Dense 3D Reconstruction from Severely Blurred Images using a Single Moving Camera

Hee Seok Lee Kyoung Mu Lee
Department of ECE, ASRI, Seoul National University, 151-742, Seoul, Korea
ultra21@snu.ac.kr kyoungmu@snu.ac.kr
http://cv.snu.ac.kr

Abstract

Motion blur frequently occurs in dense 3D reconstruction using a single moving camera, and it degrades the quality of the 3D reconstruction. To handle motion blur caused by rapid camera shakes, we propose a blur-aware depth reconstruction method, which utilizes a pixel correspondence that is obtained by considering the effect of motion blur. Motion blur is dependent on 3D geometry, thus parameterizing blurred appearance of images with scene depth given camera motion is possible and a depth map can be accurately estimated from the blur-considered pixel correspondence. The estimated depth is then converted into pixel-wise blur kernels, and non-uniform motion blur is easily removed with low computational cost. The obtained blur kernel is depth-dependent, thus it effectively addresses scene-depth variation, which is a challenging problem in conventional non-uniform deblurring methods.

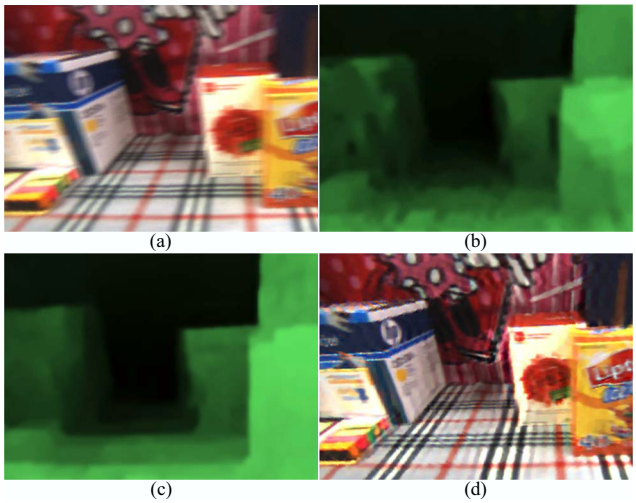


Figure 1. Depth reconstruction from five blurry images: (a) Sample from real input images. (b) Result of the conventional variational depth reconstruction. (c) Result of the proposed blur-aware depth reconstruction. (d) Deblurred image by using the estimated depth-dependent blur kernel.

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 [6, 15, 16, 22], 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 reconstruction, 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 [3, 13, 21, 24], 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, we propose a blur-handling method for 3D reconstruction, 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

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.42

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 [16]. 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 Fig.1 (b, c). The estimated depth map can be converted into pixel-wise blur kernels by using 3D geometry, and non-uniform deblurring can then be easily achieved, as shown in Fig.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.

1.1. Related Work

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. We briefly review related studies on blur kernel estimation utilizing either camera geometry or scene geometry.

Camera motion and motion blur. The relationship between the camera geometry and motion blur has been studied in multiple image deblurring [2, 13] and single image deblurring [7, 8, 25] 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*. [11] 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, [11] 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 [26] 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 [5, 17], which are closely related to our proposed method. These methods use two or more images in estimating scene depth and recovering deblurred images. However, these methods differ from our method; [5] assumes sideways translational camera motion unlike the proposed method which deals with arbitrary camera motion, a reference unblurred image is required in [17] while all input images can be blurred in our 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 [11] and handling depth variation in [26] without requiring additional inertial sensors nor a stereo camera.

2. Blur-Aware Depth Reconstruction

We convert two image blur kernel estimation problem 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.

