

Simultaneous Localization, Mapping and Deblurring

Hee Seok Lee

Junghyun Kwon

Kyoung Mu Lee

Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea

ultra21@snu.ac.kr

junghyunkwon@gmail.com

kyoungmu@snu.ac.kr

<http://cv.snu.ac.kr>

Abstract

Handling motion blur is one of important issues in visual SLAM. For a fast-moving camera, motion blur is an unavoidable effect and it can degrade the results of localization and reconstruction severely. In this paper, we present a unified algorithm to handle motion blur for visual SLAM, including the blur-robust data association method and the fast deblurring method. In our framework, camera motion and 3-D point structures are reconstructed by SLAM, and the information from SLAM makes the estimation of motion blur quite easy and effective. Reversely, estimating motion blur enables robust data association and drift-free localization of SLAM with blurred images. The blurred images are recovered by fast deconvolution using SLAM data, and more features are extracted and registered to the map so that the SLAM procedure can be continued even with the blurred images. In this way, visual SLAM and deblurring are solved simultaneously, and improve each other's results significantly.

1. Introduction

Motion blur is often an undesired phenomenon in recording images or videos. Especially in visual Simultaneous Localization and Mapping (SLAM), where a camera is moved 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 features.

Many recent visual SLAM systems can handle localization failures caused by motion blur by applying relocalization (global localization) algorithms [3, 21] after the motion blur. However, when the camera explores through a region that the camera has not visited and whose reconstruction has not been done, 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 SLAM

processes, including data association as well as mapping new landmark under motion blur.

Generally, in visual SLAM systems, many tasks on images are performed with detected interest points, such as registering new features as landmarks, or finding matching features of reconstructed landmarks. General point feature detectors, however, cannot give enough features from a blurred image. With a blurred image, moreover, feature matching between frames becomes difficult and the matching accuracy decreases. Clearly, 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 [5, 18, 20, 7], but most require a large computational budget, thus it is hard to use those methods to recover images for visual SLAM.

In this paper, we propose a visual SLAM algorithm combined with image deblurring. By considering motion blur, data association in SLAM can be greatly enhanced, and camera localization can also be performed robustly even with blurred scenes. The information obtained from SLAM are used to estimate motion blur kernels, which are then used to deblur the 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 the original blurred image. As a result, localization and mapping can be performed successfully under the blur situation.

1.1. Related work

Motion blur is an important factor which degrades the performance of visual SLAM, but there have been few studies on the methods for handling motion blur. In [16], 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 extracting interest points and building image descriptors based on SIFT. Although their method does not require explicit deblurring, the computation time is not adequate for real-time operation. They pointed out that deblurring based on deconvolution might worsen the image, 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 our approach, however, small errors in PSF are acceptable if we deblur small size of patches in which the blur is almost uniform.

[11] tried to solve the blurring effect in visual SLAM 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. Motivated by this observation, they presented a tracking method using edgelets and made their visual SLAM system to be robust to motion blur. In [11], however, 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 or edgelet for localization will be available in the map.

2. Motion Blur and 3-D geometry

In this section, we will discuss the motion blur and its relationship with the camera motion and the 3-D structure of the scene. Then the method for building blur kernel from 3-D geometry will be presented.

2.1. Motion deblurring

A general method for removing motion blur is to estimate a blur kernel and recover a sharp image by deconvolution. In blur kernel estimation, a simple but efficient approach is to assume a spatially uniform kernel for the entire image [5, 18]. 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 local blur kernels for image regions and deblur each part, but it is very computationally expensive.

Using the 3-D 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 [20, 7]. However, the 3-D structure of the scene is not considered in [20, 7], while the depth of scene point is highly correlated to blur kernel. Other non-uniform blur kernel estimation methods for a single image are based on image properties (e.g., α -channel [4], transparency [8]), but those methods can be applied only if foreground object and background scene can be distinguished. In visual SLAM, on the other hand, a camera motion and 3-D point structure of the scene are continuously estimated, then we can easily calculate the blur kernel for each individual scene point using those estimates.

