Accurate 3D Reconstruction from Small Motion Clip for Rolling Shutter Cameras

Sunghoon Im, *Student Member, IEEE,* Hyowon Ha, *Student Member, IEEE,*
Gyeongmin Choe, *Student Member, IEEE,* Hae-Gon Jeon, *Student Member, IEEE,*
Kyungdon Joo, *Student Member, IEEE* and In So Kweon, *Member, IEEE*

Abstract—Structure from small motion has become an important topic in 3D computer vision to estimate depth as the capturing of the input is so user-friendly. However, major limitations present themselves in the form of depth uncertainty due to a narrow baseline and the rolling shutter effect. In this paper, we present a dense 3D reconstruction method from small motion clips using commercial hand-held cameras, most of which cause the undesired rolling shutter artifact. To address these problems, we introduce a novel small motion bundle adjustment that effectively compensates for the rolling shutter effect. Moreover, we propose a pipeline for a fine-scale dense 3D reconstruction that models the rolling shutter effect by utilizing both sparse 3D points and the camera trajectory from narrow-baseline images. In this reconstruction part, the sparse 3D points are propagated to obtain the initial depth hypothesis using a geometry guidance term. Then, depth information on each pixel is obtained by sweeping the plane around each depth search space near the hypothesis. Consequently, the proposed framework shows accurate dense reconstruction results that are suitable for various sought-after applications. Both qualitative and quantitative evaluations show that our method consistently generates better depth maps compared to those by state-of-the-art methods.

Index Terms—3D reconstruction, geometry, structure from motion, rolling shutter, bundle adjustment, plane sweeping algorithm

1 Introduction

Accurate computation of scene depth forms the base for various computer vision applications such as photographic editing, recognition and augmented or virtual reality [1], [2]. The importance of depth estimation cannot be emphasized more, and recent commercial developments (e.g. mobile phones [3], [4], light field cameras [2], and RGB-D sensors [5], [6]) have proven the market interest.

In order to address these needs, a multi-camera system or additional hardware have been required. Two or more cameras are used to view a scene and depth map is determined by finding correspondences between views by means of image matching. RGB-D sensors [5], [6] project an invisible speckle pattern. Depth data is computed by correlation to a calibrated RGB camera. However, such system is often expensive or fails outside of controlled lighting conditions.

An alternative way of depth measurement is the use of a single passive sensor. Light field cameras [2], which have a micro-lens array attached in front of its CCD sensor, is developed to encode multi-view geometry in a single photographic shot. Although the multi-view images are utilized to estimate depth map, the camera system needs highly specialized hardware and suffers from a resolution trade-off. With no additional hardware to compute scene depth, depth from focus technique [7] uses a set of focal stack images taken during a focal plane sweep through the scene to find the best focus. However, the technique is very sensitive to the user's inevitable motion due to their hand shaking or heart beating. It also captures images with different focal lengths, which are hard to align.

Recently, structure from small motion (SfSM) [8], [9], [10] has renewed interest since its method of capturing input data is simple and effective. Small motion consistently happens when a user tries to hold the camera steadily before pressing the shutter. A common pipeline of the SfSM consists of the bundle adjustment and dense reconstruction. The small motion bundle adjustment simultaneously finds the camera poses and 3D locations of features without precomputation of those variables. Given camera poses or sparse 3D points, a dense depth map is computed through multi-view stereo [8], [10] or linear depth propagation [9]. The state-of-the-art SfSM work in [10] has already shown comparable 3D reconstruction results over RGB-D sensors regardless of environmental conditions and without a resolution trade-off. However, geometric artifacts due to the rolling shutter (RS) present in most digital cameras often causes severe errors in 3D reconstruction. These artifacts commonly occur when the motion is at a higher frequency than the read-out time of the camera, like when the user's hand is shaking as discussed in [11], [12].