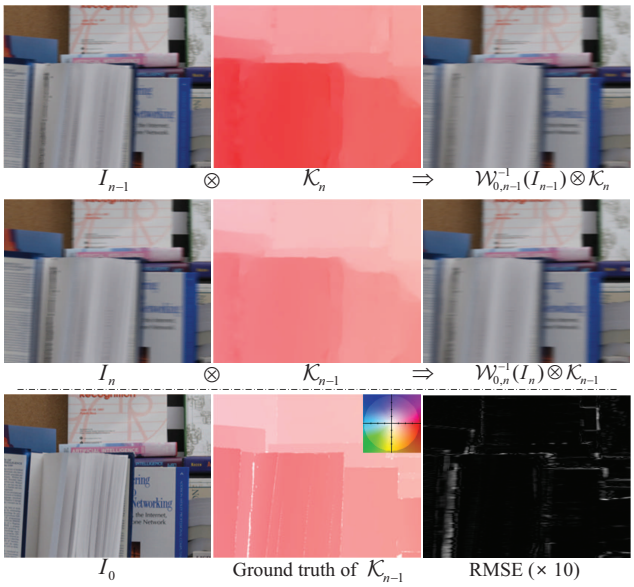


Figure 2. Commutative property of blur kernels. Top and middle: Synthesized input images I_{n-1} and I_n, the estimated blur kernels represented by motion vectors, and their commutative convolution results. Bottom: Unblurred reference image, ground truth motion vectors of I_{n-1} with a color map, and the root-mean-square (RMS) error between $\mathcal{W}_{0,n-1}^{-1}(I_{n-1})\otimes\mathcal{K}_n$ and $\mathcal{W}_{0,n}^{-1}(I_n)\otimes\mathcal{K}_{n-1}$ scaled by 10.

2.1. Motion blur estimation from two images

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 [19, 20]. 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:

$$I_n(x,y) = (L_n \otimes \mathcal{K}_n(x,y))(x,y), \tag{1}$$

where $\otimes$ denotes the convolution operator that corresponds to blur operation, and (x,y) represents a pixel coordinate. We omit the pixel coordinate notation (x,y) for images I_n and L_n as well as the blur kernel $\mathcal{K}_n$ 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:

$$L_0 \otimes \mathcal{K}_{n-1} \otimes \mathcal{K}_n = L_0 \otimes \mathcal{K}_n \otimes \mathcal{K}_{n-1}, \tag{2}$$

and it gives the approximation between two frames I_{n-1} and I_n with small changes:

$$\mathcal{W}_{0,n-1}^{-1}(I_{n-1}) \otimes \mathcal{K}_n \approx \mathcal{W}_{0,n}^{-1}(I_n) \otimes \mathcal{K}_{n-1}, \tag{3}$$

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 Fig. 2. Based on Eq. (3), we can derive the objective function to determine the correct values of $\mathcal{K}_{n-1}$ and $\mathcal{K}_n$.

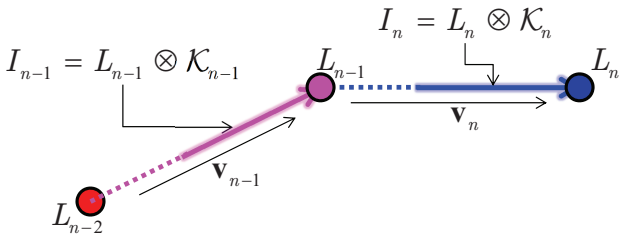


Figure 3. Proposed motion blur model: The colored dots represent the pixel positions of a 3D scene point $\mathbf{X}$ for each time n, and the intensities at these positions are represented by L. The convolution of pixel intensities along with the thick arrows corresponds to the blurred kernels $\mathcal{K}$ 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 Eq. (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 update 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 [14]:

$$\begin{aligned} L_n &= \mathcal{W}_{n-1,n}(L_{n-1}) \\ &\approx L_{n-1} + \mathcal{J}_{L_{n-1}}\mathbf{v}_n + \frac{1}{2}\mathbf{v}_n^\top \mathcal{H}_{L_{n-1}}\mathbf{v}_n, \end{aligned} \tag{4}$$

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 Fig. 3, the blurred and warped image I_n from L_{n-1} is approximated with additional coefficients as follows [12, 18]:

$$\begin{aligned} I_n &= \mathcal{W}_{n-1,n}(L_{n-1}) \otimes \mathcal{K}_n \\ &\approx L_{n-1} + a\mathcal{J}_{L_{n-1}}\mathbf{v}_n + \frac{1}{2}b\mathbf{v}_n^\top \mathcal{H}_{L_{n-1}}\mathbf{v}_n. \end{aligned} \tag{5}$$

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

$$a = \frac{\tau_o + \tau_c}{2\tau_c}, \quad b = \frac{\tau_c^2 + \tau_c\tau_o + \tau_o^2}{3\tau_c^2}, \tag{6}$$

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 Eq. (5) is equivalent to Eq. (4). Reference [18] provides the detailed derivation of this approximation.

