectory of a projected 3D scene point by camera motion during exposure time. Thus, the pixel-wise blur kernel can be determined in a closed form when camera motion, exposure time, and scene depth are given. In other words, estimating the scene depth is equivalent to estimating the pixel-wise blur kernel when camera motion and exposure time are known. These values are available in general dense reconstruction systems, where exposure time can be obtained from camera hardware and camera motion can be estimated by camera localization method.

In the proposed method, camera motion is estimated by image registration method between a reference image and an warped observed image using a reconstructed depth map, similarly to other 3D reconstruction algorithms [59]. Although the estimated camera motion has errors, the proposed method can generate a more reliable depth map than the conventional depth reconstruction methods that do not consider motion blur, as compared in Figure 4.1 (b, c). The estimated depth map

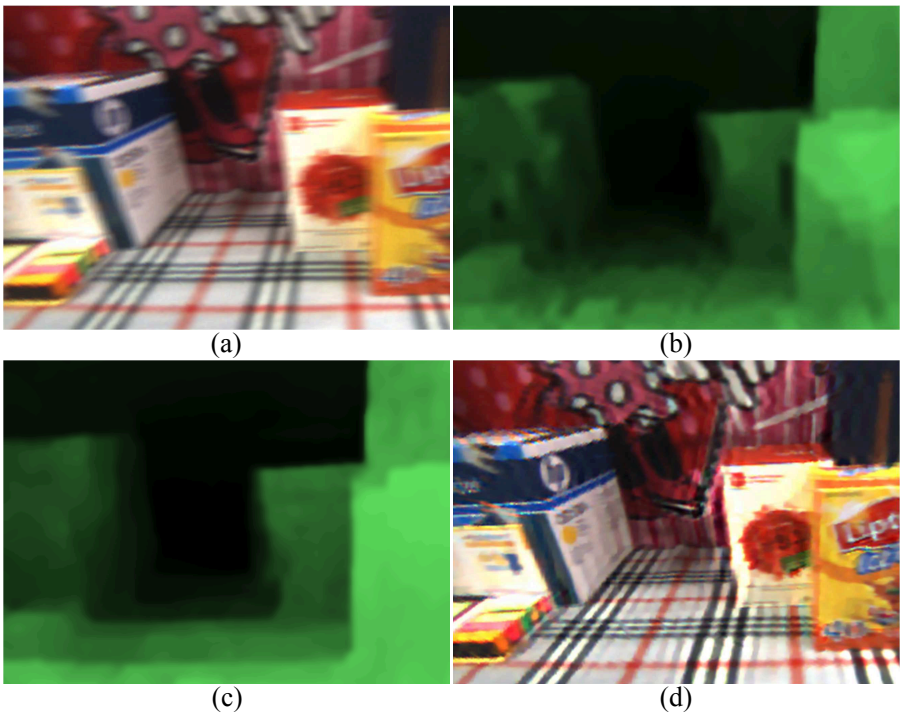


Figure 4.1: Depth reconstruction from five blurry images: (a) Sample from real input images. (b) Result of the conventional variational depth reconstruction. (c) Result of the proposed blur-aware depth reconstruction. (d) Deblurred image by using the estimated depth-dependent blur kernel.

can be converted into pixel-wise blur kernels by using 3D geometry, and non-uniform deblurring can then be easily achieved, as shown in Figure 4.1 (d). The proposed blur kernel estimation explicitly considers scene depth, thus it can provide improved deblurring results compared with previous image or video deblurring methods that disregard scene depth variation.

4.2 3D Geometry and Deblurring

Motion blur from camera shakes, rather than from object motion, has been solved in many studies by considering camera geometry. However, few methods utilize both camera geometry and scene geometry, *i.e.*, scene depth. This means that most methods that utilize camera geometry disregard the effect of scene depth variation. The related studies on blur kernel estimation utilizing either camera geometry or scene geometry are briefly reviewed here.

Camera motion and motion blur. The relationship between the camera geometry and motion blur has been studied in multiple image deblurring [62,64] and single image deblurring [3, 28, 65] to address a method for removing non-uniform motion blur attributed to camera shakes. In multiple image deblurring, camera motion is parameterized by homography under the assumption of constant scene depth, and blur kernels are derived from the estimated homographies. In single image deblurring, non-uniform motion blur is represented by a finite number of basis functions that related to camera motion or homography, and blur kernel is solved efficiently with respect to these basis functions. However, the above methods do not consider the effect of scene depth variation, which is an important factor that contribute to

the non-uniformity of motion blur.

Joshi *et al.* [66] explicitly utilize a camera motion by estimating the camera motion from inertial measurement sensors. Their camera is equipped with accelerometers and gyroscopes, and six degrees of freedom (DOF) camera motion is estimated from the sensors and it generates accurate non-uniform blur kernels. While typical image only-based blur estimation methods have limited range of measurable kernel size because they utilize image priors which are valid only for a small region, [66] can handle large size of blur kernels with the aid of additional sensors. The limitation of this method is that it also assumes uniform scene depth. Thus, this method is valid only for negligible depth variation or limited types of camera motion, such as pure rotation.

Scene depth and motion blur. To address the depth variation in blur kernel estimation, Xu and Jia [67] combined depth reconstruction by using stereopsis with blur kernel estimation. Since motion blur in stereo image pair is almost identical, a scene depth is easily estimated by classical stereo matching algorithm and the result is used in their depth-dependent blur kernel estimation. Their depth-dependent blur kernel estimation can be extended to single image deblurring, however, camera motion is limited to translation in single image cases.

In-depth studies on the relationship between scene depth and motion blur were conducted in [30, 68], which are closely related to the proposed method. These methods use two or more images in estimating scene depth and recovering deblurred images. However, these methods differ from our method; [30] assumes sideways translational camera motion unlike the proposed method which deals with arbitrary camera motion, a reference unblurred image is required in [68] while all input images

can be blurred in the proposed method.

The proposed method considers both camera motion and the effect of scene depth variation in handling motion blur. Although the proposed method has limited applications because it requires multiple input images for camera motion estimation in 3D reconstruction, the method has advantages of both handling large blur size in [66] and handling depth variation in [67] without requiring additional inertial sensors nor a stereo camera.

4.3 Blur-Aware Depth Reconstruction

Two image blur kernel estimation problem is converted into a depth estimation problem by utilizing camera motion obtained from camera localization algorithm in 3D reconstruction. This section explains the two image motion blur estimation strategy and then presents a method that converts the blur kernel estimation problem into a depth estimation problem. Finally, the two image depth reconstruction process will be extended to multiple image depth reconstruction.

4.3.1 Motion blur estimation from two images

Compared to the single image-based blur kernel estimation, blur kernel estimation using two or more images has the advantage that it can utilize a motion estimation across images, *i.e.*, optical flow. Optical flow algorithms, however, cannot be directly used for blur kernel estimation due to the following differences between optical flow and pixel's motion path in motion blur. First, optical flow provides only starting point and end point of a pixel at two images, whereas pixel's path in motion blur

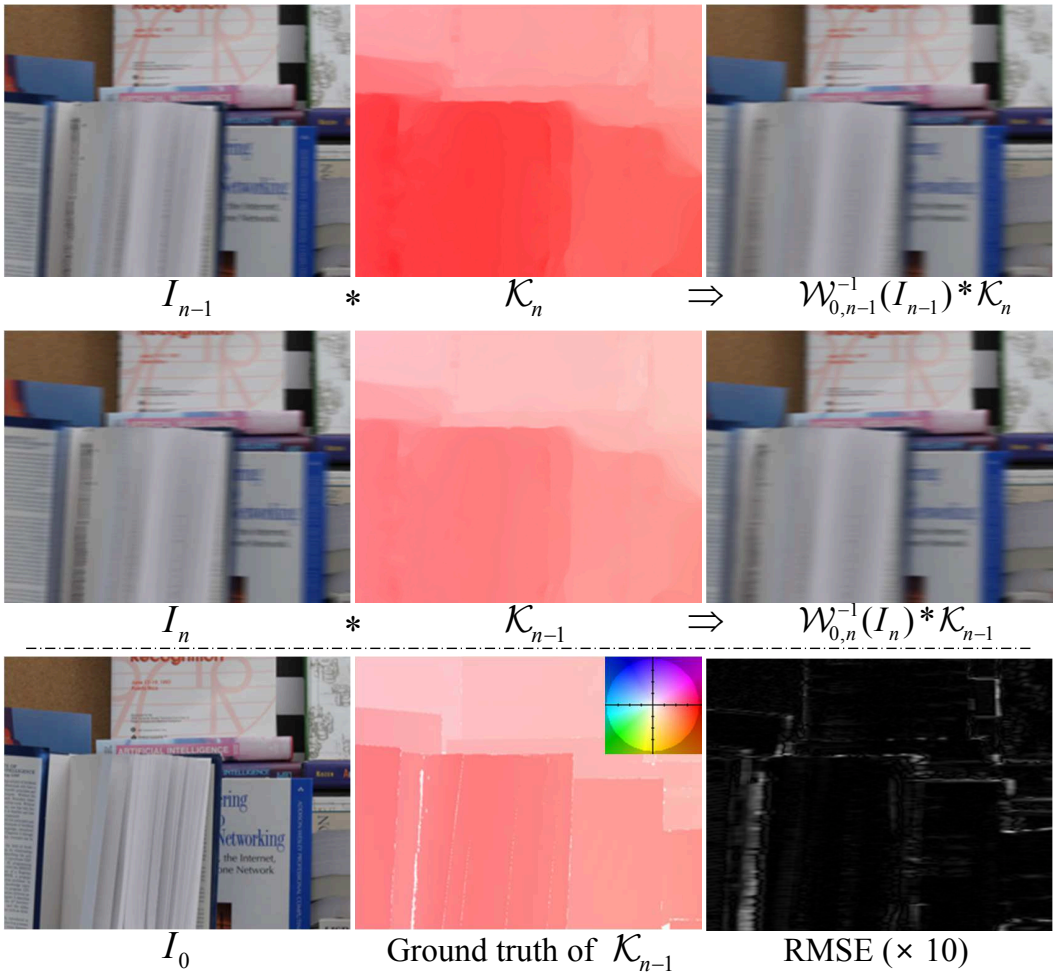


Figure 4.2: Commutative property of blur kernels. Top and middle: Synthesized input images I_{n-1} and I_n, the estimated blur kernels represented by motion vectors, and their commutative convolution results. Bottom: Unblurred reference image, ground truth motion vectors of I_{n-1} with a color map, and the root-mean-square (RMS) error between $\mathcal{W}_{0,n-1}^{-1}(I_{n-1}) * \mathcal{K}_n$ and $\mathcal{W}_{0,n}^{-1}(I_n) * \mathcal{K}_{n-1}$ scaled by 10.

contains full intermediate trajectory of the pixel. Second, motion blur contains only pixel's motion during exposure time, but optical flow is independent from exposure time. Therefore, we should carefully estimate a pixel's motion between two images with the consideration of motion blur to get the blur kernel of pixel.

Estimation of motion blur kernels from two images utilizes the idea that applying the blur kernel of each image to the other image results in the same cumulatively blurred images [4, 69]. Let I_{n-1} and I_n be two consecutive blurred images in an observed sequence, which have latent unblurred images L_{n-1} and L_n, as well as blur kernels $\mathcal{K}_{n-1}$ and $\mathcal{K}_n$, respectively. The blurred image by the pixel-wise blur kernel $\mathcal{K}_n(x, y)$ is represented as follows:

$$I_n(x, y) = (L_n * \mathcal{K}_n(x, y))(x, y), \tag{4.1}$$

where $*$ denotes the convolution operator that corresponds to blur operation, and (x, y) represents a pixel coordinate. If there is no confusion, then the pixel coordinate notation (x, y) for images I_n and L_n as well as the blur kernel $\mathcal{K}_n$ is omitted for notational simplicity. For the two blur kernels $\mathcal{K}_{n-1}$ and $\mathcal{K}_n$ as well as the reference unblurred image L_0, the following equality should hold by the commutative property of convolution:

$$L_0 * \mathcal{K}_{n-1} * \mathcal{K}_n = L_0 * \mathcal{K}_n * \mathcal{K}_{n-1}, \tag{4.2}$$

and it gives

$$\mathcal{W}_{0,n-1}^{-1}(I_{n-1}) * \mathcal{K}_n = \mathcal{W}_{0,n}^{-1}(I_n) * \mathcal{K}_{n-1}, \tag{4.3}$$

where $\mathcal{W}_{0,n}$ is the image warping function such that $L_n = \mathcal{W}_{0,n}(L_0)$. An example of estimated blur kernels and their convolution results are illustrated in Figure 4.2. Based on Equation (4.3), we can derive the objective function to determine the correct values of $\mathcal{K}_{n-1}$ and $\mathcal{K}_n$.

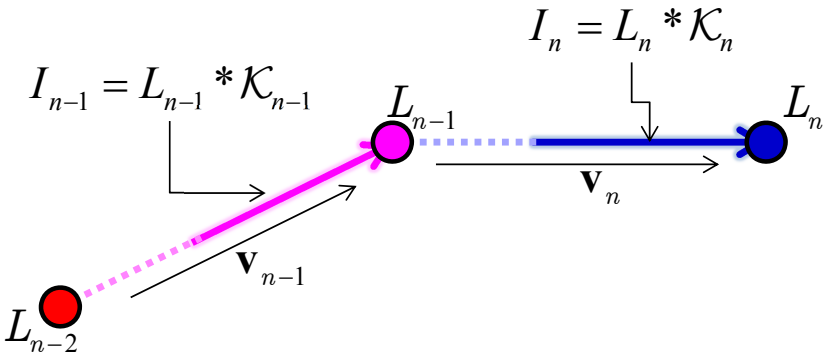


Figure 4.3: Proposed motion blur model: The colored dots represent the pixel positions of a 3D scene point $\mathbf{X}$ for each time n, and the intensities at these positions are represented by L. The convolution of pixel intensities along with the thick arrows corresponds to the blurred kernels $\mathcal{K}$ which results in the blurred intensity I. The blur kernel $\mathcal{K}$ corresponds to a part of pixel motion $\mathbf{v}$ in an exposure time.

There are four unknowns, $\mathcal{W}_{0,n-1}$, $\mathcal{W}_{0,n}$, $\mathcal{K}_{n-1}$, and $\mathcal{K}_n$ in Equation (4.3), but the dependency of blur kernel on the warping functions can reduce the number of actual unknowns. Let $\mathbf{v}_n = [u_n, v_n]^\top$ be the 2D motion vector that corresponds to the warping function $\mathcal{W}_{n-1,n}$, such that $L_n = \mathcal{W}_{n-1,n}(L_{n-1}) \equiv L_{n-1}(x+u_n, y+v_n)$. Without motion blur, the warped image L_n from L_{n-1} by a small motion $\mathbf{v}_n$ can be approximated by the second-order expansion [44]:

$$\begin{aligned} L_n &= \mathcal{W}_{n-1,n}(L_{n-1}) \\ &\approx L_{n-1} + \mathcal{J}_{L_{n-1}}\mathbf{v}_n + \frac{1}{2}\mathbf{v}_n^\top \mathcal{H}_{L_{n-1}}\mathbf{v}_n, \end{aligned} \tag{4.4}$$

where the matrices $\mathcal{J}_{L_{n-1}}$ and $\mathcal{H}_{L_{n-1}}$ represent the Jacobian and Hessian matrices, respectively, for the image L_{n-1} with respect to the x and y axes.

When motion blur is considered in the image warping between two images as shown in Figure 4.3, the blurred and warped image I_n from L_{n-1} is approximated

with additional coefficients as follows [37, 42]:

$$\begin{aligned} I_n &= \mathcal{W}_{n-1,n}(L_{n-1}) * \mathcal{K}_n \\ &\approx L_{n-1} + a\mathcal{J}_{L_{n-1}}\mathbf{v}_n + \frac{1}{2}b\mathbf{v}_n^\top \mathcal{H}_{L_{n-1}}\mathbf{v}_n. \end{aligned} \tag{4.5}$$

The coefficients a and b are determined by the exposure time τ,

$$a = \frac{\tau_o + \tau_c}{2\tau_c}, \quad b = \frac{\tau_c^2 + \tau_c\tau_o + \tau_o^2}{3\tau_c^2}, \tag{4.6}$$

where τ_o and τ_c denote open and close time of the camera shutter, respectively. Time $\tau = 0$ in capturing the image I_n corresponds to time $\tau = \tau_c$ in capturing the previous image I_{n-1}. If the exposure time is infinitesimal, then $\tau_o = \tau_c$ holds, and Equation (4.5) is equivalent to Equation (4.4). Reference [42] provides the detailed derivation of this approximation.