In this paper, we propose an accurate dense 3D reconstruction method from small motion video clip taken with an off-the-shelf camera. Our key contributions are three-fold. Firstly, we present a new rolling shutter bundle adjustment that effectively removes the geometric distortions due to the RS effect under a narrow baseline condition in Sec. 3. Secondly, we design a depth propagation method exploiting its geometric evidence obtained from the bundle adjustment for reliable estimation of an initial depth map in Sec. 4.1. Lastly, we also propose a rolling shutter plane sweeping algorithm, utilizing local plane hypotheses based on the initial depth map in Sec. 4.2.

This paper extends our previous work in [9] with a rolling shutter plane sweeping algorithm that cooperates with our pre-

- *Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo and In So Kweon are with the Department of Electrical Engineering, KAIST, Korea.*
- *Project page: see https://sites.google.com/site/shimrcv/imiccv15*
- *E-mail : {shim, hwha, gmchoe, hgjeon, kdjoo}@rcv.kaist.ac.kr, and iskweon77@kaist.ac.kr*

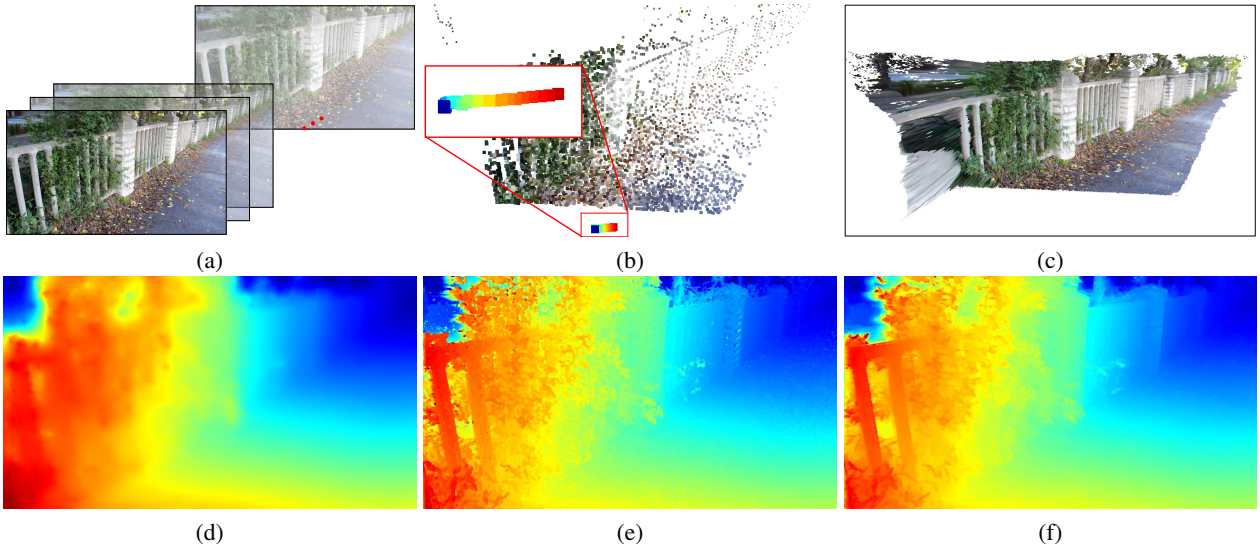


Fig. 1: Overview on the proposed method. (a) Input small motion clip. (b) Reconstructed 3D points & estimated camera trajectory in Sec. 3. (c) Final dense reconstruction using the depth map (f) in Sec. 4. (d) Initial depth map using our propagation method Sec. 4.1. (e) Depth map from winner-take-all on plane sweeping algorithm Sec. 4.2. (f) Final depth map using guided filtering.

vious pipeline and drastically improves the quality of the final results. The extended version fully utilizes the output of the bundle adjustment (*i.e.*camera poses and sparse 3D points) while the short version [9] only makes use of sparse 3D points for dense reconstruction. This provides a more accurate and edge-preserved depth map, especially for feature-less regions, as shown in Fig. 1. To verify the effectiveness of the proposed method, we conduct both qualitative and quantitative evaluations. Our method is compared to the previous method [9] and Kinect fusion [13]. Moreover, we extensively perform both qualitative and quantitative evaluations over state-of-the-art SfSM approaches and dense matching methods. Finally, we show some qualitative results in which our depth maps are applied to depth-aware image processing algorithms, such as image stylization and digital refocusing.