As shown in Eq. (5), which represents the parametrization of the blurred image I by using motion vector $\mathbf{v}$, the objective function that satisfies condition (3) can be formulated by using only the motion vectors $\mathbf{v}_{n-1}$ and $\mathbf{v}_n$. First we formulate the objective function from Eq. (3),

$$\underset{\mathcal{W}_{n-1},\mathcal{W}_n,\mathcal{K}_{n-1},\mathcal{K}_n}{\arg\min} \left\| \mathcal{W}_{0,n-1}^{-1}(I_{n-1}) \otimes \mathcal{K}_n - \mathcal{W}_{0,n}^{-1}(I_n) \otimes \mathcal{K}_{n-1} \right\|_1 . \tag{7}$$

By substituting Eq. (5) into Eq. (7) for both I_{n-1} and I_n, we can obtain the objective function with respect to $\mathbf{v}_{n-1}$ and $\mathbf{v}_n$ as follows:

$$\begin{aligned} \underset{\mathbf{v}_{n-1},\mathbf{v}_n}{\arg\min} \| & (I_{n-1} + a\mathcal{J}_{I_{n-1}}\mathbf{v}_n) \\ & - (I_n + (a-1)\mathcal{J}_{I_n}\mathbf{v}_{n-1} - a\mathbf{v}_{n-1}^\top \mathcal{H}_{I_n}\mathbf{v}_{n-1}) \|_1 . \end{aligned} \tag{8}$$

The detailed derivation of Eq. (8) is presented in the Appendix. The first term in Eq. (8) 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 $\mathcal{W}_{n-1,n}^{-1}$ and the blur kernel of I_{n-1}, respectively.

2.2. 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 (8) 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 [10, 19, 20]. For example, the directions of the blur kernels of two images are assumed to be known [20]; otherwise, additional input images are used to refine the motion vectors of the two blurry images [19]. 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 [20]. 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. We utilize inverse depth, which is a reciprocal of depth, because inverse depth has better convergence property in estimation than the original depth [4].

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:

$$\begin{aligned} (x_n^\tau, y_n^\tau) &= h(\mathbf{K}((\mathbf{P}_n^\tau)^{-1} \cdot \mathbf{X})), \\ \mathbf{X} &= \frac{1}{d}\mathbf{K}^{-1} \cdot (x, y, 1)^\top, \end{aligned} \tag{9}$$

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:

$$(\mathbf{P}_n^\tau)^{-1} \cdot \mathbf{X} = (\mathbf{R}_n^\tau)^\top \mathbf{X} - (\mathbf{R}_n^\tau)^\top \mathbf{T}_n^\tau, \tag{10}$$

where $\mathbf{R}_n^\tau$ and $\mathbf{T}_n^\tau$ are camera rotation and translation, respectively. Eq. (9) shows that the blur kernel $\mathcal{K}$ in Eq. (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.

Eq. (9) 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 Eq. (9) into the original objective function (8). To solve the objective function by means of the convex optimization framework, we linearize the relationship between the pixel motions $\mathbf{v}_{n-1}$, $\mathbf{v}_n$ and a small update value of depth Δd 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:

$$\begin{aligned} \mathbf{v}_{n-1} &= \mathcal{J}_{\mathbf{v}_{n-1}} \Delta d = \mathcal{J}_{\mathbf{v}_{n-1}} (d - \bar{d}), \\ \mathbf{v}_n &= \mathcal{J}_{\mathbf{v}_n} \Delta d = \mathcal{J}_{\mathbf{v}_n} (d - \bar{d}). \end{aligned} \tag{11}$$

where $\bar{d}$ is an initial estimate of d. The objective function with respect to d is derived from Eq. (8) and Eq. (11) as follows:

$$\begin{aligned} \underset{d}{\arg\min} \| & I_{n-1} - I_n + \{a\mathcal{J}_{I_{n-1}}\mathcal{J}_{\mathbf{v}_n} + (1-a)\mathcal{J}_{I_n}\mathcal{J}_{\mathbf{v}_{n-1}}\} \\ & \cdot (d - \bar{d}) + \{a\mathcal{J}_{\mathbf{v}_{n-1}}^\top \mathcal{H}_{I_n}\mathcal{J}_{\mathbf{v}_{n-1}}\}(d - \bar{d})^2 \|_1 . \end{aligned} \tag{12}$$