As shown in Equation (4.5), which represents the parametrization of the blurred image I by using motion vector $\mathbf{v}$, the objective function that satisfies condition (4.3) can be formulated by using only the motion vectors $\mathbf{v}_{n-1}$ and $\mathbf{v}_n$. First the objective function is formulated from Equation (4.3),

$$\underset{\mathcal{W}_{n-1},\mathcal{W}_n,\mathcal{K}_{n-1},\mathcal{K}_n}{\arg\min} \left\| \mathcal{W}_{0,n-1}^{-1}(I_{n-1}) * \mathcal{K}_n - \mathcal{W}_{0,n}^{-1}(I_n) * \mathcal{K}_{n-1} \right\|_1. \tag{4.7}$$

By substituting Equation (4.5) into Equation (4.7) for both I_{n-1} and I_n, we can obtain the objective function with respect to $\mathbf{v}_{n-1}$ and $\mathbf{v}_n$. First, the warping function $\mathcal{W}_{0,n}$ is applied to both sides of Equation (4.3) for simplification, which yields

$$\begin{aligned} &\mathcal{W}_{0,n}(\mathcal{W}_{0,n-1}^{-1}(I_{n-1})) * \mathcal{K}_n = \mathcal{W}_{0,n}(\mathcal{W}_{0,n}^{-1}(I_n)) * \mathcal{K}_{n-1} \\ &\Rightarrow \mathcal{W}_{n-1,n}(I_{n-1}) * \mathcal{K}_n = I_n * \mathcal{K}_{n-1}. \end{aligned} \tag{4.8}$$

From the approximation of Equation (4.5) to the left-hand side of Equation (4.8) up to the first-order, we have

$$\mathcal{W}_{n-1,n}(I_{n-1}) * \mathcal{K}_n \approx I_{n-1} + a\mathcal{J}_{I_{n-1}}\mathbf{v}_n. \tag{4.9}$$

Similarly, applying Equation (4.5) to the right-hand side of Equation (4.8) yields

$$\begin{aligned} I_n * \mathcal{K}_{n-1} &= \mathcal{W}^{-1}_{n-2,n-1}(\mathcal{W}_{n-2,n-1}(I_n) * \mathcal{K}_{n-1}) \\ &\approx \mathcal{W}^{-1}_{n-2,n-1}(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1}), \end{aligned} \tag{4.10}$$

and by Equation (4.4), we have

$$\begin{aligned} &\mathcal{W}^{-1}_{n-2,n-1}(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1}) \\ \approx &(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1}) - \mathcal{J}_{(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1})}\mathbf{v}_{n-1} \\ = &I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1} - (\mathcal{J}_{I_n} + a\mathbf{v}^\top_{n-1}\mathcal{H}_{I_n}\mathbf{v}_{n-1}) \\ = &I_n + (a-1)\mathcal{J}_{I_n}\mathbf{v}_{n-1} - a\mathbf{v}^\top_{n-1}\mathcal{H}_{I_n}\mathbf{v}_{n-1}. \end{aligned} \tag{4.11}$$

By subtracting the two terms, the objective function is obtained as follows:

$$\arg\min_{\mathbf{v}_{n-1},\mathbf{v}_n} \|(I_{n-1} + a\mathcal{J}_{I_{n-1}}\mathbf{v}_n) - (I_n + (a-1)\mathcal{J}_{I_n}\mathbf{v}_{n-1} - a\mathbf{v}^\top_{n-1}\mathcal{H}_{I_n}\mathbf{v}_{n-1})\|_1. \tag{4.12}$$

The first term in Equation (4.12) approximates the blurred appearance of I_{n-1} by the blur kernel of I_n, and the second term approximates the warped and blurred appearance of I_n, by warping $\mathcal{W}^{-1}_{n-1,n}$ and the blur kernel of I_{n-1}, respectively.

4.3.2 Motion blur estimation to depth estimation

Although the objective function is reduced to determining pixel-wise motion vectors $\mathbf{v}_{n-1}$ and $\mathbf{v}_n$, this problem remains ill-posed because only one pixel correspondence is given for the quadratic equation (4.12) of two variables. Therefore, an additional constraint has to be incorporated to eliminate the ambiguities in $\mathbf{v}_{n-1}$ and $\mathbf{v}_n$. The

ambiguities in motion or blur kernel estimation given two images has been addressed in several previous works [4, 40, 69]. For example, the directions of the blur kernels of two images are assumed to be known [69]; otherwise, additional input images are used to refine the motion vectors of the two blurry images [4]. The proposed method utilizes a camera motion and exposure time as additional constraints to resolve the ambiguity in motion estimation. The use of camera motion has a similar advantage as that of using known blur directions in [69]. However, the assumption of known camera motion is more general than the assumption of known blur direction because the former can address non-uniform blur kernels and any type of pixel motion, such as curved pixel motion caused by camera rotation.

When camera motion and exposure time are known, the estimation of pixel-wise blur kernels from two images is converted into an estimation of pixel-wise depth value. In the proposed method, exposure time τ_o and τ_c are provided by camera hardware, and camera pose at $\tau = \tau_c$ is obtained from the registration-based camera localization algorithm. Let $\mathbf{P}_n^\tau \in \mathbb{SE}(3)$ be the six DOF camera pose at time τ for the nth image, which is represented by the special Euclidean group in three dimensions, and let d be the inverse depth of pixel (the pixel coordinate notation is also omitted for simplicity) with respect to the unblurred reference image L_0. Inverse depth, which is a reciprocal of depth, is used in the proposed model because inverse depth has better convergence property in estimation than the original depth [70].

The 2D motion path of the projected pixel point (x_n^τ, y_n^τ) at time τ corresponding to inverse depth d is represented as follows:

$$\begin{aligned} (x_n^\tau, y_n^\tau) &= h(\mathbf{K}((\mathbf{P}_n^\tau)^{-1} \cdot \mathbf{X})), \\ \mathbf{X} &= \frac{1}{d}\mathbf{K}^{-1} \cdot (x, y, 1)^\top, \end{aligned} \tag{4.13}$$

where $h(\cdot)$ is the dehomogenization function, such that $h((x,y,z)^\top) = (x/z, y/z)$, $\mathbf{K}$ is the camera intrinsic matrix, and $\mathbf{X}$ is a 3D scene point corresponding to pixel (x,y) at the reference image. The product of inverse camera pose $\mathbf{P}_t^{-1}$ and the 3D scene point $\mathbf{X}$ is defined as follows:

$$(\mathbf{P}_n^\tau)^{-1} \cdot \mathbf{X} = (\mathbf{R}_n^\tau)^\top \mathbf{X} - (\mathbf{R}_n^\tau)^\top \mathbf{T}_n^\tau, \tag{4.14}$$

where $\mathbf{R}_n^\tau$ and $\mathbf{T}_n^\tau$ are camera rotation and translation, respectively. Equation (4.13) shows that the blur kernel $\mathcal{K}$ in Equation (4.3) can be calculated by using 3D geometric quantities only. Thus, the kernel estimation problem is reformulated into an estimation problem of inverse depth d.

Equation (4.13) shows that the pixel motions $\mathbf{v}_{n-1} = (x_{n-1}^{\tau_c}, y_{n-1}^{\tau_c}) - (x_{n-2}^{\tau_c}, y_{n-2}^{\tau_c})$ and $\mathbf{v}_n = (x_n^{\tau_c}, y_n^{\tau_c}) - (x_{n-1}^{\tau_c}, y_{n-1}^{\tau_c})$ are functions of inverse depth d. The objective function with respect to d can be derived by substituting Equation (4.13) into the original objective function (4.12). To solve the objective function by means of the convex optimization framework, the relationship between the pixel motions $\mathbf{v}_{n-1}$, $\mathbf{v}_n$ and a small update value of depth Δd is linearized using the Jacobian matrices $\mathcal{J}_{\mathbf{v}_{n-1}} = \left[\frac{\partial v_{n-1}}{\partial d} \frac{\partial u_{n-1}}{\partial d}\right]^\top$ and $\mathcal{J}_{\mathbf{v}_n} = \left[\frac{\partial v_n}{\partial d} \frac{\partial u_n}{\partial d}\right]^\top$ as:

$$\begin{aligned} \mathbf{v}_{n-1} &= \mathcal{J}_{\mathbf{v}_{n-1}} \Delta d = \mathcal{J}_{\mathbf{v}_{n-1}} (d - \bar{d}), \\ \mathbf{v}_n &= \mathcal{J}_{\mathbf{v}_n} \Delta d = \mathcal{J}_{\mathbf{v}_n} (d - \bar{d}). \end{aligned} \tag{4.15}$$

where $\bar{d}$ is an initial estimate of d. The objective function with respect to d is derived from Equation (4.12) and Equation (4.15) as follows:

$$\begin{aligned} \arg\min_d \| I_{n-1} - I_n + \{a \mathcal{J}_{I_{n-1}} \mathcal{J}_{\mathbf{v}_n} + (1-a) \mathcal{J}_{I_n} \mathcal{J}_{\mathbf{v}_{n-1}}\}(d - \bar{d}) \\ + \{a \mathcal{J}_{\mathbf{v}_{n-1}}^\top \mathcal{H}_{I_n} \mathcal{J}_{\mathbf{v}_{n-1}}\}(d - \bar{d})^2 \|_1. \end{aligned} \tag{4.16}$$

Therefore, the motion blur estimation problem is now represented by the depth estimation problem.

4.3.3 Depth reconstruction using multiple images

The proposed two view depth reconstruction can be easily extended to multiple view depth reconstruction in a manner similar to that of other multiple image reconstruction methods [59, 60]. The use of multiple images provides more accurate depth results by mitigating the effect of image noise. The objective function for the depth reconstruction of multiple images is defined as the minimization of the sum of the differences between the first image I_1 and the other images I_n considering their blurred appearances.

Given that Equation (4.16) is valid only with consecutive image indices $n-1$ and n, we should modify Equation (4.16) to define the differences between the first image I_1 and other images I_n for $n \neq 2$. To this end, the first image I_1 is warped to simulate the $(n-1)$th image I_{n-1}, such that $I'_{n-1} = \mathcal{W}_{1,n-1}(I_1)$. The warping function $\mathcal{W}_{1,n-1}$ is calculated by projecting and reprojecting the pixel of the first image by using Equation (4.13). We can then replace I_{n-1} in Equation (4.16) with I'_{n-1} and replace $\mathbf{v}_{n-1}$ with $\mathbf{v}_1$. By summing the differences of all image pairs, we can obtain the following objective function for multiple image depth reconstruction:

$$\begin{aligned}\arg\min_{d} \sum_{n=2}^{N} \| I'_{n-1} - I_n + \{a\mathcal{J}_{I'_{n-1}}\mathcal{J}_{\mathbf{v}_n} + (1-a)\mathcal{J}_{I_n}\mathcal{J}_{\mathbf{v}_1}\}(d-\bar{d}) \\ + \{a\mathcal{J}_{\mathbf{v}_1}^{\top}\mathcal{H}_{I_n}\mathcal{J}_{\mathbf{v}_1}\}(d-\bar{d})^2 \|_1.\end{aligned} \tag{4.17}$$

Considering that the image warping $\mathcal{W}_{1,n-1}$ using Equation (4.13) requires estimated depth, the initial is first estimated first by using two consecutive images I_1 and I_2 with $N = 2$. The number of used images N is then gradually increased

to improve the depth accuracy. This procedure is combined with the coarse-to-fine approach described in the next section.

4.4 Variational Optimization for Depth Reconstruction

To solve Equation (4.17) for all image pixels, the energy function is defined by comprising the data and regularization terms with a scale parameter λ, such that $E = E_{reg} + \lambda E_{data}$. From Equation (4.17), the pixel-wise data cost $\rho(d, w)$ for the data term $E_{data} = \sum_{\forall x,y} \rho(d, w)$ is defined as follows:

$$\begin{aligned}\rho(d,w) = \frac{1}{N-1}\sum_{n=2}^{N} \| I'_{n-1} - I_n + \{a\mathcal{J}_{I'_{n-1}}\mathcal{J}_{\mathbf{v}_n} + (1-a)\mathcal{J}_{I_n}\mathcal{J}_{\mathbf{v}_1}\}(d-\bar{d}) \\ + \{a\mathcal{J}_{\mathbf{v}_1}^{\top}\mathcal{H}_{I_n}\mathcal{J}_{\mathbf{v}_1}\}(d-\bar{d})^2 + \beta w\|_1,\end{aligned} \tag{4.18}$$

where w and β are the temporal illumination change term and its coefficient, respectively, which are widely used in classical optical flow formulations. For handling pixel noise and textureless regions, the data cost is combined with the Huber regularization [71] given by

$$E_{reg}(d,w) = \sum_{\forall x,y} |\nabla d|_{\alpha_d} + |\nabla w|_{\alpha_w}, \tag{4.19}$$

where ∇ denotes the gradient operator, and $|\nabla|_\alpha$ denotes the Huber norm defined by

$$|\nabla|_\alpha = \begin{cases} \frac{|\nabla|^2}{2\alpha}, & \text{if } |\nabla| \leq \alpha \\ |\nabla| - \frac{\alpha}{2}, & \text{if } |\nabla| > \alpha \end{cases}. \tag{4.20}$$

The overall energy function for solving the depth map d has the form,

$$E = \sum_{\forall x,y} |\nabla d|_{\alpha_d} + |\nabla w|_{\alpha_w} + \lambda\rho(d,w). \tag{4.21}$$

In the implementation, the fixed values of parameters $\alpha_d = \alpha_w = 0.005$, $\beta = 0.002$ are used, and different values are used for the parameter λ depending on a test scene.

The minimization of Equation (4.21) is effectively achieved by using the first-order primal-dual algorithm [23], which is designed for the optimization of continuous variable convex functions. Given its fast convergence property, the algorithm is widely used in various applications that require fast optimization performance. The optimization procedure starts with an arbitrary initial depth $\bar{d}$ and gradually updates d by using the coarse-to-fine warping scheme described in Alg. 1. The coarse-to-fine warping scheme is employed because solving Eqs. (4.16) or (4.17) by using the optimization method is valid only for the small update Δd. The Jacobian matrix $\mathcal{J}_{\mathbf{v}_n}$ and the Hessian matrix $\mathcal{H}_{\mathbf{v}_n}$ are calculated for instance of every warping in outer iteration, but not for every update of latent variables d and w to save computational cost. The method for building blur kernel $\mathcal{K}$ from depth d will be described in Section 4.5.

Image warping by approximation using the Jacobian and Hessian matrices limits the warping to a simple 2D translation, but the intermediate warping and blurring in the coarse-to-fine warping scheme (line 5 and 6 in Alg. 1) enables handling of a curved motion path caused by camera rotation. Consequently, the proposed depth-based blur model can address more general motion blur compared with [4], where the blur kernel was assumed to be linear.

4.5 Deblurring by using Estimated Depth

This section describes building blur kernels from the estimated depth for deconvolution-based image deblurring. Similar to the projective motion path model in [72], the

Algorithm 1 Warping and updating for depth reconstruction

Initialization: $d = \bar{d}$
repeat
  Resize images and depth map to finer level
  for $n = 2$ to N **do**
    $I'_{n-1} \leftarrow \mathcal{W}_{1,n-1}(I_1) * \mathcal{K}_n$
    $I_n \leftarrow \mathcal{W}^{-1}_{n-1,n}(I_n) * \mathcal{K}_1$
  end for
  repeat
    Update depth d by solving Equation (4.16)
  until Hit max iteration
until Reach the finest level

blur kernel $\mathcal{K}_n$ at pixel (x, y) is represented as a set of pixel positions $\{(x^{\tau_i}, y^{\tau_i}), i \in 0, ..., M\}$, which corresponds to the motion path of pixel (x, y) during exposure time as well as the weight $k_n(x^{\tau_i}, y^{\tau_i})$ for each pixel position. The superscript τ_i denotes the M number of uniformly discretized intervals for exposure time τ, such that $\tau_i = \tau_o + \frac{\tau_c - \tau_o}{M} i$. The blurred image I_n can then be represented by

$$I_n(x, y) = \sum_{i=0}^{M} L(x^{\tau_i}, y^{\tau_i}) k(x^{\tau_i}, y^{\tau_i})_n. \tag{4.22}$$

The weight of blur kernel should satisfy the constraint $\sum_{i=0}^{M} k_n(x^{\tau_i}, y^{\tau_i}) = 1$ to preserve the image intensity, thus $k_n(x^{\tau_i}, y^{\tau_i}) = 1/(M+1)$ holds for all i. To calculate an intermediate pixel position (x^{τ_i}, y^{τ_i}) by using Equation (4.13), an intermediate camera pose $\mathbf{P}_n^{\tau_i}$ is interpolated by using the input camera poses $\mathbf{P}_{n-1}^{\tau_c}$ and $\mathbf{P}_n^{\tau_c}$ on

the manifold of $\mathbb{SE}(3)$ as follows:

$$\mathbf{P}_n^{\tau_i} = \exp(\frac{1}{\tau_c}(\tau_o \frac{\tau_c - \tau_o}{M} i)\Delta\mathbf{P}) \cdot \mathbf{P}_{n-1}^{\tau_c}, \tag{4.23}$$

where $\Delta\mathbf{P}$ is the camera motion between two input images, such that $\Delta\mathbf{P} = \log(\mathbf{P}_n^{\tau_c} \cdot (\mathbf{P}_{n-1}^{\tau_c})^{-1})$.

The blur kernel generated by this method is used for image warping in depth reconstruction as well as deblurring after obtaining the final depth map. Notably, deblurring is not essential for 3D reconstruction purpose, and we can optionally deblur input images for further computer vision tasks. By using the estimated kernel $\mathcal{K}_n$ for each pixel, Richardson-Lucy deconvolution with total variation regularization is performed similarly to [72]. Given that the pixel's motion path in images for 3D reconstruction is uncomplicated, a small number of Richardson-Lucy iterations (less than 50) are sufficient to obtain satisfactory deblurring results.

4.6 Experiments

In the experiment, the analysis of several important parameters is initially presented, then the comparative evaluations of the proposed method with other methods with respect to depth reconstruction, optical flow estimation, and deblurring then follow. The results of proposed method are obtained from gray scale images.