2.2. Motion blur and 3-D geometry

If the exposure time of a camera is not infinitesimal and the camera moves fast, lights (photons) from an object are

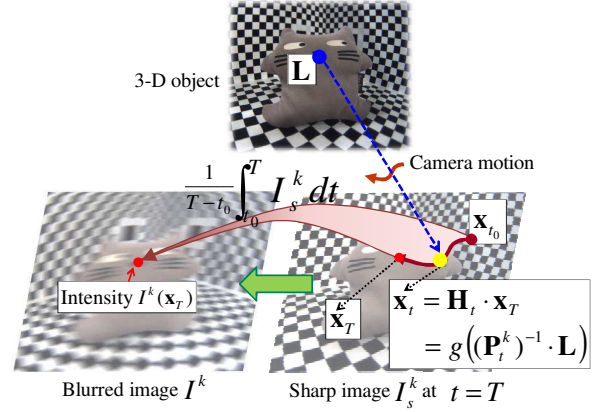


Figure 1. Movement of the projected point caused by camera motion.

not projected to a single image point. Rather, they make a ‘line’, and motion blur occurs. Figure 1 illustrates the projection process of a 3-D point by a moving camera. We use a superscript k for a frame index, and a subscript t for time in capturing one image. Let \mathbf{L} and \mathbf{x}_t be a 3-D scene position and its projected point, respectively, with homogeneous coordinates. During exposure time $t \in [t_0, T]$, the projected point \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 dependent on the 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

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

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 (non-blurred) image at $t = T$.

The pixel \mathbf{x}_t and the 3-D scene point \mathbf{L} are related by the camera pose \mathbf{P}_t^k with the equation $\mathbf{x}_t = g((\mathbf{P}_t^k)^{-1} \cdot \mathbf{L})$. The camera pose \mathbf{P}_t^k is defined on the Special Euclidean group $SE(3)$, which represents the rigid transformation of camera composed of 3-D translation and 3-D rotation from the origin [12], and $g(\cdot)$ is a perspective camera projection function with camera intrinsic parameters. Then, we can rewrite the equation (1) as

$$I^k(\mathbf{x}_T) = \int_{t_0}^T \frac{1}{T - t_0} I_s^k(g((\mathbf{P}_t^k)^{-1} \cdot \mathbf{L})) dt. \quad (2)$$

In the motion deblurring algorithm based on the convolution model, the blur kernel \mathcal{K} is inferred from the homography \mathbf{H} , then \mathcal{K} is used to deconvolve the blurred image.

In a general situation of blur, the motion of a pixel, equivalently \mathbf{H} or \mathcal{K} , is not given. This problem is called blind deconvolution, and complicated methods are used to estimate both the blur kernel \mathcal{K} and the sharp image I_s . On the other hand, if we have the estimates of \mathbf{P}_t and \mathbf{L} , then the kernel \mathcal{K} can be easily obtained and the problem becomes non-blind, which is simpler and faster to solve than the blind deconvolution.

2.3. Blur kernel from SLAM Data

For our visual SLAM purpose, we make two assumptions on the blur kernel. First, we assume that the blur kernel is a straight line. Many recent visual SLAM systems ensure a frame rate of at least 10 Hz for the real-time performance. The exposure time does not exceed the frame interval, and within this short exposure time the movement of a pixel can be approximated as a straight line. By this assumption, we can parameterize the blur kernel with a magnitude ℓ and a direction ϕ of the blur, representing \mathcal{K} as $\mathcal{K}(\ell, \phi)$. The magnitude is defined by the distance between \mathbf{x}_{t_0} and \mathbf{x}_T , and the direction is defined by the angle between the vector $[\mathbf{x}_T - \mathbf{x}_{t_0}]$ and the horizontal axis of the image.

The second assumption is that the blur kernel is locally uniform. Although the blur kernel is different depending on a 3-D object position, pixels in a small object area have similar blur aspects. Therefore, we estimate the kernels for landmark points mapped by SLAM, then we use this kernel to deblur the adjacent regions of the landmark's 2-D point.

To calculate the magnitude and direction of the blur, \mathbf{x}_T and \mathbf{x}_{t_0} are needed to be estimated, which is difficult. Alternatively, we use the camera pose \mathbf{P} and 3-D landmark position \mathbf{L} to acquire the kernel. Here, we assume that \mathbf{P} and \mathbf{L} are already known, and we will discuss how to estimate them later.

Let $\tilde{\mathbf{x}}_t$ be the non-homogeneous representation of \mathbf{x}_t , and $h(\cdot)$ be a mapping function such that $\tilde{\mathbf{x}}_t = h(\mathbf{x}_t)$. Since the pixel $\tilde{\mathbf{x}}_t$ moves as the camera pose \mathbf{P}_t^k changes, a derivative

$$\left. \frac{d\tilde{\mathbf{x}}_t}{dt} \right|_{t=t_0} = \frac{dh(g((\mathbf{P}_t^k)^{-1} \cdot \mathbf{L}))}{d\mathbf{P}_t} \frac{d\mathbf{P}_t}{dt} \Big|_{t=t_0} \equiv [u, v]^T \quad (3)$$