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

2.3. 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 [16, 22]. 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 Eq. (12) is valid only with consecutive image indices $n-1$ and n, we should modify Eq. (12) to define the differences between the first image I_1 and other images I_n for $n \neq 2$. To this end, we warp the first image I_1 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 Eq. (9). We can then replace I_{n-1} in Eq. (12) 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:

$$\arg\min_{d} \sum_{n=2}^{N} \| I'_{n-1} - I_n + \{a\mathcal{J}_{I'_{n-1}}\mathcal{J}_{\mathbf{v}_n} + (1-a)\mathcal{J}_{I_n}\mathcal{J}_{\mathbf{v}_1}\} \cdot (d-\bar{d}) + \{a\mathcal{J}_{\mathbf{v}_1}^{\top}\mathcal{H}_{I_n}\mathcal{J}_{\mathbf{v}_1}\}(d-\bar{d})^2 \|_1. \tag{13}$$

Considering that the image warping $\mathcal{W}_{1,n-1}$ using Eq. (9) requires estimated depth, we first estimate the initial depth by using two consecutive images I_1 and I_2 with $N=2$. We then gradually increase N to improve the depth accuracy. This procedure is combined with the coarse-to-fine approach described in the next section.

3. Variational Optimization for Depth Reconstruction

To solve Eq. (13) for all image pixels, we define the energy function comprising the data and regularization terms with a scale parameter λ, such that $E = E_{reg} + \lambda E_{data}$. From Eq. (13), 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:

$$\rho(d,w) = \frac{1}{N-1}\sum_{n=2}^{N} \| I'_{n-1} - I_n + \{a\mathcal{J}_{I'_{n-1}}\mathcal{J}_{\mathbf{v}_n} + (1-a) \cdot \mathcal{J}_{I_n}\mathcal{J}_{\mathbf{v}_1}\}(d-\bar{d}) + \{a\mathcal{J}_{\mathbf{v}_1}^{\top}\mathcal{H}_{I_n}\mathcal{J}_{\mathbf{v}_1}\}(d-\bar{d})^2 + \beta w \|_1, \tag{14}$$

where w and β are the temporal illumination change term and its coefficient, respectively, which are widely used in classical optical flow formulations. For pixel noise and textureless regions we use the Huber regularization [9] given by

$$E_{reg}(d,w) = \sum_{\forall x,y} |\nabla d|_{\alpha_d} + |\nabla w|_{\alpha_w}, \tag{15}$$

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

$$|\nabla|_\alpha = \begin{cases} \frac{|\nabla|^2}{2\alpha}, & \text{if } |\nabla| \leq \alpha \\ |\nabla| - \frac{\alpha}{2}, & \text{if } |\nabla| > \alpha \end{cases}. \tag{16}$$

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

$$E = \sum_{\forall x,y} |\nabla d|_{\alpha_d} + |\nabla w|_{\alpha_w} + \lambda\rho(d,w). \tag{17}$$

We utilize the fixed values of parameters $\alpha_d = \alpha_w = 0.005$, $\beta = 0.002$, and vary the parameter λ depending on a scene.

The minimization of Eq. (17) is effectively achieved by using the first-order primal-dual algorithm [1], 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. (12) or (13) 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.

Algorithm 1 Warping and updating for depth reconstruction

1: Initialization: $d = \bar{d}$
2: **repeat**
3: Resize images and depth map to finer level
4: **for** $n = 2$ to N **do**
5: $I'_{n-1} \leftarrow \mathcal{W}_{1,n-1}(I_1) \otimes \mathcal{K}_n$
6: $I_n \leftarrow \mathcal{W}^{-1}_{n-1,n}(I_n) \otimes \mathcal{K}_1$
7: **end for**
8: **repeat**
9: Update depth d by solving Eq. (12)
10: **until** Hit max iteration
11: Update initial value: $\bar{d} \leftarrow d$
12: **until** Reach the finest level

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 [19], where the blur kernel was assumed to be linear.

4. 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 [23], we represent the blur kernel $\mathcal{K}_n$ at pixel (x, y) 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

$$I_n(x, y) = \sum_{i=0}^{M} L(x^{\tau_i}, y^{\tau_i}) k(x^{\tau_i}, y^{\tau_i})_n. \tag{18}$$

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 we have $k_n(x^{\tau_i}, y^{\tau_i}) = 1/(M+1)$ for all i. To calculate an intermediate pixel position (x^{τ_i}, y^{τ_i}) by using Eq. (9), we interpolate an intermediate camera pose $\mathbf{P}_n^{\tau_i}$ from 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:

$$\mathbf{P}_n^{\tau_i} = \exp(\frac{1}{\tau_c}(\tau_o \frac{\tau_c - \tau_o}{M} i)\Delta\mathbf{P}) \cdot \mathbf{P}_{n-1}^{\tau_c}, \tag{19}$$

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 our 3D reconstruction, 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 [23]. 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.