4.6.1 Analysis of the initial depth value

The initial value of depth for the proposed depth reconstruction is important, because the depth estimation is solved by variational optimization combined with a coarse-to-fine scheme. Therefore, the optimization performance is tested by varying the initial value of depth, as shown in Figure 4.4. The initial value $\bar{d}$ is uniformly

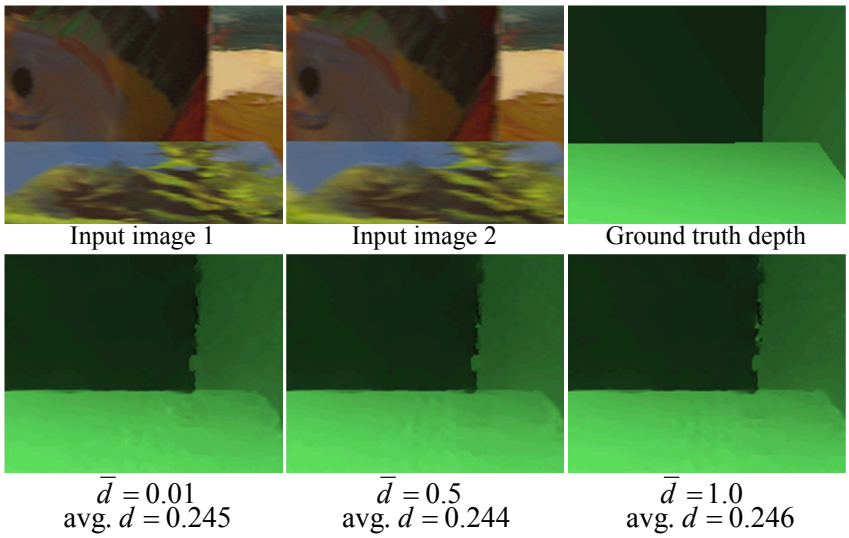


Figure 4.4: Depth maps for synthesized image set by using different initial depth values $\bar{d}$ at the coarsest level. The arbitrary initial values yield almost the similar depth results.

assigned to all pixels at the coarsest level. We can verify that the optimization is not excessively sensitive to the initial value and converges to similar results for an arbitrary initial depth value only if the initial depth is not extremely far from the true value.

4.6.2 Analysis of the number of input images

The performance gain achieved by multiple real images is shown in Figure 4.5. The use of multiple images generally provides a more accurate depth map for real noisy data. However, this is invalid when motion blur occurs in the image sequence. With motion blur, finding the pixel correspondences becomes more difficult as the number of image increases because motion blur varies for each image. Meanwhile, the proposed blur-handled depth reconstruction provides a more accurate depth map

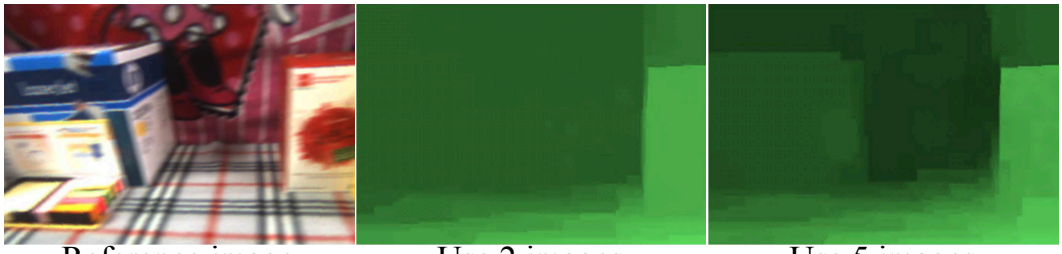


Figure 4.5: Improvement of depth map accuracy for real sequence by increasing the number of input blurry images.

as the number of input images increases.

4.6.3 Comparison of depth reconstruction results

The blur-robustness of the proposed algorithm is verified by comparing the depth reconstruction results with the conventional variational depth reconstruction implemented by removing the blur-handling parts of the proposed method. First, each method is tested for unblurred sequence to show that each implementation works correctly as shown in Figure 4.6 (a, c). The methods are then tested for blurred sequence to compare their robustness to motion blur, as shown in Figure 4.6 (b, d). The RMS error of the estimated depth are measured for the synthesized images and presented in the figure.

4.6.4 Comparison of optical flow results

The effectiveness of the proposed blur handling is demonstrated by comparing the optical flow results, *i.e.*, vector $\mathbf{v}_n$, with those of other blur-robust method for optical flow. The estimated depth map is converted into motion vectors by using Equation (4.13) and then the motion vectors are compared with the results of the

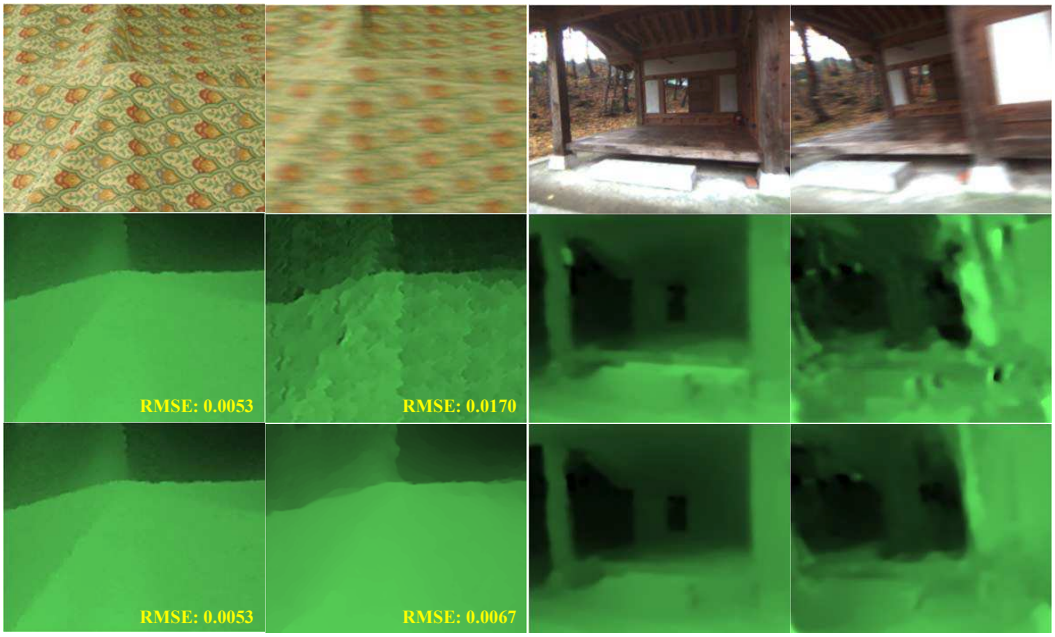


Figure 4.6: Depth reconstruction for synthetic and real sequences respectively comprises six unblurred (a, c) and blurred (b, d) images. From top to bottom: Input images, variational depth reconstruction without blur handling, and the proposed blur-robust reconstruction.

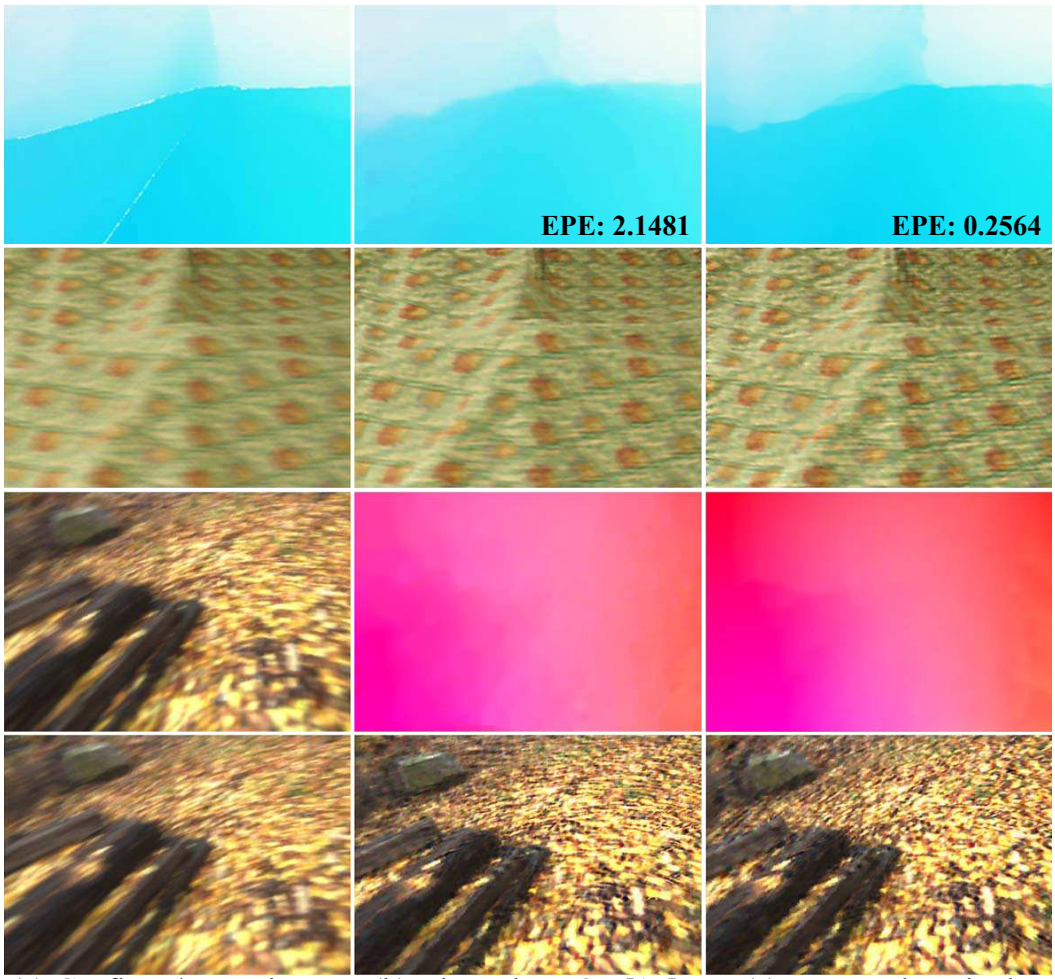


Figure 4.7: Comparison of optical flow and deblurring results. (a) Input image and ground truth motion vector of synthetic data and two input images of real data. (b) Blur-robust optical flow method in [4]. (c) Proposed method.

blur-robust optical flow method in [4] to evaluate the pixel correspondence accuracy between images. As described in Section 4.3.2, two additional images are used in [4] as additional information, whereas camera motion is used in the proposed method. The optical flow results are compared in Figure 4.7 with the average endpoint error (EPE), and deblurring results from the estimated motion vectors are shown to verify the optical flow accuracy. By re-parameterizing the optical flow to depth, the proposed method is found to be capable of handling more complex shape of motion blur and thus achieves improved results.

4.6.5 Comparison of deblurring results

Finally, the deblurring results for real image data by the proposed method are compared with the results of multiple image deblurring [6] as well as single image deblurring [5] as presented in Figure 4.8. The deblurring results by using blur kernels from blur-aware optical flow [4] is also presented. The input image has a significant depth variations in a vertical direction, which cannot be addressed by conventional video deblurring methods. Thus, the input blurry image is partially recovered. On the other hand, the proposed method successfully removes the motion blur by using the depth-aware blur kernels.

4.7 Summary

The blur-robust 3D reconstruction method was presented in this chapter. The approximation technique for blurred appearance of image was successfully combined with the depth map estimation framework based on the variational optimization. The proposed geometry-combined blur estimation enabled handling of scene depth

(a) Input image 1 (b) Input image 2 (c) Single image deblurring

(d) Video deblurring (e) Deblur using optical flow (f) Proposed deblurring

Figure 4.8: Deblurring results for real image: (a, b) Sample images from input sequence. (c) Single image deblurring [5]. (d) Video deblurring [6]. (e) Deblur using optical flow [4]. (f) Proposed method.

variation and large blur kernels, which are difficult in traditional image-only-based deblurring methods. The proposed method can be applied to not only multiple image 3D reconstruction, but also video deblurring only if the camera is calibrated for its intrinsic parameters.

Chapter 5

Dense 3D Reconstruction and Image Super-Resolution

5.1 Introduction

Note that if we employ the information about the 3D scene geometry, the super-resolution problem can be solved more efficiently since we can directly use it for enhancing the accuracy of the correspondences. That is, with estimated camera poses, the problem of finding pairwise pixel correspondences through an image sequence can be converted into estimating the depth value of corresponding pixels. Although this converted problem has an error source related to the camera pose error, because it is casted in a much lesser dimensional solution space than the original pairwise correspondence problem, it can be solved much easily and faster. Therefore, depth reconstruction and super-resolution problems are interrelated and boost each other's accuracy. So, in this study, the depth estimation is combined with the high-resolution image estimation in a unified framework, and propose a

simultaneous solution to both problems.

In the proposed method, the depth estimation and image super-resolution are formulated with a single convex energy function, which consists of data term and regularization term. The solution is estimated by convex optimization of the energy function. Although both pixel correspondences (re-parameterized by depth) and high-resolution image are estimated, the computational cost is not so expensive compared to the conventional high-resolution image estimation only because an alternating method such as EM is not used. Additionally, due to the simultaneous estimation of depth and high-resolution image, the results of the two problems are greatly enhanced.

5.2 Related Works

In this section, research combining 3D reconstruction and super-resolution that have objectives similar to the proposed method are investigated first. The works on the primal-dual algorithm for 3D reconstruction or super-resolution then follow.

5.2.1 3D reconstruction and image super resolution

In [18–20,73], the close relationship between super-resolution and 3D scene structure is pointed out and their cooperative solution is studied. In [18], the super-resolution is formulated with the calibrated 3D geometry and solved using the MAP-MRF framework. Occlusions are effectively handled in their super-resolution method using depth information, but super-resolution does not contribute to depth map estimation in this method. In [20], a method for increasing the accuracy of 3D video reconstruction using multiple static cameras is presented. The 3D video is composed

of texture images and 3D shapes, and increasing their accuracy is achieved by simultaneous super-resolution using MRF formulation and graph-cut [74]. High-quality texture and 3D reconstruction is presented in [73] where texture and shape of a 3D model are alternately estimated with joint energy functional. Compared to [73] the proposed method has more challenging settings in which neither accurate camera motions nor initial pixel correspondences are available.

The work most closely related to the proposed mehtod with respect to its objective is [19]. The authors formulate a full frame super-resolution problem combined with a depth map estimation problem, and attempt to enhance the results of both problems. However, their solution is not fully simultaneous but follows an EM-style alternating method instead. They fix the current high-resolution image for the estimation of the depth map, and vice versa. Graph-cut and iterated conditional modes (ICM) are used for the depth and high-resolution image estimation, respectively, for each iteration, which result in an inevitably large computation cost. In contrast, the globally optimum solution is searched directly with a single convex energy function in the proposed method, and very fast optimization speed is achieved for the real-time capability.

5.2.2 Primal-dual algorithm for 3D reconstruction and super-resolution

The formulation of the proposed algorithm is based on the variational approach, especially the primal-dual algorithm [23–25]. The first-order primal-dual algorithm is a very effective tool for convex variational problems due to its parallelizable characteristics. The algorithm has been used in various computer vision problems, with the wide use of parallel computing acceleration such as general-purpose computing on graphics processing units (GPGPU).

The first-order primal-dual algorithm has been applied recently for the 3D reconstruction and super-resolution problems. In [59] and [60], a dense 3D reconstruction is studied and its real-time implementations are demonstrated. They used conventional energy functions consisting of photometric consistency-based data term and L^1 or Huber norm-based smoothness term, but achieved a breakthrough performance in computation time using the primal-dual algorithm combined with the GPGPU implementation.

In [75], the first-order primal-dual algorithm is applied to the super-resolution problem. The reconstruction-based super-resolution is formulated by image downsampling, blurring, and warping, and then the latent high-resolution image is estimated with the Huber norm regularization. This method achieves a fast computation of high-quality super-resolution comparable to other methods, but has certain limitations such that highly accurate initial image warping is required and no updating procedure is involved in estimating the super-resolution.

The combination of 3D reconstruction and super-resolution is also formulated in the first-order primal-dual framework. However, unlike [59] and [60], the proposed super-resolution combined framework enables more accurate depth map estimation with respect to its resolution. The proposed image super-resolution is also accelerated by finding pixel correspondences in a depth domain instead of optical flows between images with the help of camera geometry obtained from the 3D reconstruction.

5.3 Energy Model for Simultaneous Estimation of Depth and Super-Resolution Image

In this study, a new energy function is proposed for a simultaneous estimation of depth map and high-resolution image. The inputs are $M \times N$ size low-resolution image sequence $I_j \in \mathbb{R}^{MN}$ and their corresponding camera poses $\mathbf{P}_j \in \mathbb{SE}(3)$ with $j \in \{0, ..., J\}$. Let $\mathbf{g} \in \mathbb{R}^{s^2MN}$ be the latent super-resolution image with the gray scale, and $\mathbf{d} \in \mathbb{R}^{s^2MN}$ be the latent inverse depth map, where s is the predefined upscale factor. The solution of $\mathbf{g}$ and $\mathbf{d}$ is estimated with respect to the reference view $\mathbf{P}_1$. The energy function to solve this problem is composed of the data cost E_{data} based on the photometric constancy and the regularization cost E_{reg} for smoothing undesirable artifacts. With the parameter λ which controls the degree of regularization, the energy function has the form $E(\mathbf{g}, \mathbf{d}) = E_{reg} + \lambda E_{data}$. The super-resolution image $\mathbf{g}$ can also be the color, but the gray scale notation is used here for simplicity and the color image results are shown in the experiment section.