is an instantaneous direction of pixel $\tilde{\mathbf{x}}_t$ at time $t = t_0$. The derivative $\frac{d\mathbf{P}_t}{dt}$ corresponds to the camera velocity. From the straight line assumption of the blur kernel, $\frac{d\tilde{\mathbf{x}}_t}{dt} = \frac{d\tilde{\mathbf{x}}_t}{dt} \Big|_{t=t_0}$

holds for all $t \in [t_0, T]$, and the calculation of $\frac{d\tilde{\mathbf{x}}_t}{dt} \Big|_{t=t_0}$ can be simplified using an inter-frame difference $\tilde{\mathbf{x}}_T^k - \tilde{\mathbf{x}}_{T-1}^{k-1}$, as

$$\begin{aligned} \left. \frac{d\tilde{\mathbf{x}}_t}{dt} \right|_{t=t_0} &= \frac{\tilde{\mathbf{x}}_T^k - \tilde{\mathbf{x}}_{T-1}^{k-1}}{T} \\ &= \frac{h(g((\mathbf{P}_T^k)^{-1} \cdot \mathbf{L})) - h(g((\mathbf{P}_{T-1}^{k-1})^{-1} \cdot \mathbf{L}))}{T}. \end{aligned} \quad (4)$$

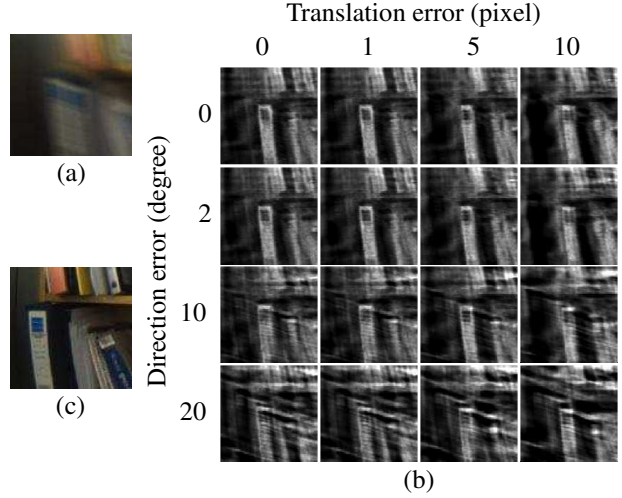


Figure 2. Results of deblurring in a presence of kernel error. (a) A blurred image. (b) Deblurred images by kernels with different translation and direction error. (c) A sharp (non-blurred) image taken at different moment for comparison.

This derivative is used for a direction ϕ of the kernel, such that $\phi = \arctan(\frac{v}{u})$.

The magnitude ℓ of the kernel is dependent on the exposure time. The exposure time information is available in general digital imaging devices. The length of the blur is defined as $\ell = \int_{t_0}^T \left| \frac{d\tilde{\mathbf{x}}_t}{dt} \right| dt$, and from the assumption on kernel again, the equation

$$\ell = \int_{t_0}^T \left| \frac{d\tilde{\mathbf{x}}_t}{dt} \right| dt = \left| \frac{d\tilde{\mathbf{x}}_t}{dt} \right| \cdot (T - t_0) \quad (5)$$

holds. The calculation of magnitude is also simplified using the inter-frame difference $\tilde{\mathbf{x}}_T^k - \tilde{\mathbf{x}}_{T-1}^{k-1}$. By plugging the equation (4) to (5), we get

$$\ell = \left| h(g((\mathbf{P}_T^k)^{-1} \cdot \mathbf{L})) - h(g((\mathbf{P}_{T-1}^{k-1})^{-1} \cdot \mathbf{L})) \right| \cdot \frac{(T - t_0)}{T}. \quad (6)$$

The meaning of the equation (6) is that while the projection $\tilde{\mathbf{x}}$ starts from $\tilde{\mathbf{x}}_{T-1}^{k-1}$ and go to $\tilde{\mathbf{x}}_T^k$, only lights during exposure time are captured in an image and they make a motion blur.

2.4. Reconstruction error and blur kernel error

One problem to be considered in this kernel estimation approach is that the 3-D reconstruction error can influence the accuracy of the blur kernel. Landmarks with large reconstruction error that usually arises from wrong data associations can be handled by outlier rejection. However, landmarks with a small error due to measurement noise may not be filtered out by the outlier rejection method, and this might affect the accuracy of the blur kernel estimation.