5. 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.

5.1. 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 Fig. 4. The initial value $\bar{d}$ is uniformly 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.

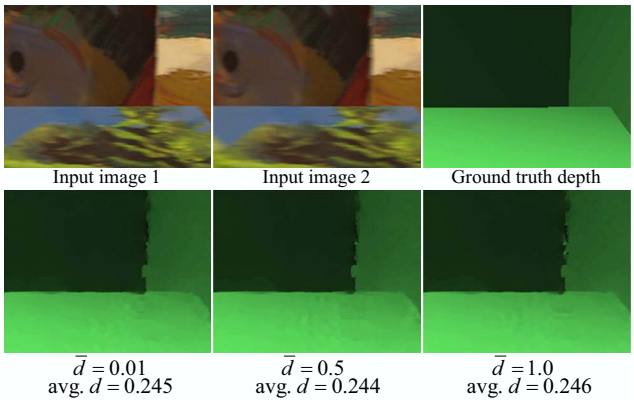


Figure 4. Depth maps for synthesized image set by using different initial depth values $\bar{d}$ at the coarsest level. The arbitrary initial values yield almost the similar depth results.

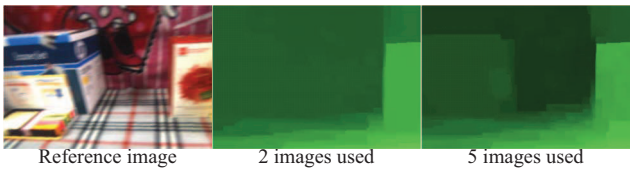


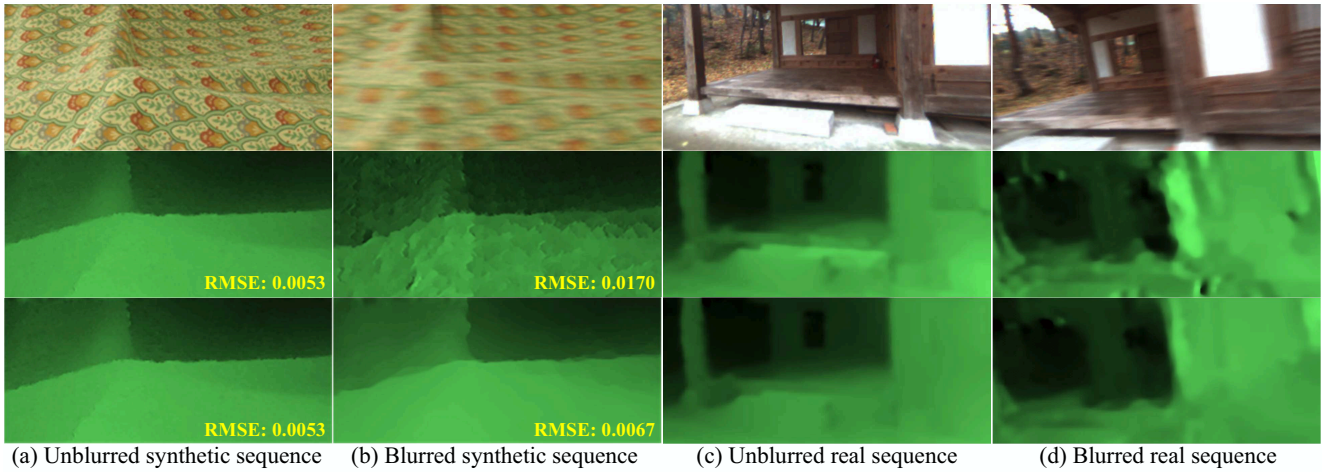
Figure 5. Improvement of depth map accuracy for real sequence by increasing the number of input blurry images.

5.2. Analysis of the number of input images

The performance gain achieved by multiple real images is shown in Fig. 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 as the number of input images increases.