5.3.1 Data cost

The derivation of data cost starts from the relationship between the high-resolution image $\mathbf{g}$ for the reference image I_1 and the low-resolution image I_j from an adjacent view. With the camera internal parameter $\mathbf{K}$ including the focal length and the principal point, the reprojected 3D position X of pixel (x, y) in I_1 with the inverse depth $\mathbf{d}(x, y)$ by the reference camera $\mathbf{P}_1$ is given by $X = \frac{1}{\mathbf{d}(x,y)}\mathbf{K}^{-1} \cdot (x, y, 1)^\top$, and its projection to the adjacent view with $\mathbf{P}_j$ is calculated as $h(\mathbf{K}\mathbf{P}_{j,1}\frac{1}{\mathbf{d}(x,y)}\mathbf{K}^{-1} \cdot (x, y, 1)^\top)$, where $\mathbf{P}_{j,1} = \mathbf{P}_j\mathbf{P}_1^{-1}$ and h is the dehomogenization function such that $h((x, y, z)^\top) = (x/z, y/z)$. Figure 5.1 illustrates these relationships.

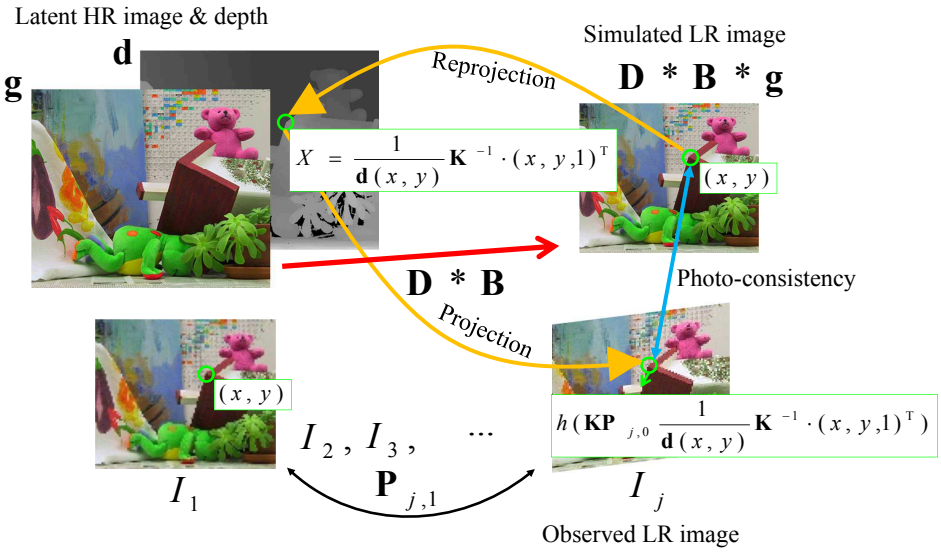


Figure 5.1: The relationship between the low-resolution input sequence I_j and the super-resolution image $\mathbf{g}$, induced by the depth map $\mathbf{d}$: The photometric consistency should hold for I_j and the simulated low-resolution image $\mathbf{D} * \mathbf{B} * \mathbf{g}$.

For notational simplicity, the non-bold characters g and d are used for the pixel-wise values $\mathbf{g}(x, y)$ and $\mathbf{d}(x, y)$, respectively, and their corresponding dual variables later. The image warping $\mathcal{W}(I_j, \mathbf{d})$, which transforms the image I_j to the reference image I_1, is defined by using the pixel projection and reprojection discussed above,

$$\mathcal{W}(I_j, \mathbf{d})(x, y) = I_j(h(\mathbf{K}\mathbf{P}_{j,1}\frac{1}{d}\mathbf{K}^{-1} \cdot (x, y, 1)^\top)). \tag{5.1}$$

Then, by the photometric consistency between the reference image and the adjacent image, the equation

$$I_1(x, y) = I_j(h(\mathbf{K}\mathbf{P}_{j,1}\frac{1}{d}\mathbf{K}^{-1} \cdot (x, y, 1)^\top)) = \mathcal{W}(I_j, \mathbf{d})(x, y) \tag{5.2}$$

holds for all $j \in \{0, ..., J\}$ if the inverse depth d has the exact value. By incorporating the image resolution degradation model, the equation

$$(\mathbf{D} * \mathbf{B} * \mathbf{g})(x, y) = I_1(x, y) = \mathcal{W}(I_j, \mathbf{d})(x, y) \tag{5.3}$$

also holds for all $j \in \{0, ..., J\}$. Here, $\mathbf{D}$ and $\mathbf{B}$ are the downsampling and the blurring operator, respectively. From the equality in Equation (5.3), we can set an objective function which finds an optimum value of $\mathbf{g}$ and $\mathbf{d}$, such that

$$\arg\min_{\mathbf{g},\mathbf{d}} \sum_{j=1}^{J} \|\mathbf{D} * \mathbf{B} * \mathbf{g} - \{\mathcal{W}(I_j, \mathbf{d})\}\|_1. \tag{5.4}$$

To find the optimized value of $\mathbf{d}$ through an iterative update, the first-order Taylor expansion to $\mathcal{W}(I_j, \mathbf{d})$ is applied to approximate a change in image $\mathcal{W}(I_j, \mathbf{d})$ with respect to a small change of depth at the initial value $\mathbf{d}_0$,

$$\mathcal{W}(I_j, \mathbf{d}) \simeq \mathcal{W}(I_j, \mathbf{d}_0) + \frac{\partial}{\partial \mathbf{d}}\mathcal{W}(I_j, \mathbf{d})\Big|_{\mathbf{d}=\mathbf{d}_0} \cdot (\mathbf{d} - \mathbf{d}_0). \tag{5.5}$$

Then, the objective function (5.4) can be rewritten as a linearized form,

$$\arg\min_{\mathbf{g},\mathbf{d}} \sum_{j=1}^{J} \|\mathbf{D} * \mathbf{B} * \mathbf{g} - \{\mathcal{W}(I_j, \mathbf{d}_0) + I_{j_\mathbf{d}} \cdot (\mathbf{d} - \mathbf{d}_0)\}\|_1, \tag{5.6}$$

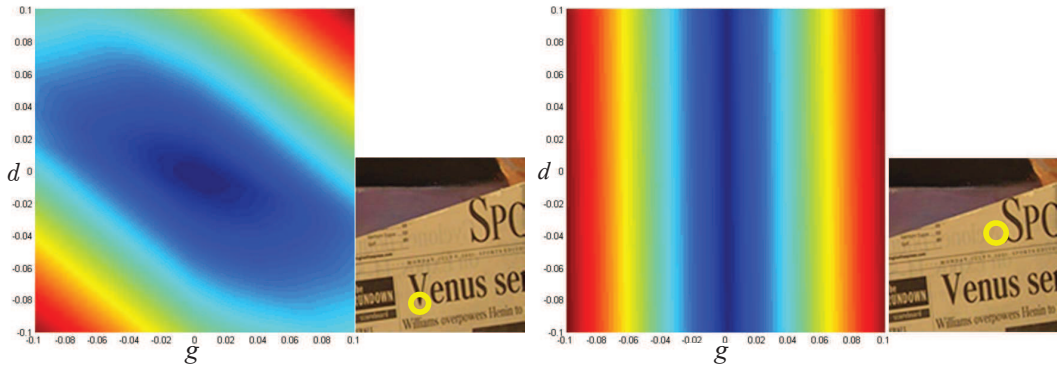


Figure 5.2: The shape of data cost $\rho(\mathbf{g}, \mathbf{d})$ for textured (left) and untextured (right) region.

where $I_{j_\mathbf{d}}$ is the simplified notation of the image derivative $\frac{\partial}{\partial \mathbf{d}}\mathcal{W}(I_j, \mathbf{d})$, which can be calculated pixel-wise using the chain-rule,

$$I_{j_\mathbf{d}} = \frac{\partial \mathcal{W}(I_j, \mathbf{d}_0)}{\partial d} = \frac{\partial \mathcal{W}(I_j, \mathbf{d}_0)}{\partial x}\frac{\partial x}{\partial d} + \frac{\partial \mathcal{W}(I_j, \mathbf{d}_0)}{\partial y}\frac{\partial y}{\partial d}. \tag{5.7}$$

The blur kernel $\mathbf{B}$ is predefined with the simple Gaussian blur model, with the standard deviation s and the kernel size of $(s-1)^{1/2}$. To handle the downsampling operator $\mathbf{D}$ efficiently, the low-resolution input images are upscaled to the high-resolution size $sM \times sN$ as $I_j \in \mathbb{R}^{MN} \rightarrow \hat{I}_j \in \mathbb{R}^{s^2MN}$ using bicubic interpolation and the optimization process is performed with the resized image space $\mathbb{R}^{s^2MN}$. The resulting data cost then has the form,

$$\begin{aligned} E_{data} &= \int_{X,Y} \rho(\mathbf{g}, \mathbf{d}) \\ &= \int_{X,Y} \sum_{j=1}^{J} \|\mathbf{B} * \mathbf{g} - \{\mathcal{W}(\hat{I}_j, \mathbf{d}_0) + \hat{I}_{j_\mathbf{d}}(\mathbf{d} - \mathbf{d}_0)\}\|_1. \end{aligned} \tag{5.8}$$

Figure 5.2 shows an example of the convexity of data cost $\rho(\mathbf{g}, \mathbf{d})$ for different image points. The shape of the cost function is obviously convex, but the shape

of the function varies from image point to point according to the image gradient. In a low texture region, the data cost is dominated by the high-resolution intensity $\mathbf{g}$ than the depth $\mathbf{d}$. Therefore, regularization is required to get a more plausible solution for depth $\mathbf{d}$.

5.3.2 Regularization

For image intensity $\mathbf{g}$ and inverse depth $\mathbf{d}$, a Huber norm based regularization is used to get a smoothed and discontinuity-preserved result. The Huber norm for $\mathbf{g}$ is defined by following pixel-wise function:

$$\|\nabla \mathbf{g}\|_{\alpha_g}(x,y) = \begin{cases} \frac{|\nabla g|^2}{2\alpha_g}, & \text{if} |\nabla g| \leq \alpha_g \\ |\nabla g| - \frac{\alpha_g}{2}, & \text{if} |\nabla g| > \alpha_g \end{cases}, \tag{5.9}$$

where ∇ is the linear operator that computes derivatives of x and y direction. The Huber norm for $\|d\|_{\alpha_d}$ is defined in the same way. In the implementation of algorithm, the parameters are set to $\alpha_g = \alpha_d = 0.001$.

By combining the data cost (6.7) and the regularization (6.9), the target energy function $E(\mathbf{g}, \mathbf{d})$ is obtained by

$$E(\mathbf{g}, \mathbf{d}) = \int_{X,Y} \|\nabla \mathbf{g}\|_{\alpha_g} + \|\nabla \mathbf{d}\|_{\alpha_d} + \lambda \rho(\mathbf{g}, \mathbf{d}). \tag{5.10}$$

In the next section, the solution of this energy function is described.

5.4 Solution of Energy Function

5.4.1 Initial depth estimation

In the data cost (6.7), the first-order Taylor expansion, which can only handle a small update for $\mathbf{g}$, and $\mathbf{d}$ is applied. This step requires the starting point of optimization

to be close to the global optimum. The initial value of $\mathbf{g}$ can be easily obtained by upscaling the input image at reference view using simple bicubic interpolation. However, the initial value of $\mathbf{d}$ should be estimated using the low-resolution input sequence.

The cost function for initial depth estimation is easily obtained from Equation (6.7) and (5.10) by replacing $\mathbf{B} * \mathbf{g}$ and $\hat{I}_j$ with the low-resolution images I_1 and I_j, respectively, and removing the regularization on $\mathbf{g}$. The resulting energy function for low-resolution depth map $\check{\mathbf{d}}$ is

$$E(\check{\mathbf{d}}) = \int_{X,Y} \|\check{\mathbf{d}}\|_{\alpha_d} + \lambda \sum_{j=1}^{J} \|I_1 - \{\mathcal{W}(I_j, \check{\mathbf{d}}_0) + I_{j\check{\mathbf{d}}} \cdot (\check{\mathbf{d}} - \check{\mathbf{d}}_0)\}\|_1. \tag{5.11}$$

The equation (5.11) is actually a conventional formulation for depth map estimation. The optimization of this energy function is almost similar to the optimization of Equation (5.10), which will be explained below, so the optimization of (5.11) is skipped here. The limitation of a small update also holds for Equation (5.11). Thus, a coarse-to-fine approach is used to approach the global optimum of $\mathbf{d}$ gradually by starting from an arbitrary initial solution, *e.g.*, filled with 1.0. The depth result obtained at the finest level is upscaled using bicubic interpolation and is fed to the optimization of (5.10) as an initial value.

5.4.2 High-resolution image and depth estimation

Now the solution of Equation (5.10) is described based on the first-order primal-dual optimization algorithm. By interpreting the objective function (5.10) as the primal-dual formulation, we can rewrite it as a generic saddle point problem with the dual variables $\mathbf{p}$ and $\mathbf{q}$, which corresponds to $\mathbf{g}$ and $\mathbf{d}$, respectively:

$$\min_{\mathbf{g},\mathbf{d}} \max_{\mathbf{p},\mathbf{q}} \langle \nabla\mathbf{g}, \mathbf{p}\rangle + \langle \nabla\mathbf{d}, \mathbf{q}\rangle + \lambda\|\rho(\mathbf{g}, \mathbf{d})\|_1 - \delta_{\mathbf{P}}(\mathbf{p}) - \frac{\alpha_g}{2}\|\mathbf{p}\|_2^2 - \delta_{\mathbf{Q}}(\mathbf{q}) - \frac{\alpha_d}{2}\|\mathbf{q}\|_2^2, \tag{5.12}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product, and the functions $\delta_{\mathbf{P}}$ and $\delta_{\mathbf{Q}}$ are the indicator functions given as $\delta_{\mathbf{P}}(\mathbf{p}) = \begin{cases} 0, & \text{if } \|\mathbf{p}\|_\infty \leq 1 \\ \infty, & \text{if else.} \end{cases}$ and $\delta_{\mathbf{Q}}(\mathbf{q}) = \begin{cases} 0, & \text{if } \|\mathbf{q}\|_\infty \leq 1 \\ \infty, & \text{if else.} \end{cases}$, respectively.

This problem can be optimized through the iteration,

$$\begin{aligned} (\mathbf{p}, \mathbf{q})^{n+1} &= \mathcal{R}_{\mathbf{p},\mathbf{q}} \left((\mathbf{p}, \mathbf{q})^n + \sigma \nabla (\bar{\mathbf{g}}, \bar{\mathbf{d}})^n \right) \\ (\mathbf{g}, \mathbf{d})^{n+1} &= \mathcal{R}_{\mathbf{g},\mathbf{d}} \left((\mathbf{g}, \mathbf{d})^n - \tau \nabla^* (\bar{\mathbf{p}}, \bar{\mathbf{q}})^n \right) \\ (\bar{\mathbf{g}}, \bar{\mathbf{d}})^{n+1} &= 2(\mathbf{g}, \mathbf{d})^{n+1} - (\bar{\mathbf{g}}, \bar{\mathbf{d}})^n. \end{aligned} \tag{5.13}$$

where the operator ∇^*, the conjugate of ∇ as $\nabla^* = -\operatorname{div}$, computes the divergence [23], and $\bar{\mathbf{g}}$ and $\bar{\mathbf{d}}$ are the intermediate variables for the convergence of algorithm. The initial value $(\mathbf{g}, \mathbf{d})^0$ is obtained from Section 4.1, and $(\mathbf{p}, \mathbf{q})^0$ is set to zero. The operators $\mathcal{R}_{\mathbf{p},\mathbf{q}}$ and $\mathcal{R}_{\mathbf{p},\mathbf{q}}$ are the resolvent operators that search lower energy values using subgradients. τ and σ are constants that control the convergence of primal and dual variable, respectively. The resolvent operators will be discussed in more detail.

The regularization term (5.10) is a typical form used in [23]. Thus, the resolvent operator of the dual variables is a pixel-wise projection

$$\mathcal{R}_{p,q}(p, q) = \left(\frac{p}{max(1, |p|)}, \frac{q}{max(1, |q|)} \right). \tag{5.14}$$

On the other hand, the data cost has a difference with the standard form in previous primal-dual algorithm applications. This difference comes from the summation of absolute value in the data cost for image sequence. Since L^1 norm is used for the difference between two images, there are some critical (non-differentiable) points in their summation. Therefore, these non-differentiability should be handled in the

optimization procedure. The minimization of similar cost function is introduced in [60], but the solution space of [60] is for the depth map only, so the minimization can be efficiently achieved by evaluating and sorting all critical points. On the other hand, the solution space of the problem is composed of depth map and image intensity, so there are J^2 critical points. Searching them is not straightforward, and thus optimization by evaluating and sorting critical points is inefficient. Instead, the general gradient descent and critical point searching are combined to accelerate the minimization procedure.