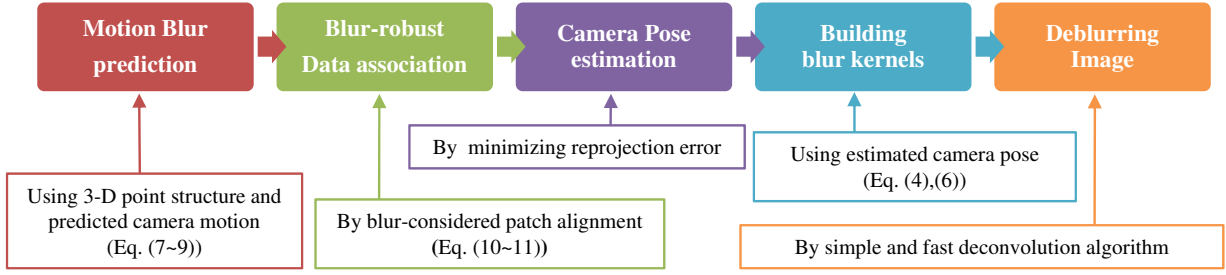


Figure 3. Overall procedure of the proposed algorithm.

When a landmark is reconstructed from N number of observations (images) with measurement noise ϵ_n ($n = 1, \dots, N$), the 3-D position \mathbf{L} of the landmark is determined by minimizing the reprojection errors for all measurements. Then \mathbf{L} will be projected into next frames with the expected reprojection error $\sqrt{\frac{1}{N} \sum_{n=1}^N \epsilon_n^2}$, and the blur kernel made by the equation (4) and (6) is affected by this error. Empirically, we observed that the result of deblurring is not significantly affected by the kernel error whenever the landmark reconstruction error is only induced by the measurement noises. Figure 2 shows the results of various kernel errors. In consideration that the measurement error does not exceed 2 pixels in general visual SLAM, the results of deblurring in the presence of such kernel errors is acceptable.

3. Visual SLAM and Deblurring

Figure 3 summarizes the overall procedure of the proposed algorithm. First, we predict a motion blur and approximate a blurred version of landmark's template to perform the blur-robust data association. After data association, the camera pose is refined. Finally, we build the blur kernels for each landmark and recover the deblurred image using the obtained kernel to conduct the remaining tasks of visual SLAM. Detailed algorithm is as follows.

3.1. Skeleton of the visual SLAM system

Our visual SLAM system is developed based on [10] which uses a parallel processing of localization and mapping, and the *keyframe* concept. 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 3-D landmarks are registered with their appearances in the form of small patches. To handle viewpoint changes of landmarks, the landmark patches are updated by affine warping calculated from the camera pose. In our blur-robust data association, the patches are additionally blur adjusted by predicted motion blur.

We use an image pyramid to extract point features because high-level (low-resolution) images are less sensitive to motion blur than low-level images. We use four levels of a pyramid and detect point features using FAST-10 [17] 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.

3.2. 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 [9], the image region tracking with blurred images is performed by blurring the template image, rather than deblurring the current blurred image. [15] improved the efficiency of [9] by approximating a blurred image using image derivatives. Those tracking methods are performed in the 2-D image space. On the other hand, using a 3-D structure, we can easily predict a motion blur using that information and use the predicted value as an initial value for the tracking to boost the tracking performances.

With the help of bundle adjustment, high accuracy reconstruction and localization can be achieved with non-blurred scenes even with a monocular camera. However, it is hard to estimate accurate \mathbf{P}^k when the image is blurred, since we use point features 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, we propose a blur-robust data association method as follows.

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

$$\mathbf{a}^k = a \log((\mathbf{P}_T^{k-1})^{-1} \cdot \mathbf{P}_T^k), \quad (7)$$

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

$$\widehat{\mathbf{P}}_T^k = \mathbf{P}_T^{k-1} \cdot \exp(\mathbf{a}^{k-1}). \quad (8)$$

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 stored appearances (8×8 patches in our implementation) of reconstructed landmarks. For successful matches, subpixel refinement using patch alignment algorithm such as inverse compositional algorithm [2] 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 few successfully associated landmarks.

To handle appearance differences between the stored landmark patches and the blurred patches in the current image, we generate a blurred version of the landmark patch using the approximation method presented in [15]. If the motion blur is assumed to be straight, the blurred image can be approximated by the second-order Taylor expansion of the warping function $\mathcal{W}(\Theta)$ and calculated fast. The warping function $\mathcal{W}(\Theta)$ is the geometric transform by the landmark patch's pose Θ between the previous frame and the current frame. The pose Θ can simply be modeled by 2-D position (translation) of patch center, or more parameters such as rotation and affine transformation. In our algorithm, we use 2-D translation for Θ , as $\Theta = [\theta_u, \theta_v]^\top = \tilde{\mathbf{x}}$, to simplify the algorithm and save on computational costs.