2 Related Work

Our algorithm is composed of two modules: the first module estimates accurate camera poses and 3D points from narrow-baseline image sequences, and the second module computes a dense 3D depth map via local plane sweeping for a RS camera. We suggest the reader refer to [14], [15] for a comprehensive review of 3D reconstruction with image sequences.

Depth from narrow baseline As it is widely known, 3D reconstruction from a narrow baseline is a very challenging task. The magnitude of the disparities are reduced to sub-pixel levels, and the depth error grows quadratically with respect to the decreasing baseline width [16]. In this context, there are other ways to estimate 3D information from the narrow-baseline instead of the conventional correspondence matching in computer vision.

A more general approach is to use video sequences as presented in [8], [17], [10], [18]. Yu and Gallup [8] utilize random depth points relative to a reference view and identical camera poses for the initialization of the bundle adjustment. The bundle adjustment produces the camera poses and sparse 3D points. Based on the output camera poses, a plane sweeping algorithm is performed to reconstruct a dense depth map. Joshi and Zitnick [17] compute per-pixel optical flow to estimate camera projection matrices of image sequences. Then, the computed projection matrices are used to align the images, and a dense disparity map is computed by rank-1 factorization. Ha *et al.* [10] present an uncalibrated SfSM that iteratively estimates camera parameters and undistorted images, and propose a plane sweeping stereo with a robust measure based on the variance of pixel intensity. Im *et al.* [18] design a SfSM framework for spherical panoramic cameras equipped with two fish-eye lenses.

While the studies in [8], [17], [10], [18] have a similar purpose to our work, which is estimating depth from narrow-baseline image sequences, we observe that the performance depends on the presence of the RS effect.

Rolling shutter Most off-the-shelf cameras are equipped with a RS to reduce manufacturing cost. However, the RS causes distortions in the image when the camera is moving. This distortion limits the performances of 3D reconstruction algorithms, such as Structure from Motion (SfM). Many works in [11], [12], [19], [20] have recently studied how to handle the RS effect. Forssen *et al.* [11] rectify the RS video through a linear interpolation scheme for camera translations and a spherical linear interpolation (SLERP) [21] for camera rotations. Hedborg *et al.* [12] formulate the RS bundle adjustment for general SfM using the SLERP schemes. While the RS bundle adjustment is effective in refining the camera poses and 3D points in a wide-baseline condition, it is inadequate for small motion due to the high order of the SLERP model. Therefore, we formulate a new RS bundle adjustment with a simple but effective interpolation scheme for small motion.

Plane sweeping algorithm Given camera poses, multi-view stereo can provide a depth for each pixel via dense matching of the images. While scenes with slanted surfaces are challenging for window-based stereo approaches [22], [23], plane sweeping algorithms have accounted for the slanted surfaces [24]. Gallup *et al.* [25] present a real-time plane sweeping algorithm to be particularly efficient on typical urban scenes. In [25], each plane-sweep is intended to reconstruct planar surfaces with a particular normal to handle the slanted surfaces.

Saurer *et al.* [26] generalize homography transfer across a

plane known for global shutter cameras to the setting of RS. Based on the rolling shutter principle, a plane sweeping algorithm suitable for rolling shutter cameras is developed. In [27], a rolling shutter bundle adjustment is proposed. Its cost function that models the rolling shutter effect incorporates GPS/INS readings, and enforces pairwise smoothness between neighboring poses. Using optimized camera poses, a dense depth map is computed by a plane sweeping algorithm used in [26].

In contrast to previous literature, especially for the rolling shutter plane sweeping algorithm, our method sweeps the plane on certain depth hypotheses while alleviating the RS artifacts. Our local plane hypotheses are derived from the sparse 3D points followed by the bundle adjustment step in Sec. 3. Although conventional plane sweeping methods have numerous plane hypotheses to produce fine-scale depth, it significantly increases the computational cost. In contrast, our work estimates accurate depth without increasing the number of hypotheses.