Let per-image data cost $\|\rho_j(\mathbf{g},\mathbf{d})\|_1 = \|\mathbf{B} * \mathbf{g} - \{\mathcal{W}(\hat{I}_j, \mathbf{d}_0) + \hat{I}_{j_\mathbf{d}} \cdot (\mathbf{d} - \mathbf{d}_0)\}\|_1$, then we can write $\rho(\mathbf{g},\mathbf{d})$ as

$$\rho(\mathbf{g},\mathbf{d}) = \sum_{j=1}^{J} \|\rho_j(\mathbf{g},\mathbf{d})\|_1 = \sum_{j=1}^{J} sgn(\rho_j(\mathbf{g},\mathbf{d})) \cdot \rho_j(\mathbf{g},\mathbf{d}), \tag{5.15}$$

where $sgn(\cdot)$ is a signum function. Then the derivatives of (5.15) are calculated as

$$\partial\rho(\mathbf{g},\mathbf{d}) = \sum_{j=1}^{J} sgn(\rho_j(\mathbf{g},\mathbf{d})) \cdot \left(1, -\hat{I}_{j_\mathbf{d}}^{\top}\right). \tag{5.16}$$

The domain of resolvent operator is divided into two intervals based on the cost ρ and the magnitude of gradient $\|\partial\rho\|_2^2$, and the gradient descent search and critical point search are applied by

$$\mathcal{R}_{g,d}(g,d) = \begin{cases} (g,d) - \tau\lambda\left(\partial\rho(g,d)\right), \\ \quad \text{if } \rho(g,d) > \tau\lambda\|\partial\rho(g,d)\|_2^2 \\ (g,d) - \frac{\rho_j^*(g,d)\cdot\partial\rho_j^*(g,d)}{\left\|\partial\rho_{j^*}(g,d)\right\|_2^2}, \\ \quad \text{if } \rho(g,d) < \tau\lambda\|\partial\rho(g,d)\|_2^2 \end{cases}, \tag{5.17}$$

where

$$j^* = \underset{\{j|\rho_j(g,d)\cdot sgn(\nabla\rho(g,d))>0\}}{\arg\min} \|\rho_j(g,d)\|_1. \tag{5.18}$$

The operation of the second case in (5.17) is searching the closest critical point with a lower cost value by (5.18), and moving the variable to this critical point. By iterating Equation (5.13) and checking the amount of changes in total cost (5.10), we can terminate the iteration and can get the final results of $\mathbf{g}$ and $\mathbf{d}$.

5.5 Implementation of 3D Reconstruction

5.5.1 Camera localization

To use the proposed depth map estimation and super-resolution algorithm in the single camera 3D reconstruction system, the camera localization algorithm needs to be incorporated. Before the depth map is estimated for an initial few frames, the sparse point-based SLAM is performed for camera localization. After the initial depth map is created, the image registration method similar to the 2.5D image registration in [59] is used between the input frame and the pre-warped image from the estimated high-resolution image and depth map to estimate a new camera pose $\mathbf{P}_{J+1}$ as:

$$\mathbf{P}_{J+1} = \arg\max_{\mathbf{P}} \int_{X,Y} \|\mathbf{g}(x,y) - I_{J+1}(h(\mathbf{K}\mathbf{P}\mathbf{P}_1^{-1}\frac{1}{d}\mathbf{K}^{-1}\cdot(x,y,1)^\top)\|. \tag{5.19}$$

The optimization of this function can be achieved by predicting $\mathbf{P}_{J+1}$ using the motion dynamics and iteratively approaching to optimum value using the gradient-based method.

There are advantages to estimating a camera pose using high-resolution image $\mathbf{g}$. The image registration can be robust to image degradation such as image noise, downsampling, and blurring. Since the input images are the degraded version of a scene by those effects, the recorded images are different from the real appearance of

the scene. The estimated image **g** can be regarded as the most probable appearance of a real scene, because it is estimated from a number of instance images.

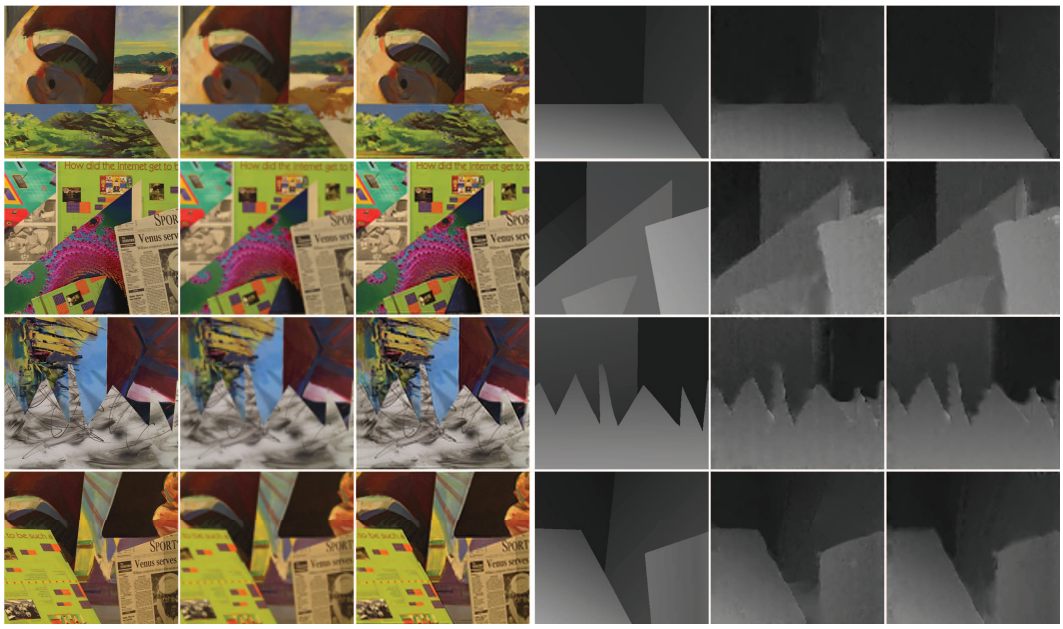
5.5.2 Map management

The proposed method estimates an inverse depth map instead of 3D points of sparse features or full 3D surface; hence, the map does not increase continuously. The depth map is reconstructed for some selected keyframes, and the relationship between depth maps is calculated and stored as a relative representation [13]. Although the depth map-based representation does not provide a visually attractive 3D surface, it has the advantage that the depth map merging step which takes large amount of computation is not required in this representation.

When the overlap between the reconstructed depth map and the current input image goes below threshold, the estimation of new depth map and high-resolution image is then performed. The overlapped depth map is propagated to new depth estimation and used as an initial value. The relative pose between the previous keyframe and the new keyframe is stored, and the current camera pose is set to identity. The camera poses for subsequent frames are estimated with respect to the current keyframe's pose.

5.6 Experiments

The proposed algorithm is implemented using NVIDIA's CUDA for GPGPU parallelization, and the implementation is tested by using 3.3GHz quad core processor and GeForce GTX 570 which has 480 stream processors. The algorithm performance is evaluated by three factors; super-resolution result, depth map estimation result, and



(a) GT image (b) LR image (c) SR image (d) GT depth (e) LR depth (f) SR depth

Figure 5.3: Depth map estimation and super-resolution results on the synthesized low-resolution image sequences *Bull*, *Poster*, *Sawtooth*, and *Venus* in [7]. (a) Original images. (b) Synthesized low-resolution images. (c) Super resolution images. (d) Ground truth depth. (e) Depth map without super-resolution. (f) Depth map with super-resolution.

registration error for camera localization. The proposed algorithm is evaluated by performing a quantitative analysis using synthetic data and a feasibility test using real image sequence.

5.6.1 Results on simulated data

The images and depth maps from [7] which have no occlusion information are used as a simulated data. For a given high-resolution image and its ground truth depth map from a reference view, the low-resolution image set is synthesized by warping

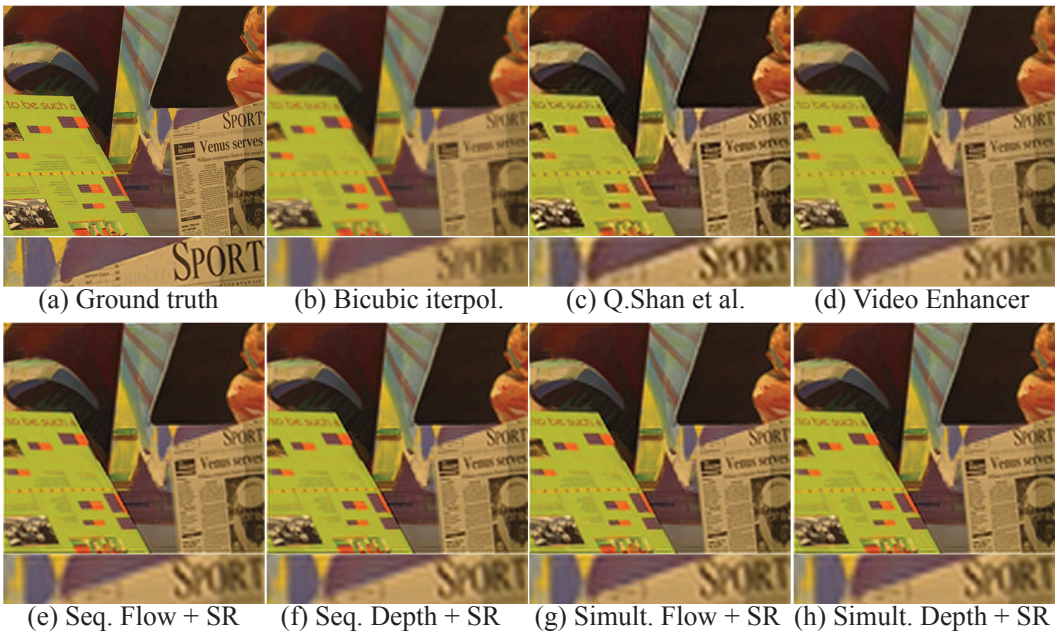


Figure 5.4: Comparison of super-resolution results ($\times$ 4) on the synthesized *Venus* sequence with other super-resolution methods.

and downsampling the high-resolution image. The virtual camera motion is simulated with a combination of arbitrary translation and rotation, and 20 low-resolution images of one-fourth scale (for example, 109 $\times$ 96 size for *Venus* image data) are obtained. The super-resolution image and depth map are estimated with their original scale, and their errors with respect to ground truth are calculated.

Figure 5.3 shows the results on synthetic data. The low-resolution images and depth maps Figure 5.3-(b, e) are obtained by bicubic interpolation of the input images and initial depth maps. From the results of the proposed algorithm shown in Figure 5.3-(c, f), we can see the improved depth map result as well as super-resolution image. In the closed-up region, the low-resolution input image has a degraded texture which makes depth estimation difficult. By recovering high-resolution

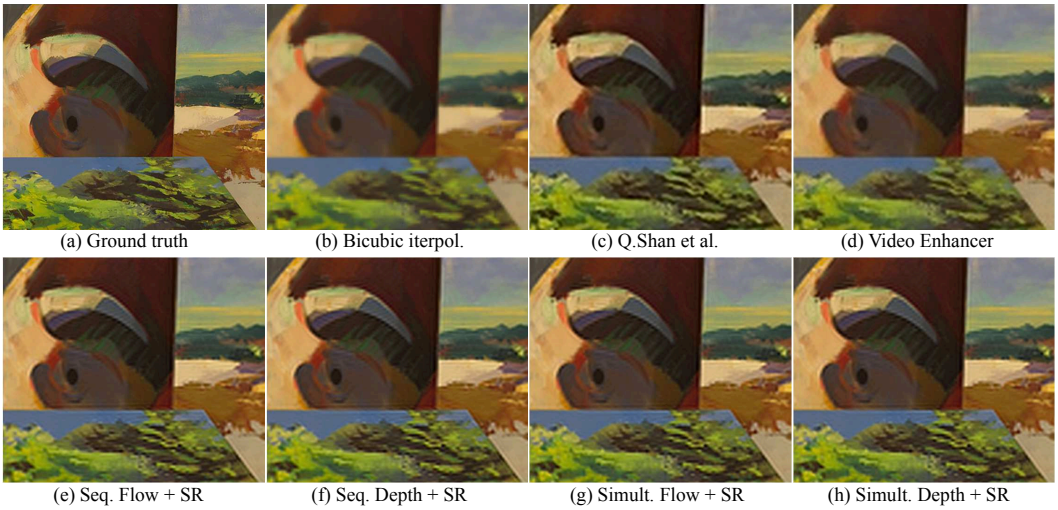


Figure 5.5: Comparison of super-resolution results ($\times$ 4) on the synthesized *Bull* sequence with other super-resolution methods.

texture using super-resolution, we can also recover the correct depth map.

Various methods for the super-resolution are tested to analyze the accuracy and efficiency of the proposed algorithm. To test the contribution of the simultaneous estimation of depth map and high-resolution images, the simultaneous implementation is replaced with the sequential method. In the sequential algorithm, the energy function (5.10) is minimized with a fixed $\mathbf{g}$ obtained from the bicubic interpolation of reference view, and then $\mathbf{g}$ is estimated with the obtained $\mathbf{d}$ fixed. The result of sequential method is shown in Figure 5.4-(f), where we can see the limitation of sequential methods in the quantitative analysis in Table 5.1.

The efficiency of depth based formulation for super-resolution is also verified by comparing the results and computation time with the pairwise correspondence (optical flow) based formulation in which the optical flow vectors between the reference view and the other view are estimated simultaneously. The objective has a form

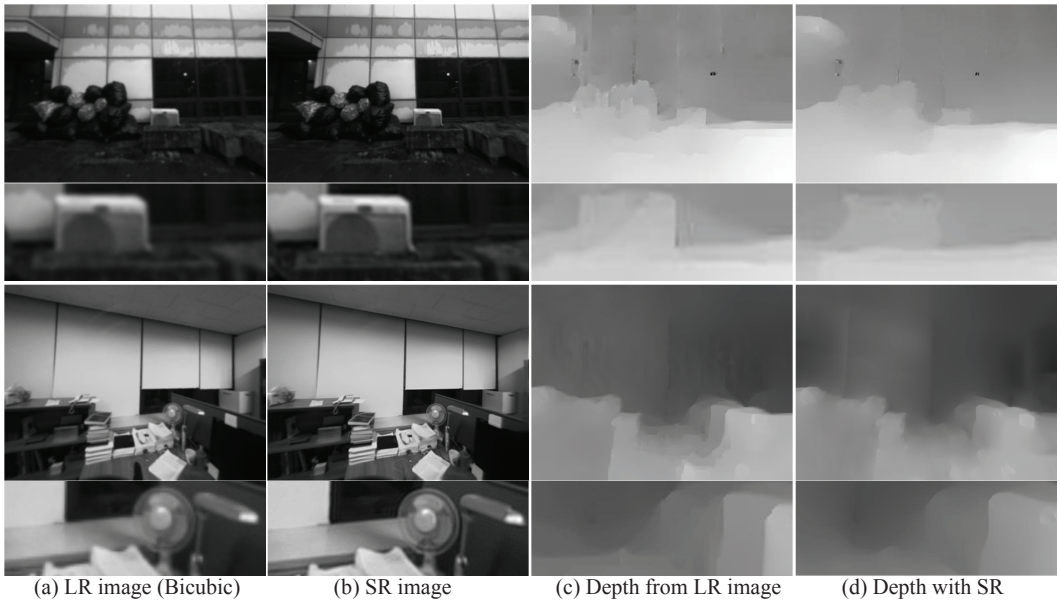
Table 5.1: PSNR (in dB), SSIM (Structural similarity, closer to 1 is better), and computation time (in second) of various super-resolution algorithm.

Image		Bicubic	[76]	[77]	Seq. Flow +SR	Seq. Depth +SR	Sim. Flow +SR	Sim. Depth +SR
Bull	PSNR	15.69	16.08	16.59	16.78	16.76	16.82	**16.83**
	SSIM	0.77	0.7762	**0.79**	0.78	0.78	0.79	**0.79**
Poster	PSNR	13.71	12.69	**13.98**	13.65	13.67	13.85	13.87
	SSIM	0.54	0.57	0.57	0.56	0.56	**0.57**	0.57
Sawtooth	PSNR	12.67	12.63	13.17	12.91	12.86	**13.20**	13.19
	SSIM	0.66	0.67	0.69	0.67	0.67	**0.69**	0.69
Venus	PSNR	15.14	14.75	15.66	15.74	15.74	**15.87**	15.86
	SSIM	0.71	0.71	0.72	0.72	0.72	**0.73**	0.73
Avg. comp. time		-	22.93	1.21	19.05	1.625	18.26	0.97

similar to Equation (5.6) as follow:

$$\underset{\mathbf{g},\mathbf{v}_1,\ldots,\mathbf{v}_J}{\arg\min} \sum_{j=1}^{J} \|\mathbf{D} * \mathbf{B} * \mathbf{g} - \{\mathcal{W}(I_j, \mathbf{v}_j) + I_{j_{\mathbf{v}_j}}^{\top} \cdot (\mathbf{v}_j - \mathbf{v}_{j0})\}\|_1, \tag{5.20}$$

where $\mathcal{W}(I_j, \mathbf{v}_j)$ is the image warping by flow $\mathbf{v}_j$, and $I_{j_{\mathbf{v}_j}}$ is the image derivative in the x and y direction, respectively. The results are shown in Figure 5.4-(g), together with its sequential estimation version in Figure 5.4-(e). Figure 5.4-(g) shows very similar accuracy with the proposed algorithm shown in Figure 5.4-(h), but it and its sequential version take much more computation time due to their high dimensional $(2 \times J + 1)$ solution space. Another comparison of super-resolution results for the synthesized *Bull* sequence is presented in Figure 5.5 and Table 5.1 summarizes the PSNR, SSIM, and computation time for each algorithm, together with the results from other high-performance super-resolution algorithms [76] and [77] whose executables are available for public.



(a) LR image (Bicubic) (b) SR image (c) Depth from LR image (d) Depth with SR

Figure 5.6: Depth map estimation and super-resolution results on the real image sequences. (a) Input images. (b) Super resolution images. (c) Depth map without super-resolution. (d) Depth map with super-resolution.