Let $\Delta\Theta$ be the motion vector of the patch pose between the previous frame and the current frame, and $I(\Delta\Theta)$ be the warped and blurred patch by the motion vector. Since we do not know the patch's exact position of current frame, we use the projected feature position $\hat{\Theta}^k = h(g((\widehat{\mathbf{P}}_T^k)^{-1} \cdot \mathbf{L}))$. Then the predicted vector of motion blur $\Delta\Theta_b$ is

$$\Delta\Theta_b = \frac{T - t_0}{T} \Delta\Theta = \frac{T - t_0}{T} (\hat{\Theta}^k - \Theta^{k-1}) \quad (9)$$

and $I(\Delta\Theta)$ can be approximated as

$$I(\Delta\Theta) \approx \mathcal{T} + a\mathcal{J}_\mathcal{T}\Delta\Theta + b\Delta\Theta^\top \mathcal{H}_\mathcal{T}\Delta\Theta, \quad (10)$$

where \mathcal{T} is a landmark patch from the deblurred keyframe where the landmark is registered. Constants a and b are related to the exposure time as follows:

$$a = \frac{t_0 + T}{2T}, \quad b = \frac{T^2 + Tt_0 + t_0^2}{3T^2}. \quad (11)$$

Matrices $\mathcal{J}_\mathcal{T}$ and $\mathcal{H}_\mathcal{T}$ are the Jacobian and the Hessian of the patch \mathcal{T} . With this approximately blurred patch, we set the sliding window around the projected position of each landmark and find the feature's observed position. We do not change $I(\Delta\Theta)$ during the sliding window search and use same $I(\Delta\Theta)$ for all pixel positions to save on computational cost. Since searching with the fixed $I(\Delta\Theta)$ gives a rough searching result, a fine search is needed described as follows.

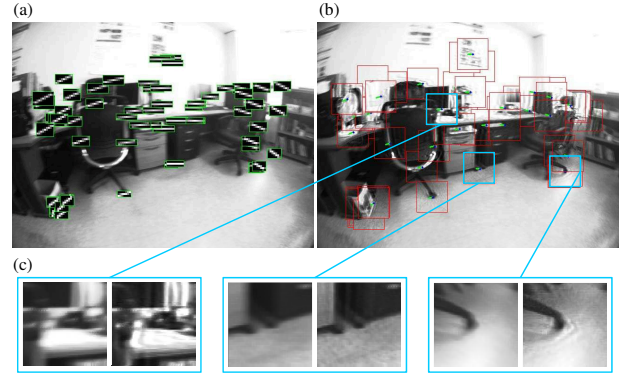


Figure 4. Example of deblurred patches. (a) Blur kernels at each landmark. (b) A partially deblurred image. (c) Close-up of patches (left: input, right: deblurred)

For roughly matched landmarks, we need to compensate for the effect of changing $\Delta\Theta$ and refine the position to sub-pixel accuracy. Since the image is blurred, we cannot use the conventional patch alignment algorithm. Instead we estimate the accurate patch position by the blur-robust version [15] of Efficient Second-order Minimization (ESM) [14] tracking algorithm. We refine the positions of landmark patches with ESM iteration, then successfully matched and refined landmarks with sub-pixel accuracy are obtained and will be used to estimate the accurate camera pose.

3.3. Blur-handled visual SLAM

After the blur-robust data association described in the previous section, we have to filter the data association outliers since ESM does not guarantee the result to be global optimum. Any types of outlier filtering methods such as RANSAC can be used, but simple thresholding based on the reprojection error is sufficient in our case.

After the outlier rejection, we calculate the new camera pose \mathbf{P}_T^k by minimizing the reprojection errors with a set of inlier matches. Then using the kernel estimation method described in Section 2, the blur kernel for each landmark with successful data association is obtained and image deblurring can be easily done using those kernels. Figure 4 shows an example of estimated kernels at different landmarks and their deblurring results for small regions. We can deblur every input frame for further vision tasks such as scene recognition, or only deblur the keyframes when new keyframe is added to register new landmarks. In this work, we choose the second option, because we focus on the SLAM performance and have to save the computational cost.

To deblur a whole keyframe image, we divide the image into small subregions (64×64 patches in our experiments) and choose blur kernel for each subregion by selecting the nearest kernel from its center. Using selected kernels, we deblur the subregions by Lucy-Richardson (LR) deconvolution.