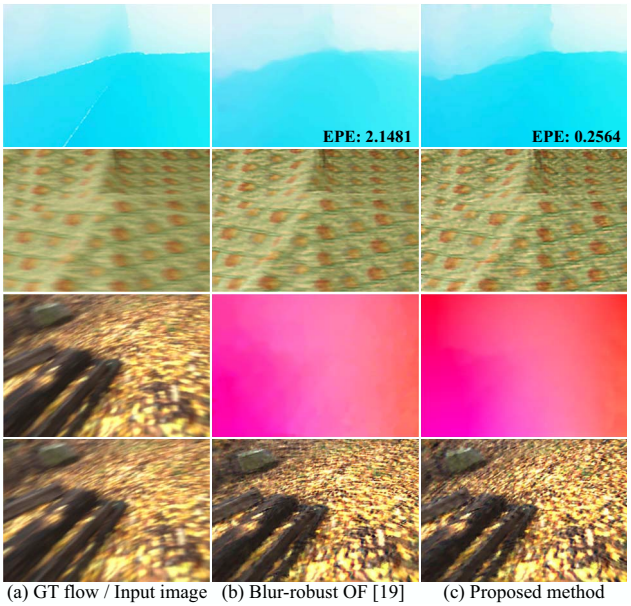
5.3. Comparison of depth reconstruction results

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, we test each method for unblurred sequence to show that each implementation works correctly as shown in Fig. 6 (a, c). We then test the methods for blurred sequence to compare their robustness to motion blur, as shown in Fig. 6



(a) Unblurred synthetic sequence (b) Blurred synthetic sequence (c) Unblurred real sequence (d) Blurred real sequence

Figure 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.



(a) GT flow / Input image (b) Blur-robust OF [19] (c) Proposed method

Figure 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 [19]. (c) Proposed method.

(b, d). The RMS error of the estimated depth are measured for the synthesized images and presented in the figure.

5.4. Comparison of optical flow results

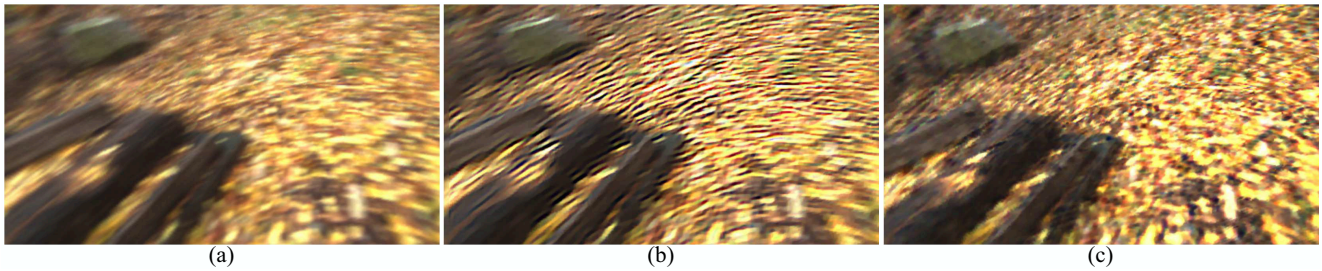
The effectiveness of the proposed blur handling is demonstrated by comparing the optical flow results, *i.e.*, vector $\mathbf{v}_n$, with those of other blur-robust method for optical flow. We convert the estimated depth into motion vectors by using Eq. (9) and then compare the motion vectors with the results of the blur-robust optical flow method in [19] to evaluate the pixel correspondence accuracy between images. As described in Section 2.2, two additional images are used in [19] as additional information, whereas camera motion is used in our method. The optical flow results are compared in Fig. 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.

5.5. Comparison of deblurring results

Finally, the deblurring results for real image data by the proposed method and the multiple image deblurring method from [24] are presented in Fig. 8. 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.

6. Conclusion

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



(a) (b) (c)

Figure 8. Deblurring results for real image: (a) Sample image from input sequence. (b) Result of [24]. (c) Result of the proposed method.