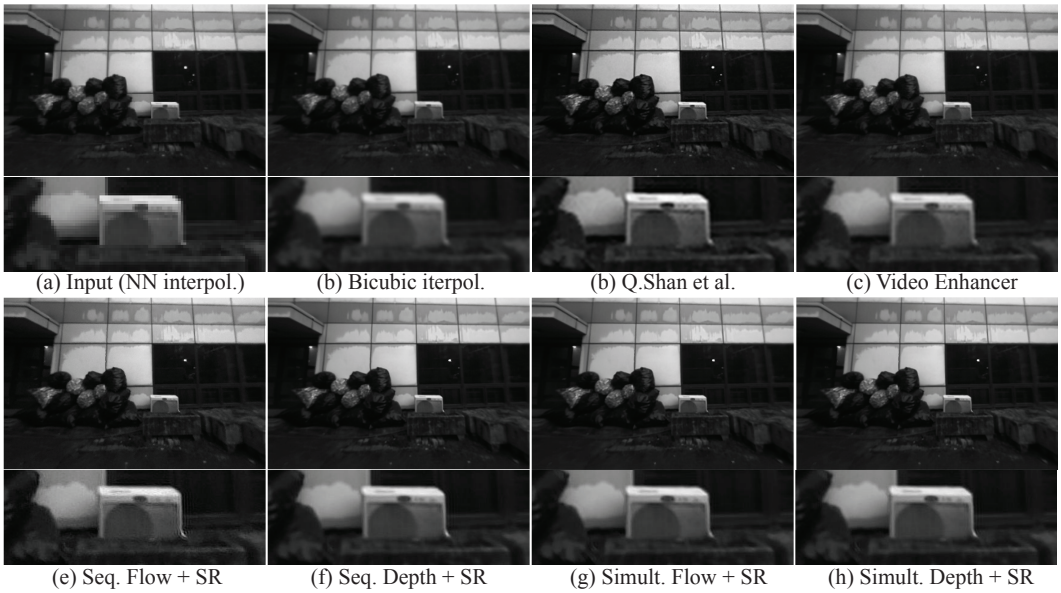


Figure 5.7: Comparison of super-resolution results on the real image sequences.

5.6.2 Results on real sequence

Different from the synthesized data, the real data have camera pose errors because it is estimated from the real image sequence. Therefore, the effect of camera pose error in the proposed algorithm can be analyzed using a real data set. A wide FOV camera is used for the effective 3D reconstruction, and the radial distortion is removed in advance. Figure 5.6 shows the reconstructed depth map and super-resolution images, and Figure 5.7 shows the comparison of various super-resolution algorithms previously discussed in the simulated data experiments. The results indicate that the camera pose error is not an important error factor for super-resolution.

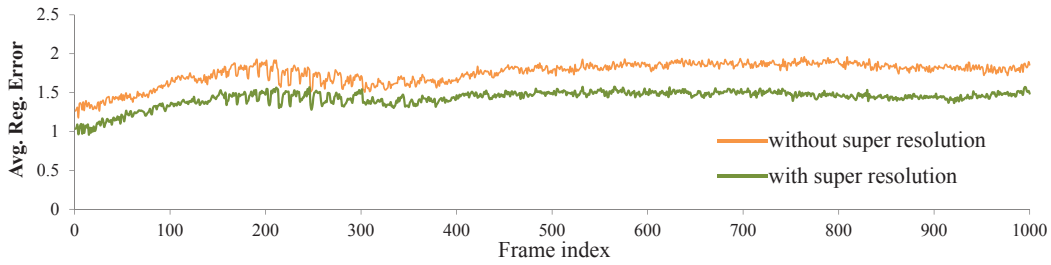


Figure 5.8: Plot of registration error for camera localization with high-resolution and low-resolution image and depth map for *outdoor* sequence.

5.6.3 Camera localization performance

The improvement of the camera localization performance is measured by registration error from Equation (5.19) through the image sequence. For a fair comparison, the original input images are used in the registration error calculation, because super-resolution images can reduce the photometric errors by themselves. Thus, only the depth map and the camera pose can affect the registration error, and the system which has a consistent depth map and camera trajectory through the whole sequence will have a lower average registration error. The plot of registration error for *indoor* sequence is shown in Figure 5.8. The average per-pixel registration error (with intensity interval [0, 255]) with the high-resolution estimation is 1.430, whereas it is 1.752 for the camera localization with low-resolution images and depth map.

5.7 Summary

A novel optimization framework for simultaneous super-resolution and depth map estimation is proposed. Two closely related problems are formulated by a single convex problem using the camera geometry and solved efficiently by the first-order

primal-dual algorithm. The proposed simultaneous solution gives results comparable to other high-performance algorithms for each problem, but takes much less computation time. Thus, the proposed framework can be applied to real-time 3D reconstruction systems for improving their accuracy.

Chapter 6

Dense 3D Reconstruction, Image Deblurring, and Super-Resolution

6.1 Introduction

The deblurring and super-resolution methods for dense 3D reconstruction presented in Chapter 4 and 5 are combined in this chapter. In fact, the deblurring and super-resolution problems are closely related to each other. Deblurring can be regarded as a temporal super-resolution that the resolution of image sequence increases with respect to time axis. Reversely, super-resolution can be interpreted as a deblurring of defocus blur for an upscaled image. Therefore, the joint problem of deblurring and super-resolution for a video or image sequence can be formulated and has been addressed in several works [63, 78, 79].

By incorporating the motion blur and super-resolution model considered in the

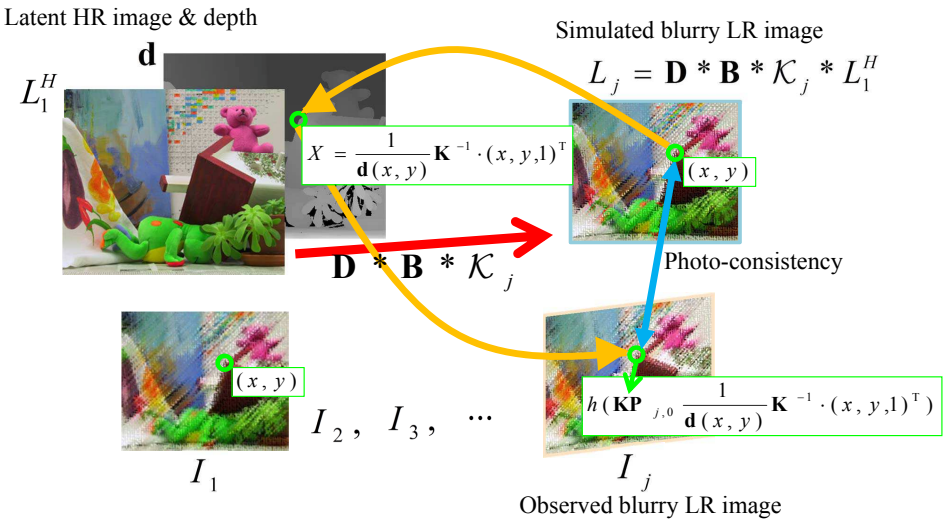


Figure 6.1: The relationship between the blurred low-resolution input sequence I_j and the sharp high-resolution image L. The photometric consistency should hold for I_j and the simulated low-resolution image $\mathbf{D} * \mathbf{B} * \mathcal{K}_j * L$.

energy function for depth map estimation, a new energy function for the simultaneous estimation of high-resolution depth map and blur kernels for deblurring is proposed in this chapter. Both motion blur process and image downsampling process are modeled in the proposed energy function to synthesize low-resolution blurry images from latent high-resolution image and depth map. The synthesized images are then compared with observed low-resolution images to update latent variables. At first, the convergence of the proposed energy function is analyzed, and then the optimization of the energy function is presented in this chapter.

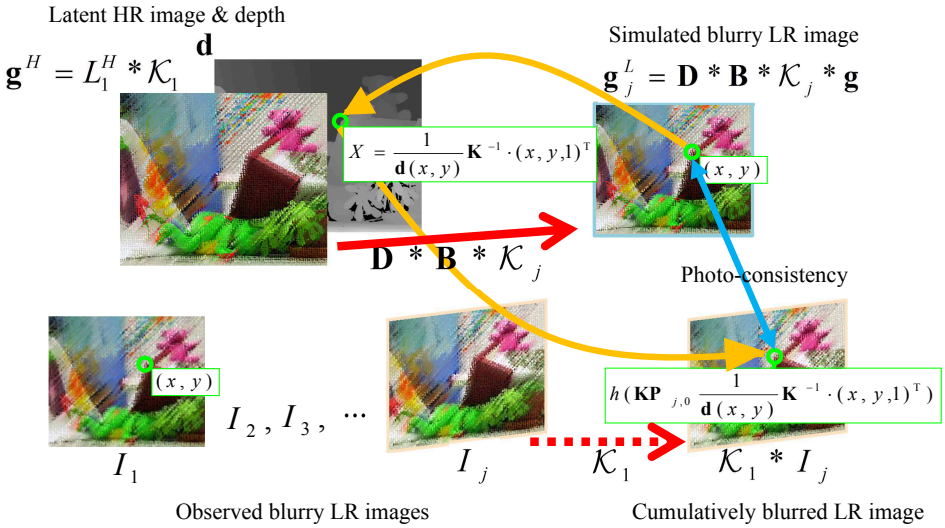


Figure 6.2: The modified model to formulate a energy function with respect to $\mathbf{g}$ and $\mathbf{d}$. The photometric consistency should hold for the cumulatively blurred image $\mathcal{K}_1 * I_j$ and the simulated low-resolution image $\mathbf{D} * \mathbf{B} * \mathcal{K}_j * \mathbf{g}$.

6.2 Energy Model for Simultaneous Estimation of Depth and Recovered Image

The main ideas used in the deblurring or super-resolution combined dense reconstruction method described in the previous chapters are incorporated to propose the unified energy function for the simultaneous estimation of a high-resolution depth map and images as well as blur kernels for removing motion blur in high-resolution image. An ideal energy function to obtain both high-resolution depth map and high-resolution deblurred image is to use them as latent variables for the energy function as illustrated in Figure 6.1. However, a direct estimation of deblurred image without deconvolution is very difficult as we can see that most deblurring methods apply explicit deconvolution after the blur kernel estimation. In the proposed method,

therefore, the latent variable for the energy function is given by a high-resolution version of the blurry reference image instead of its deblurred image as illustrated in Figure 6.2. The final deblurred high-resolution image, denoted by L_1^H, is obtained by applying deconvolution with the estimated blur kernel $\mathcal{K}_1$ for the reference image to the estimated high-resolution image $\mathbf{g}$ which satisfies the equation $\mathbf{g} = L_1^H * \mathcal{K}_1$.

The total latent variables in the energy function are a high-resolution image $\mathbf{g}$ and a depth map $\mathbf{d}$ with respect to the reference image I_1 which is from an input image sequence $I_j, j \in [1, ..., M]$. The data cost of energy function is defined by the difference between the cumulatively blurred images similarly to deblurring model in Chapter 4. The latent high-resolution image $\mathbf{g}$ is warped to jth view and blurred by kernel $\mathcal{K}_j$, and the jth observed is blurred by kernel $\mathcal{K}_1$ that is blur kernel for the reference image. Their difference is then computed and used to update values of $\mathbf{g}$ and $\mathbf{d}$.

Let $\mathbf{D}$ and $\mathbf{B}$ be the donwsampling operator and $\mathcal{K}_j$ be the blur kernel for jth input image. Synthesizing a blurred low-resolution image is then composed of downsampling and blurring operation by $\mathbf{D}$ and $\mathcal{K}$ as follows:

$$\mathbf{g}_j^L = \mathbf{D} * \mathbf{B} * \mathcal{K}_j * \mathbf{g}. \tag{6.1}$$

The blur kernel $\mathcal{K}$ is given by pixel-wise, and the notation for pixel index is also omitted here for notational simplicity. Since the estimation is performed on the high-resolution image space for $\mathbf{g}$ and $\mathbf{d}$, the input image I_j is upscaled to high-resolution scale, and the upscaled image is represented by $\hat{I}_j$. The downsampling operation is then not required for synthesizing $\mathbf{g}_j^L$ and the synthesized image $\hat{\mathbf{g}}_j^L$ is modeled as

$$\hat{\mathbf{g}}_j^L = \mathbf{B} * \mathcal{K}_j * \mathbf{g}. \tag{6.2}$$

The commutative property holds between Gaussian blur $\mathbf{B}$ and motion blur $\mathcal{K}$, and Equation (6.3) can be rewritten as

$$\hat{\mathbf{g}}_j^L = \mathcal{K}_j * \mathbf{B} * \mathbf{g} = \mathcal{K}_j * \mathbf{g}^B. \tag{6.3}$$

where $\mathbf{g}^B$ is the Gaussian blurred version of the latent high-resolution image $\mathbf{g}$.

By applying the approximation of blurred image described in Equation (4.5) up to the first order, the current estimate of latent image g is transformed to simulate the jth observed blurry image g_j^L as follows:

$$\hat{\mathbf{g}}_j^L \simeq \mathbf{g}^B + a\mathbf{g}_{j,d}^B(d - \bar{d}), \tag{6.4}$$

where $\bar{d}$ is an initial depth for the optimization, and $\mathbf{g}_{j,d}^B$ is the derivative of g with respect to depth d for jth view, which can be calculate by the chain-rule:

$$\mathbf{g}_{j,d}^B = \frac{\partial \mathbf{g}_{j,d}^B}{\partial d} = \frac{\partial \mathbf{g}_{j,d}^B}{\partial x}\frac{\partial x}{\partial d} + \frac{\partial \mathbf{g}_{j,d}^B}{\partial y}\frac{\partial y}{\partial d}. \tag{6.5}$$

The approximation of Equation (6.4) is only valid for a small difference value $(d-\bar{d})$, thus the coarse-to-fine approach is used again in the optimization of the proposed cost function.

By replacing the Gaussian blurred image $\mathbf{B}*g$ in Equation (5.6) with the motion blurred image $g_{j,d}^L$, we obtain the pixel-wise data cost $\rho(g,d)$ for the simultaneous estimation of latent image and scene depth as follows:

$$\rho(g,d) = \sum_{j=0}^{J} \|\{g + ag_{j,d}^L(d - \bar{d})\} - \{\mathcal{W}_j^L(I_j, \bar{d}) + I_{j,d}^L(d - \bar{d})\}\|_1, \tag{6.6}$$

which can be rewritten as

$$\rho(g,d) = \sum_{j=0}^{J} \|g - \mathcal{W}_j^L(I_j, \bar{d}) + (ag_{j,d}^L - I_{j,d}^L)(d - \bar{d})\|_1, \tag{6.7}$$

where $\mathcal{W}_j^L$ is a function that warps the observation image I_j into the reference view given with a depth. The derivative $I_{j,d}^L$ is given similarly to Equation (5.7),

$$I_{j,d}^L = \frac{\partial \mathcal{W}_j^L(I_j, d-\bar{d})}{\partial d} = \frac{\partial \mathcal{W}_j^L(I_j, d-\bar{d})}{\partial x}\frac{\partial x}{\partial d} + \frac{\partial \mathcal{W}_j^L(I_j, d-\bar{d})}{\partial y}\frac{\partial y}{\partial d}. \tag{6.8}$$

In the super-resolution combined 3D reconstruction method, the Huber norm based regularization is used for both latent high-resolution image and depth map. However, the L^1 norm based regularization term in the Huber regularization can generate negative effects on estimation of latent high-resolution image g because it destroys motion blurred edges in the latent image, not recovering the sharp image using their actual blur kernels. Therefore, the regularization is only applied to latent depth map d with the Huber norm as follows:

$$\|\nabla d\|_{\alpha_d} = \begin{cases} \frac{|\nabla d|^2}{2\alpha_d}, & \text{if} |\nabla d| \leq \alpha_d \\ |\nabla d| - \frac{\alpha_d}{2}, & \text{if} |\nabla d| > \alpha_d \end{cases}. \tag{6.9}$$

The overall energy function is then given by

$$\begin{aligned} E(\mathbf{g}, \mathbf{d}) &= E_{reg}(\mathbf{g}, \mathbf{d}) + \lambda E_{data}(\mathbf{g}, \mathbf{d}) \\ &= \sum_{\forall x,y} \|\nabla d\|_{\alpha_d} + \lambda \rho(g, d). \end{aligned} \tag{6.10}$$

Before investigating a solution of the proposed energy function, an analysis of the function is presented in the next section.

6.3 Analysis of Energy Function

The convergence of the proposed cost function needs to be discussed before its solution is studied. The regularization cost in Equation (6.10) is same as the regularization cost used in the previous chapters which is easy to optimize. On the other

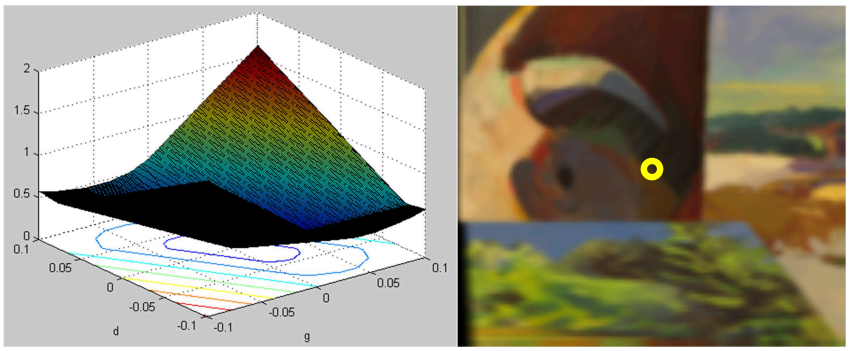


Figure 6.3: An example of the shape of pixel-wise data cost $\rho(g, d)$ at an image edge (indicated by the yellow circle).

hand, the data cost is composed of two derivative terms $g^L_{j,d}$ and $I^L_{j,d}$, thus the convexity may differ from that of the previous problems. The theoretic analysis of the cost function is difficult because the cost function differs from pixel to pixel. Thus, the empirical analysis is alternatively performed. Figure 6.3 presents an example shape of the pixel-wise data cost $\rho(g, d)$ at an image edge from four images. The example shows that the data cost is balanced with depth d and pixel intensity g and their optimum point can be easily found. The first-order primal-dual algorithm is employed again, and the proposed cost function is solved by using the coarse-to-fine approach. The data cost is a sum of L^1 norms, thus there can exist several critical points in the cost function and the critical point search is used in the update of a primal variable.