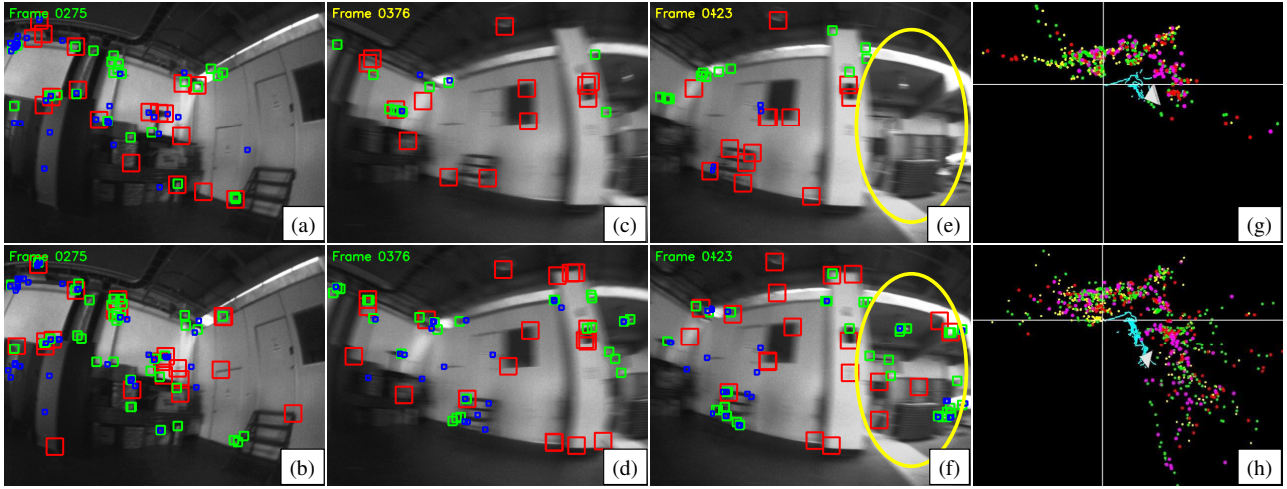


Figure 6. Data association and mapping of SLAM systems with (bottom row) and without (top row) the blur handling, respectively. Data association results of selected frames (a ~ f). Results of mapping by each system (g, h). The colors of landmarks in the scenes and the map represent the different levels of image pyramids where the landmarks are extracted.



Figure 5. Extracted FAST-10 corners from the blurred (left) and deblurred (right) image. Only points having ST measure higher than 50 are indicated by crosses (12 points in left, 78 points in right).

lution algorithm [13]. Although the LR deconvolution is simpler and faster than other non-blind deconvolutions, it is still slow for real-time processing for the SLAM system. Thus, we implement the deconvolution using general-purpose graphics processing units (GPGPUs), based on the implementation of [6]. We use *edgetaper* [1] and cut off the boundary regions to reduce the ringing artifacts. 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. Figure 5 shows an example of a deblurred image and detected FAST-10 corners with high cornerness values measured by the Shi and Tomasi (ST) cornerness measure [19]. In the deblurred image, 208 corners are extracted and their average ST measure is 57.56 for 7×7 window. While, in blurred image, 63 corners are detected and their average ST measure is 33.03. This means that by deblurring images, we can obtain much more *good features to track*. This is critical when the cam-

era 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 substantially, and sometimes the camera pose can be lost.

4. 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. We use Point grey research's Dragonfly 2 with the fish-eye lens of 160° field of view. The size of the input image is 640×480 , and all tasks are processed with gray scale images. The experiments are done on a 2.4GHz quad core PC and two threads (mapping thread and localization thread) run on each core at the same time. For the GPU-based deconvolution, we use NVIDIA's GeForce 9600GT with 512MB video memory.

When the blur-robust data association is activated, the average processing time for all localization processes is about 30ms per frame, while it takes 15ms with no blur handling. Thus the system ensures the frame rate of at least 30 Hz. The processing time for image deblurring is about 200ms, which is acceptable because the keyframes are added infrequently, and adding keyframe is done at the background thread. When the length of blur kernel is less than 2 pixels, we skip the deblurring and use original input image for mapping.