Appendix

Derivation of Eq. (8). First we apply $\mathcal{W}_{0,n}$ to both sides of Eq. (3) for simplification, which yields

$$\begin{aligned}\mathcal{W}_{0,n}(\mathcal{W}_{0,n-1}^{-1}(I_{n-1}))\otimes\mathcal{K}_n &= \mathcal{W}_{0,n}(\mathcal{W}_{0,n}^{-1}(I_n))\otimes\mathcal{K}_{n-1}\\ \Rightarrow \mathcal{W}_{n-1,n}(I_{n-1})\otimes\mathcal{K}_n &= I_n\otimes\mathcal{K}_{n-1}.\end{aligned} \tag{20}$$

From the approximation of Eq. (5) to the left-hand side of Eq. (20) up to the first-order, we have

$$\mathcal{W}_{n-1,n}(I_{n-1})\otimes\mathcal{K}_n \approx I_{n-1} + a\mathcal{J}_{I_{n-1}}\mathbf{v}_n, \tag{21}$$

Similarly, applying Eq. (5) to the right-hand side of Eq. (20) yields

$$\begin{aligned}I_n\otimes\mathcal{K}_{n-1} &= \mathcal{W}_{n-2,n-1}^{-1}(\mathcal{W}_{n-2,n-1}(I_n)\otimes\mathcal{K}_{n-1})\\ &\approx \mathcal{W}_{n-2,n-1}^{-1}(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1}),\end{aligned} \tag{22}$$

and by Eq. (4), we have

$$\begin{aligned}&\mathcal{W}_{n-2,n-1}^{-1}(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1})\\ \approx&(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1}) - \mathcal{J}_{(I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1})}\mathbf{v}_{n-1}\\ =&I_n + a\mathcal{J}_{I_n}\mathbf{v}_{n-1} - (\mathcal{J}_{I_n} + a\mathbf{v}_{n-1}^\top\mathcal{H}_{I_n}\mathbf{v}_{n-1})\\ =&I_n + (a-1)\mathcal{J}_{I_n}\mathbf{v}_{n-1} - a\mathbf{v}_{n-1}^\top\mathcal{H}_{I_n}\mathbf{v}_{n-1}.\end{aligned} \tag{23}$$

By subtracting the two terms, we have the objective function as presented in Eq. (8).

References

[1] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1), 2011. 5

[2] S. Cho, H. Cho, Y.-W. Tai, and S. Lee. Registration based non-uniform motion deblurring. *Computer Graphics Forum*, 31(7), 2012. 2

[3] S. Cho, J. Wang, and S. Lee. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Trans. Graph.*, 31(4), 2012. 1

[4] J. Civera, A. J. Davison, and J. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Trans. Robotics*, 24(5), 2008. 4

[5] P. Favaro, M. Burger, and S. Soatto. Scene and motion reconstruction from defocused and motion-blurred images via anisotropic diffusion. In *Proc. ECCV*, 2004. 2

[6] 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. 1

[7] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proc. ECCV*, 2010. 2

[8] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *Proc. ICCV*, 2011. 2

[9] P. J. Huber. Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics*, 1(5), 1973. 5

[10] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *Proc. CVPR*, 2005. 4

[11] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29(4), 2010. 2

[12] H. S. Lee, J. Kwon, and K. M. Lee. Simultaneous localization, mapping, and deblurring. In *Proc. ICCV*, 2011. 3

[13] Y. Li, S. B. Kang, N. Joshi, S. M. Seitz, and D. P. Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *Proc. CVPR*, 2010. 1, 2

[14] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *Proc. IEEE International Conference on Robotics and Automation*, 2004. 3

[15] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *Proc. CVPR*, 2010. 1

[16] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. ICCV*, 2011. 1, 2, 5

[17] C. Paramanand and A. N. Rajagopalan. Inferring image transformation and structure from motion-blurred images. In *Proc. British Machine Vision Conference*, 2010. 2

[18] 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. 3

[19] T. Portz, L. Zhang, and H. Jiang. Optical flow in the presence of spatially-varying motion blur. In *Proc. CVPR*, 2012. 3, 4, 5, 7

[20] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3), 2005. 3, 4

[21] O. Shahar, A. Faktor, and M. Irani. Space-time super-resolution from a single video. In *Proc. CVPR*, 2011. 1

[22] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Proc. DAGM conference on Pattern recognition*, 2010. 1, 5

[23] Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *IEEE Trans. PAMI*, 33, 2011. 6

[24] H. Takeda and P. Milanfar. Removing motion blur with space-time processing. *IEEE Transactions on Image Processing*, 20(10), 2011. 1, 7, 8

[25] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-unifrom deblurring for shaken images. In *Proc. CVPR*, 2010. 2

[26] L. Xu and J. Jia. Depth-aware motion deblurring. In *Proc. IEEE International Conference on Computational Photography*, 2012. 2