6.4 Experiments

The proposed deblurring and super-resolution combined method is evaluated with synthesized image sequences as well as real image sequences. The deblurring and

super-resolution results for each data are compared with the results by methods that deblurring and super-resolution are sequentially performed.

6.4.1 Synthesized data

The synthetic data is obtained by warping a reference image using its ground truth depth map, and integrating the warped images for a specified exposure time, and Gaussian blurring and downsampling are applied in the warping process. Due to the downscale operation, the degree of motion blur is weekly observed in the synthesized low-resolution image, but it still makes the depth estimation difficult.

The results of high-resolution depth map and image estimation of synthetic data *Bull* and *Cloth* sequence are presented in Figure 6.4 and Figure 6.5, respectively, compared with the depth maps obtained from low-resolution images without motion blur model and upscaled input images by bicubic interpolation. We can see that without consideration of motion blur for low-resolution input images, the resulting depth maps have severe errors. Although the parameter λ which controls the smoothness of depth map increases, the errors of depth map without consideration of motion blur do not decrease as shown in Figure 6.6.

The effectiveness of the proposed simultaneous blur kernel estimation and super-resolution method is tested by comparing the results with sequential methods that blur kernel estimation and super-resolution are performed sequentially. Two sequential implementations are tested here. The first method, denoted by *Seq. DB-SR*, performs depth estimation with motion blur model from Chapter 4 and deblurs low-resolution input images with estimated blur kernels. High-resolution depth map and image estimation from Chapter 5 then follows using the deblurred low-resolution images. Reversely, the second method denoted by *Seq. SR-DB*, performs super-

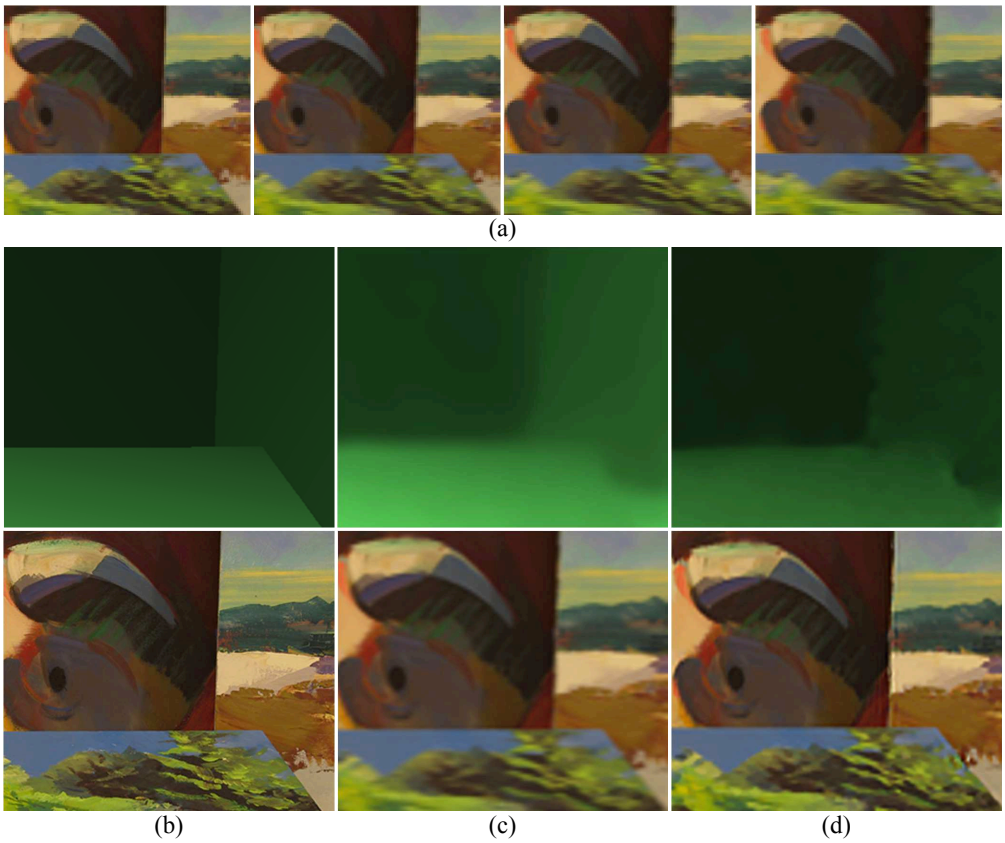


Figure 6.4: High-resolution depth and image estimation on synthetic data *Bull*: (a) Low-resolution blurred input images. (b) Ground truth depth map and image. (c) Low-resolution depth map without motion blur model and upscaled image using bicubic interpolation. (d) Results by the proposed method.

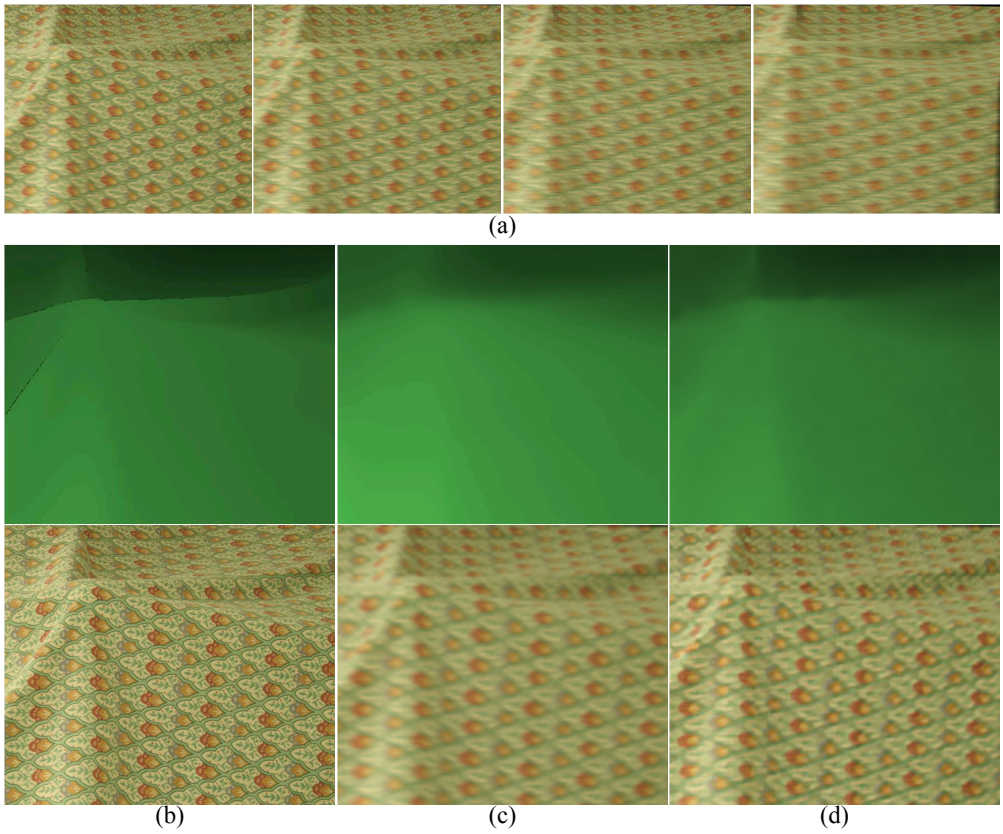


Figure 6.5: High-resolution depth and image estimation on synthetic data *Cloth*: (a) Low-resolution blurred input images. (b) Ground truth depth map and image. (c) Low-resolution depth map without motion blur model and upscaled image using bicubic interpolation. (d) Results by the proposed method.

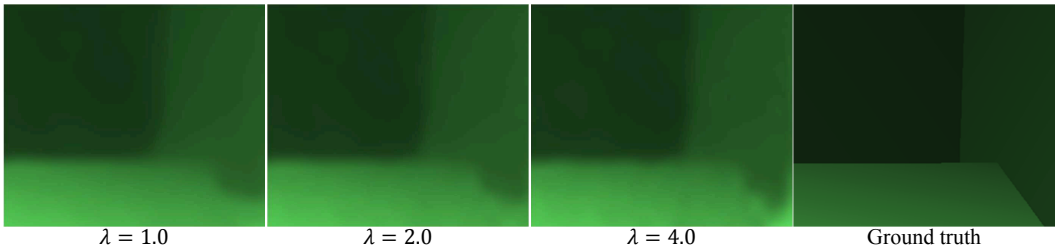


Figure 6.6: Depth map without consideration of motion blur is not improved although the smoothness parameter λ is tuned.

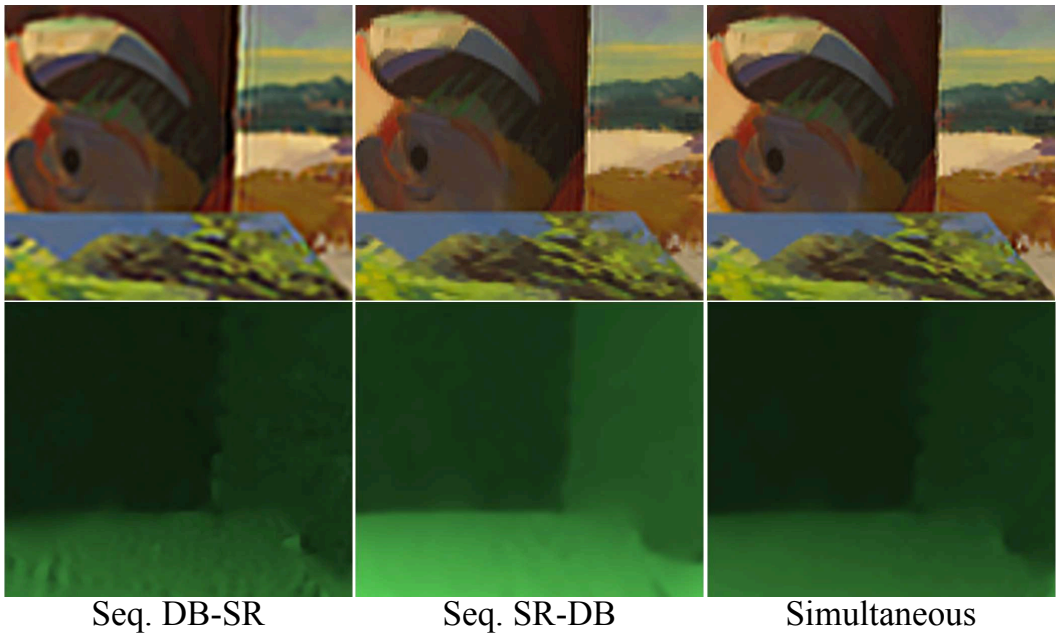


Figure 6.7: Comparison of high-resolution depth and image estimation by the sequential methods (*Seq. DB-SR Seq. SR-DB*) and the proposed simultaneous method.

resolution first and then performs deblurring later. The result of each method is shown in Figure 6.7. *Seq. DB-SR* suffers from artifacts of deblurring low-resolution image in its super-resolution task, and *Seq. SR-DB* has difficulties in finding accurate correspondence of pixels since it does not consider the effect of motion blur. On the other hand, the proposed method provides better results by considering both motion blur and super-resolution models in its depth estimation.

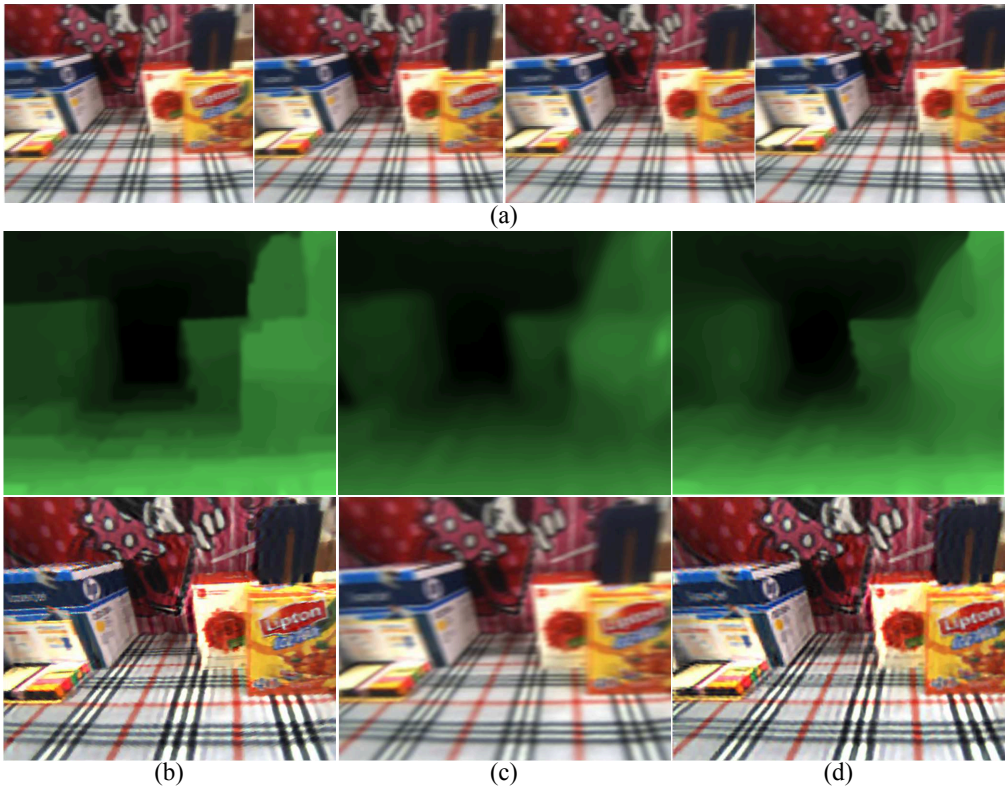


Figure 6.8: High-resolution depth and image estimation on real image sequence *Desk*: (a) Low-resolution blurred input images. (b) Depth map and deblurred image using original high-resolution images. (c) Low-resolution depth map without motion blur model and upscaled image using bicubic interpolation. (d) Results by the proposed method.

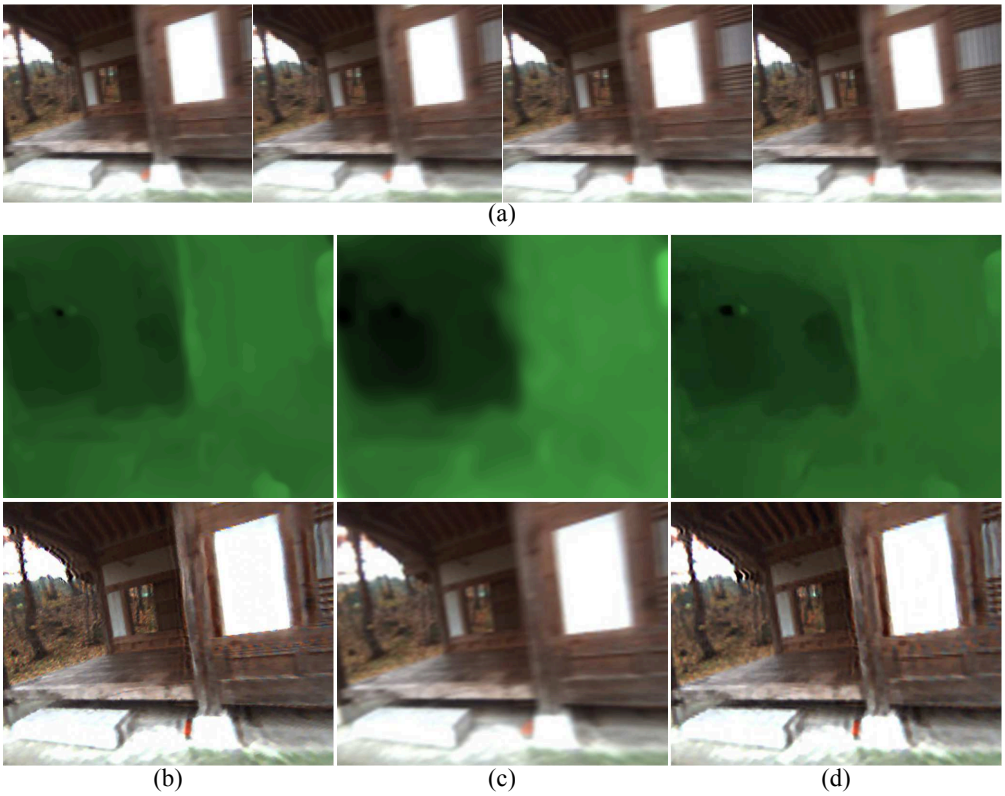


Figure 6.9: High-resolution depth and image estimation on real image sequence *House*: (a) Low-resolution blurred input images. (b) Depth map and deblurred image using original high-resolution images. (c) Low-resolution depth map without motion blur model and upscaled image using bicubic interpolation. (d) Results by the proposed method.

6.4.2 Real data

The proposed method is also tested by using real image data. The low-resolution sequences are obtained by downsampling the real blurry image sequences used in Chapter 4. The input images and results for *Desk* and *House* data set are presented

in Figure 6.8 and Figure 6.9, respectively. With the proposed super-resolution combined motion blur-aware 3D reconstruction, the results of reconsturction are much better than the reconstruction results without considering motion blur and super-resolution model as compared in (c) and (d) of Figure 6.8 and Figure 6.9. The deblurring results of proposed method are very close to deblurring results by the original high-resolution images.

6.5 Summary

The 3D reconstruction method that can address both motion blur and low-resolution problem is presented. The modeling of motion blur and super-resolution is effectively unified by a single optimization framework to estimate high-resolution image, depth map, and blur kernels for deblurring. Different from the image enhancement results by sequentially performing deblurring and super-resolution, the results by the proposed does not suffer from the error amplification in early stage.