4.1. Performances of visual SLAM

We test the performance of our blur-handled visual SLAM algorithm by comparing it with conventional keyframe-based SLAM [10]. First, we experiment on the

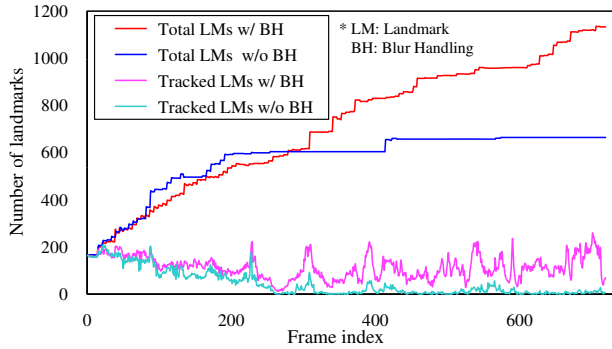


Figure 7. Comparison of the numbers of total landmarks in maps, and the numbers of currently tracked landmarks in each frame.

blur-robust data association. After the initial reconstruction and mapping for some frames, we move the camera rapidly to make motion blur. Without blur effect, both systems show good data association results as shown in Fig. 6-(a, b). When a motion blur occurs, the number of tracked landmarks decreases without the blur-robust data association (Fig. 6-(c, d)). When the camera observes unmapped region (yellow ellipse in Fig. 6-(e, f)) where a motion blur exists, no new landmark is registered to the map with the conventional system (Fig. 6-(e)), while our system deblurs the image and extracts and registers new landmarks (Fig. 6-(f)). As a result, conventional SLAM system fails to continue mapping, and the resulting map is incomplete (Fig. 6-(g)). On the other hand, our blur-handling system reconstructs the map of entire visited region (Fig. 6-(h)).¹

We compare the number of reconstructed and tracked landmarks of the conventional SLAM system and our blur-handled system, respectively. The number of reconstructed landmarks demonstrates the contribution of our deblurring for mapping, and the number of tracked landmarks shows how our blur-robust data association improves the tracking quality. Since the camera pose is frequently lost without blur-robust data association, we use the keyframe-based relocalization [10] to recover the camera poses to continuously compare the number of landmarks. Figure 7 shows the plots of those values with respect to the frame index. Before the severe motion blur occurs at about the 250th frame, the numbers of landmarks are similar for both systems. Under 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 localization and mapping accuracy of SLAM since it is hard to obtain ground truth data. Instead, we measure the SLAM quality indirectly by measuring the reprojection errors for reconstructed landmarks. We compare the reprojection errors of

¹The video containing a whole sequence is available at <http://cv.snu.ac.kr/research/slam/slamd/>

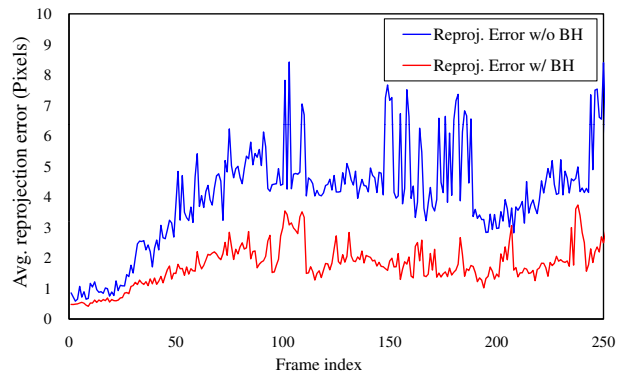


Figure 8. Comparison of the reprojection error.

Table 1. Comparison of conventional and blur-handled systems.

	Total reconstructed LM	Average # of Tracked LM	Average Reproj. Err
Conventional	664	44.6	4.02
Blur-handled	1133	114.2	1.75

the conventional system and our blur-handled system until the conventional system lost the camera pose. Figure 8 shows the results, demonstrating the superiority of our system.

We summarize the performance comparison by presenting the average values of above measured values in Table 1. The total number of reconstructed landmarks is taken 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 the first 250th frames, because after the 250th frame the conventional SLAM system frequently loses the camera pose and relies on the relocalization.

4.2. Deblurring qualities

Deblurring every input image is not necessary for improving the visual SLAM performances, but it can be helpful for some applications such as scene and object recognition during visual SLAM. Figure 9 shows the deblurring result of our algorithm for motion blur, compared with other single image deblurring methods including the uniform deblurring [18] and non-uniform deblurring [20]. The results of [18] and [20] are obtained using the software provided by the original authors. Although the results of our deblurring method suffer from some ringing artifacts due to the frequency domain operations in LR deconvolution, the overall qualities are better than the results of other methods, especially in recovered edges of objects.