Chapter 7

Conclusion

7.1 Summary of Dissertation

In this dissertation, various 3D reconstruction methods combined with image enhancement have been presented. In particular, handling motion blur and low image resolution is addressed for both sparse point-based 3D reconstruction and dense depth reconstruction. It is apparent that image quality degradation is an important issue in 3D reconstruction, but conventional image enhancement methods have rarely been applied to 3D reconstruction systems due to their high computational costs. The proposed methods in this dissertation utilizes 3D geometry of camera and target scenes to obtain information for enhancing degraded input images such as blur kernels for motion deblurring and and pixel correspondences for super-resolution. This geometric information makes the 3D reconstruction methods robust to those image degradation factors, and makes fast and accurate image enhancement possible.

In Chapter 2, the motion blur problem is addressed for sparse point-based reconstruction, and the effective blur-robust data association is proposed. The approx-

imation method for motion blurred appearances of landmark patches is incorporated with the 3D geometry estimation in visual SLAM, and the mapped landmarks are then robustly matched even with blurred images. Furthermore, non-uniformly blurred images are easily recovered by using the obtained kernel for each landmark, and new landmarks can be extracted and registered to the map. In Chapter 3, image super-resolution is incorporated with the visual SLAM system that locally planar landmarks are mapped to the map with their poses. The high-resolution patches are simultaneously estimated with landmark poses and camera pose via the Rao-Blackwellized particle filter framework. The super-resolution patches improve the accuracy of data association of landmarks, and the poses of camera and landmarks are also accurately estimated.

The deblurring and super-resolution approaches for sparse point-based 3D reconstruction are then extended to dense 3D reconstruction method. The geometric relationship between 3D reconstruction and image enhancement is applied in dense reconstruction, and the energy function for blur-aware depth map estimation is proposed in Chapter 4, and the energy function for simultaneous depth map and high-resolution image estimation is proposed in Chapter 5. Finally, the unified energy function for depth estimation with deblurring and super-resolution is proposed in Chapter 6. The proposed energy functions are effectively solved by the continuous optimization based on the first-order primal-dual algorithm, and the parallel implementation of the optimization enables a fast depth map generation which is essential for the single camera-based reconstruction. In the proposed depth estimation, blur kernels for deblurring or pixel registrations for super-resolution are simultaneously obtained with the depth map, and those are utilized for the fast enhancement of input images.

In addition to the accuracy improvement of 3D reconstruction by enhanced input images, the proposed geometry-aware image enhancement also has a performance gain compared with conventional image enhancement methods, with respect to enhancement accuracy as well as computation speed. The problems of traditional image enhancement methods, such as handling scene depth variation for blur kernel estimation and time-consuming pixel-wise correspondence estimation, are efficiently addressed by utilizing 3D geometry information. This complementary estimation of 3D reconstruction and image enhancement is the main contribution of this study.

7.2 Future Works

For sparse point-based 3D reconstruction, deblurring and super-resolution are applied to different visual SLAM systems depending on their characteristics. In the future works, deblurring for filtering-based SLAM and super-resolution for optimization-based SLAM should be investigated, respectively. The major difference of filtering-based SLAM compared to optimization-based SLAM is that a camera and landmarks in filtering-based SLAM have uncertainties in their poses, thus handling these uncertainties should be addressed. To apply super-resolution for optimization-based SLAM, another estimation method for high-resolution patch estimation instead of Kalman filter should be studied. Furthermore, more general approaches for deblurring and super-resolution that can be applied to any types of visual SLAM need to be investigated.

In the current study, a basic 3D reconstruction model which relies only on pixel correspondence is tested. More sophisticated models for 3D reconstruction can improve the accuracy of reconstruction result. For example, using the visibility and

occlusion model can provide more reliable depth estimation in object boundaries as well as improved image enhancement results. In super-resolution, the downsampling and blurring model is 3D geometry-dependent, but a simple constant model is used in the current implementation. If 3D geometry is considered in the downsampling and blurring model, then more accurate super-resolution results can be obtained.

The proposed image enhancement method can be used not only for 3D reconstruction system, but also for general video enhancement if the camera intrinsic parameters are calibrated. Reversely, various advanced techniques for image enhancement for a single image or a video can be added to improve the result of proposed method. Therefore, incorporating both approaches is believed to be a valuable research.

Bibliography

[1] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of Alvey Vision Conference*, 1988.

[2] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 27, no. 3, 2008.

[3] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-unifrom deblurring for shaken images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

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

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

[6] H. Takeda and P. Milanfar, "Removing motion blur with space-time processing," *IEEE Transactions on Image Processing*, vol. 20, no. 10, 2011.

[7] D. Scharstein and R. Szelisk, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, 2002.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, 2007.

[9] B. Williams, G. Klein, and I. Reid, "Real-time SLAM relocalisation," in *Proceedings of IEEE International Conference on Computer Vision*, 2007.

[10] M. Pollefeys, D. Nister, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. N. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3D reconstruction from video," *International Journal of Computer Vision*, vol. 78, no. 2-3, 2008.

[11] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[12] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*, 2007.

[13] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "RSLAM: A system for large-scale mapping in constant-time using stereo," *International Journal of Computer Vision*, vol. 94, no. 2, 2010.

[14] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[15] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - A modern synthesis," in *Vision Algorithms: Theory and Practice*, 2000.

[16] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," in *Proceedings of European Conference on Computer Vision*, 2008.

[17] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello, "A visual odometry framework robust to motion blur," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2009.

[18] U. Mudenagudi, A. Gupta, L. Goel, A. Kushal, P. Kalra, and S. Banerjee, "Super resolution of images of 3D scenecs," in *Proceedings of Asian Conference on Computer Vision*, 2007.

[19] A. V. Bhavsar and A. Rajagopalan, "Resolution enhancement in multi-image stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, 2010.

[20] T. Tung, S. Nobuhara, and T. Matsuyama, "Simultaneous super-resolution and 3D video using graph-cuts," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[21] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of Conference on Uncertainty in Artificial Intelligence*, 2000.

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

[23] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, 2011.

[24] E. Esser, X. Zhang, and T. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *Journal on Imaging Sciences*, vol. 3, no. 4, 2010.

[25] J. Lellmann, D. Breitenreicher, and C. Schnörr, "Fast and exact primal-dual iterations for variational problems in computer vision," in *Proceedings of European Conference on Computer Vision*, 2010.

[26] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway, "Robust real-time visual SLAM using scale prediction and exemplar based feature description," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[27] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 25, no. 3, 2006.

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

[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, 2004.

[30] P. Favaro, M. Burger, and S. Soatto, "Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion," in *Proceedings of European Conference on Computer Vision*, 2004.

[31] C. Paramanand and A. N. Rajagopalan, "Unscented transformation for depth from motion-blur in videos," in *Proceedings of IEEE Workshop on Three Dimensional Information Extraction for Video Analysis and Mining in conjunction with CVPR 2010*, 2010.

[32] S. Gauglitz, T. Hollerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International Journal of Computer Vision*, vol. 94, no. 3, 2011.

[33] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical Journal*, vol. 79, no. 6, 1974.

[34] S. Dai and Y. Wu, "Motion from blur," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[35] J. Jia, "Single image motion deblurring using transparency," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[36] T. Drummond and R. Cipolla, "Application of lie algebras to visual servoing," *International Journal of Computer Vision*, vol. 37, no. 1, 2000.

[37] H. S. Lee, J. Kwon, and K. M. Lee, "Simultaneous localization, mapping, and deblurring," in *Proceedings of IEEE International Conference on Computer Vision*, 2011.

[38] V. Gorbatsevich, A. Onishchik, and E. Vinberg, *Foundations of Lie Theory and Lie Transformation Groups*. Springer, 1997.

[39] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of European Conference on Computer Vision*, 2006.

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

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

[42] Y. Park, V. Lepetit, and W. Woo, "ESM-blur: Handling & rendering blur in 3D tracking and augmentation," in *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*, 2009.

[43] S. Baker, R. Gross, and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, 2004.

[44] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

[45] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[46] R. Fransens, C. Strecha, and L. V. Gool, "Optical flow based super-resolution: A probabilistic approach," *Computer Vision and Image Understanding*, vol. 106, no. 1, 2007.

[47] C. Liu and D. Sun, "A bayesian approach to adaptive video super resolution," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[48] M. Irani and S. Peleg, "Improving resolution by image registration," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, vol. 53, no. 3, 1991.

[49] M. Elad and A. Feuer, "Super-resolution reconstruction of image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, 1999.

[50] F. Dellaert, C. Thorpe, and S. Thrun, "Super-resolved texture tracking of planar surface patches," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

[51] B. B. Ready, C. N. Taylor, and R. W. Beard, "A Kalman-filter based method for creation of super-resolved mosaicks," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2006.

[52] Y. K. Yu, S. H. Or, K. H. Wong, and M. M. Y. Chang, "Accurate 3D motion tracking with an application to super-resolution," in *Proceedings of IEEE International Conference on Pattern Recognition*, 2006.

[53] K. Murphy, "Bayesian map learning in dynamic environments," in *Proceedings of Advances in Neural Information Processing Systems*, 1999.

[54] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[55] S. J. Julier, Jeffrey, and K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, 2004.

[56] F. Dellaert, S. Thrun, and C. Thorpe, "Jacobian images of super-resolved texture maps for model-based motion estimation and tracking," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 1998.

[57] G. Graber, T. Pock, and H. Bischof, "Online 3D reconstruction using convex optimization," in *1st Workshop on Live Dense Reconstruction from Moving Cameras. In conjunction with ICCV 2011*, 2011.

[58] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

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

[60] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Proceedings of German Association for Pattern Recognition (DAGM) Conference on Pattern Recognition*, 2010.

[61] S. Cho, J. Wang, and S. Lee, "Video deblurring for hand-held cameras using patch-based synthesis," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 31, no. 4, 2012.

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

[63] O. Shahar, A. Faktor, and M. Irani, "Space-time super-resolution from a single video," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[64] S. Cho, H. Cho, Y.-W. Tai, and S. Lee, "Registration based non-uniform motion deblurring," *Computer Graphics Forum*, vol. 31, no. 7, 2012.

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

[66] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 29, no. 4, 2010.

[67] L. Xu and J. Jia, "Depth-aware motion deblurring," in *Proceedings of IEEE International Conference on Computational Photography*, 2012.

[68] C. Paramanand and A. N. Rajagopalan, "Inferring image transformation and structure from motion-blurred images," in *Proceedings of British Machine Vision Conference*, 2010.

[69] A. Rav-Acha and S. Peleg, "Two motion-blurred images are better than one," *Pattern Recognition Letters*, vol. 26, no. 3, 2005.

[70] J. Civera, A. J. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, 2008.

[71] P. J. Huber, "Robust regression: Asymptotics, conjectures and monte carlo," *The Annals of Statistics*, vol. 1, no. 5, 1973.

[72] Y. W. Tai, P. Tan, and M. S. Brown, "Richardson-Lucy deblurring for scenes under a projective motion path," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, 2011.

[73] B. Goldlucke and D. Cremers, "A super-resolution framework for high-accuracy multiview reconstruction," in *Proceedings of German Association for Pattern Recognition (DAGM) Conference on Pattern Recognition*, 2009.

[74] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, 2001.

[75] M. Unger, T. Pock, M. Werlberger, and H. Bishof, "A convex approach for variational super-resolution," in *Proceedings of German Association for Pattern Recognition (DAGM) conference on Pattern recognition*, 2010.

[76] Q. Shan, Z. Li, J. Jia, and C.-K. Tang, "Fast image/video upsampling," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, vol. 27, no. 5, 2008.

[77] Video Enhancer, "http://www.infognition.com/videoenhancer/," 2012, version 1.9.7.

[78] B. Bascle, A. Blake, and A. Zisserman, "Motion deblurring and super-resolution from an image sequence," in *Proceedings of European Conference on Computer Vision*, 1996.

[79] E. Shechtman, Y. Caspi, and M. Irani, "Space-time superresolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, 2005.

국문 초록

영상 기반 3차원 복원은 컴퓨터 비전의 기본적인 연구 주제 가운데 하나로 최근 몇 년간 많은 발전이 있어왔다. 특히 자동 로봇을 위한 네비게이션 및 휴대 기기를 이용한 증강 현실 등에 널리 활용될 수 있는 단일 카메라를 이용한 3차원 복원 기법은 복원의 정확도, 복원 가능 범위 및 처리 속도 측면에서 많은 실용 가능성을 보여주고 있다. 그러나 그 성능은 여전히 조심스레 촬영된 높은 품질의 입력 영상에 대해서만 시험되고 있다. 움직이는 단일 카메라를 이용한 3차원 복원의 실제 동작 환경에서는 입력 영상이 화소 잡음이나 움직임에 의한 번짐 등에 의하여 손상될 수 있고, 영상의 해상도 또한 정확한 카메라 위치 인식 및 3차원 복원을 위해서는 충분히 높지 않을 수 있다. 많은 연구에서 고성능 영상 화질 향상 기법들이 제안되어 왔지만 이들은 일반적으로 높은 계산 비용을 필요로 하기 때문에 실시간 동작 능력이 중요한 단일 카메라 기반 3차원 복원에 사용되기에는 부적합하다.

본 논문에서는 보다 정확하고 안정된 복원을 위하여 영상 개선이 결합된 새로운 단일 카메라 기반 3차원 복원 기법을 다룬다. 이를 위하여 영상 품질이 저하되는 중요한 두 요인인 움직임에 의한 영상 번짐과 낮은 해상도 문제가 각각 점 기반 복원 및 조밀 복원 기법들과 결합된다. 영상 품질 저하를 포함한 영상 획득 과정은 카메라 및 장면의 3차원 기하 구조와 관측된 영상 사이의 관계를 이용하여 모델링 할 수 있고, 이러한 영상 품질 저하 과정을 고려함으로써 정확한 3차원 복원을 하는 것이 가능해진다. 또한, 영상 번짐 제거를 위한 번짐 커널 또는 영상의 초해상도 복원을 위한 화소 대응 정보 등이 3차원 복원 과정과 동시에 얻어지는것이 가능하

여, 영상 개선이 보다 간편하고 빠르게 수행될 수 있다. 제안되는 기법은 3차원 복원과 영상 개선 문제를 동시에 해결함으로써 각각의 결과가 상호 보완적으로 향상된다는 점에서 그 장점을 가지고 있다. 본 논문에서는 실험적 평가를 통하여 제안되는 3차원 복원 및 영상 개선의 효과성을 입증하도록 한다.

주요어: 영상 기반 3차원 복원, 비주얼 슬램, 영상 개선, 영상 번짐 제거, 초해상도 영상 복원.

학번: 2006-21271

감사의 글

박사는 스스로 연구할 능력이 있는 사람이라고 들어왔는데 학위 논문의 마지막 페이지를 쓰고 있는 지금도 아직 제가 그러한 자격이 있는지는 의문이 듭니다. 그럼에도 부족한 저를 끊임없는 열정과 가르침으로 이끌어주신 이경무 교수님께서 계셨기에 무사히 학위 과정을 마치게 되지 않았나 생각합니다. 교수님으로부터 배운 연구자의 자세를 항상 감사하고 기억하여 연구실의 명성에 누가 되지 않도록 최선을 다하겠습니다. 미완의 학위 논문이 보다 온전해 질 수 있도록 여러 차례의 심사를 통하여 지도해주신 서울대 이상욱 교수님, 서강대 이상욱 교수님, 한양대 박종일 교수님과 임종우 교수님께도 고개숙여 감사드립니다.

컴퓨터 비전이라는 학문이 매력적이기도 하였지만 7년간의 연구실 생활이 웃음으로 가득할 수 있었던 것은 가족같은 연구실 선후배님들 덕분이었습니다. 정말 많은 분들과 함께 하였습니다. 현목형, 우연형, 영기형, 민수형, 준영이형, 동우형, 준석형, 영민형, 효찬형, 주용형, 정현형, 원식이, 정민이, 태현이, 희수, 상돈이, 효진이, 준하, 유민이, 병주, 광모, 명섭이, 장훈이 까지, 모두들 연구에 몰두하느라 학교에 있는 동안 유흥의 시간을 많이 가지지 못한 것이 아쉽지만 앞으로 자주 만나뵈면서 그 아쉬움을 풀 수 있었으면 좋겠습니다. 세미나 시간 등을 통하여 좋은 말씀 자주 해주신 윤일동 교수님, 박인규 교수님, 그리고 함께 연구해서 든든하였던 신호처리 연구실의 선후배님들께도 고마운 마음을 전하고 싶습니다.

제 오랜 학업의 마침을 누구보다 기뻐하실 분은 아버지, 어머니가 아닐까 생각합니다. 두 분의 저에 대한 믿음과 희생은 어떤 말로도 감사하기 힘들 것 같습니다.

이제 그 은혜에 조금이나마 보답할 수 있도록 노력하겠습니다. 또한 항상 든든한 힘이 되어 준 누나와 매형에게도 고맙다는 말 전하고 싶습니다. 끝으로 대학원 기간 동안 언제나 옆에서 응원해주고 생활의 활력이 되어 준 혜진이, 그 동안 바쁘다는 핑계로 늘 소홀했던 것에 미안하고, 이제는 제가 더 많은 힘이 되어 줄 수 있도록 하겠습니다.

그 동안 도움주신 다른 모든 분들께 다시 한 번 감사드리며, 그 보답의 길은 보다 세상에 이로운 연구를 하는 것이라 다짐하며 감사의 글을 마칩니다.