Since the input scene is non-planar, the uniform deblurring method [18] gives a bad result (Fig. 9-(c)). Some regions (e.g., upper left region) are deblurred correctly, but

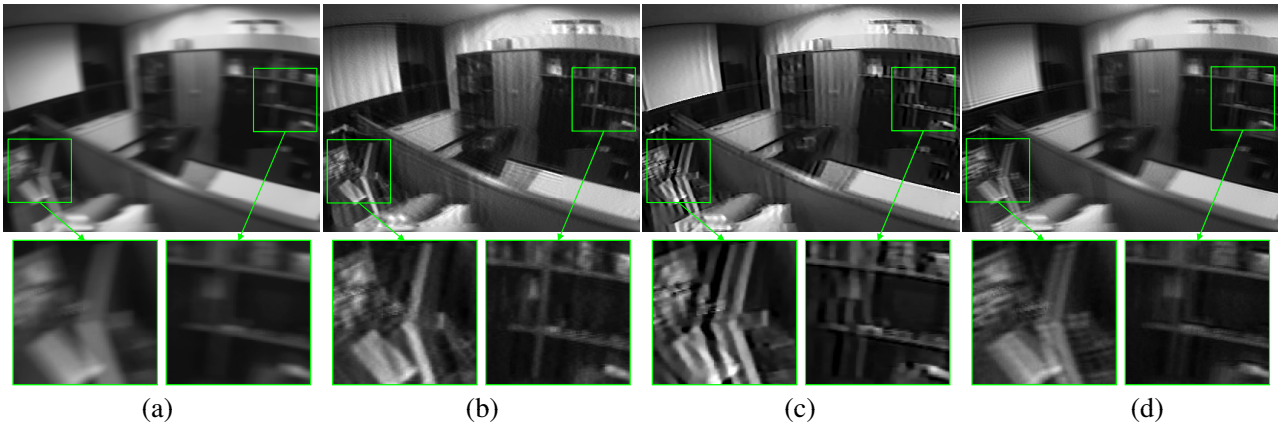


Figure 9. Comparison of image deblurring results. (a) A blurred input image. (b) A result by our proposed deblurring method. (c) A result by uniform deblurring of [18]. (d) A result by non-uniform deblurring of [20].

most regions have severe errors. [20] is non-uniform deblurring method, but it cannot deal with the scene which has more than one plane or not enough distance from the camera, like the test image in Fig. 9. Therefore, the result of [20] are also unsatisfactory (Fig. 9-(d)). Only our SLAM combined method, which considers the structure of a scene in kernel estimation, can deblur this type of scene appropriately (Fig. 9-(b)). If the real-time constraint for SLAM is not required, more advanced deconvolution algorithms can be used and much better deblurring results can be obtained.

5. Conclusion

In this paper, we proposed a new approach for handling motion blur in visual SLAM. From a camera pose and a reconstructed 3-D 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 can 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 successfully continue the SLAM process for the following frames.

Acknowledgements

This research was supported by the MKE, Korea under the ITRC Support program supervised by the NIPA. (NIPA-2011-C1090-1111-0002)

References

- [1] *edgetaper*. <http://www.mathworks.com/help/toolbox/images/ref/edgetaper.html>. 6
- [2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vision*, 56(3), 2004. 5
- [3] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway. Robust real-time visual slam using scale prediction and exemplar based feature description. In *Proc. CVPR*, 2007. 1
- [4] S. Dai and Y. Wu. Motion from blur. In *Proc. CVPR*, 2008. 2
- [5] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *Proc. SIGGRAPH*, 2006. 1, 2
- [6] J. Fung and T. Murray. Building cuda photoshop filters for the gpu, 2008. NVIDIA corporation. 6
- [7] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proc. ECCV*, 2010. 1, 2
- [8] J. Jia. Single image motion deblurring using transparency. In *Proc. CVPR*, 2007. 2
- [9] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *Proc. CVPR*, 2005. 4
- [10] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2007. 4, 6, 7
- [11] G. Klein and D. Murray. Improving the agility of keyframe-based slam. In *Proc. ECCV*, 2008. 2
- [12] J. Kwon and K. M. Lee. Monocular slam with locally planar landmarks via geometric rao-blackwellized particle filtering on lie groups. In *Proc. CVPR*, 2010. 2
- [13] L. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79(6), 1974. 6
- [14] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *Proc. IEEE International Conference on Robotics and Automation*, 2004. 5
- [15] Y. Park, V. Lepetit, and W. Woo. Esm-blur: Handling & rendering blur in 3d tracking and augmentation. In *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2009. 4, 5
- [16] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello. A visual odometry framework robust to motion blur. In *Proc. IEEE International Conference on Robotics and Automation*, 2009. 1
- [17] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. ECCV*, 2006. 4
- [18] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *Proc. SIGGRAPH*, 2008. 1, 2, 7, 8
- [19] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, 1994. 6
- [20] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proc. CVPR*, 2010. 1, 2, 7, 8
- [21] B. Williams, G. Klein, and I. Reid. Real-time slam relocation. In *Proc. ICCV*, 2007. 1