3 Structure from Small Motion (SfSM)

In this section, we describe the rolling shutter bundle adjustment for small motion, which provides both camera poses and 3D points. These are the key components for the next step, *dense reconstruction*. Firstly, we introduce the geometrical RS model for small motion, which carefully designed with a low-order model in Sec. 3.1. Given the model, we present the objective function for our bundle adjustment avoiding strong non-linearity in Sec. 3.2.

3.1 Geometric Model for Bundle Adjustment

Basically, we follow the conventional perspective projection model [15] which describes the relationship between a 3D point in the world and its projection onto the image plane for a perspective camera. Based on this projection model, a 3D coordinates of a world point $\mathbf{X} = [X, Y, Z, 1]^\top$ and its corresponding 2D coordinates in the image plane $\mathbf{u} = [u, v, 1]^\top$ are described as follows:

$$s\mathbf{u} = \mathbf{KPX}, \text{ where } \mathbf{K} = \begin{bmatrix} f_x & \alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where s is a scale factor and $\mathbf{P}$ is the extrinsic parameters. $\mathbf{K}$ is the intrinsic matrix of a camera that contains focal lengths f_x and f_y, principal points c_x and c_y, and skew factor α.

Contrast to the conventional Structure-from-Motion (SfM), we modify the camera pose to adopt the small angle approximation in rotation matrix representation [28]. Yu and Gallup [8] show that this adaptation reduces the order of the objective function and helps optimize whole parameters in a small motion bundle adjustment framework. Under small angular deviations, the camera extrinsic matrix for SfSM can be simplified as

$$\mathbf{P} = [R(\mathbf{r})|\ \mathbf{t}], \text{ where } R(\mathbf{r}) = \begin{bmatrix} 1 & -r^z & r^y \\ r^z & 1 & -r^x \\ -r^y & r^x & 1 \end{bmatrix}, \tag{2}$$

where $\mathbf{r} = [r^x, r^y, r^z]^\top$ is the rotation vector and $\mathbf{t} = [t^x, t^y, t^z]^\top$ is the translation vector of the camera. The function $R(\cdot)$ transforms the rotation vector $\mathbf{r}$ into the approximated rotation matrix.

This parameterization is validated well in SfSM for global shutter cameras, but significant error inevitably occurs for rolling shutter cameras. It's because a RS camera captures each row at different time instances, and each row belongs to different camera poses when the camera is moving as shown in Fig. 2. To handle this RS artifacts, several studies [21], [11], [12], [20] adapt interpolation schemes for rotation and translation components depending on its vertical position and temporal information. While translations are linearly interpolated, rotations take the SLERP (Spherical Linear intERPolation) method that covers the discontinuous change of the rotation vector caused by the periodic structure of the rotation matrix. Although this interpolation scheme helps alleviate the effect of conventional bundle adjustment [12], this high-order model can hardly be applied in small motion bundle adjustment. Thus, simple and effective interpolation method is needed for the proposed objective function.

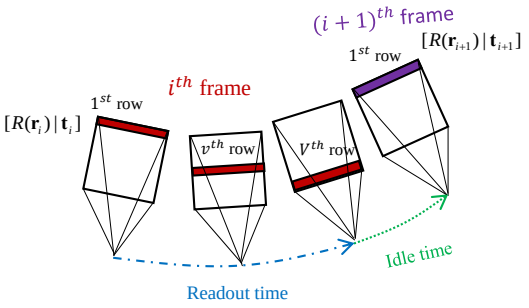


Fig. 2: The data on the red rows for the i^{th} frame are read during the readout time. The $(i+1)^{th}$ frame starts to be captured after the idle time.

To combine an interpolation in our objective function without increasing its order, we simplify the rotation interpolation by reformulating its expression under a linear form. This formulation is effective in modeling the continuously changing rotation for small motion, where the rotation matrix is composed not of periodic functions, but only of linear elements. The rotation $\mathbf{r}_{ij}$ and translation $\mathbf{t}_{ij}$ vectors for j-th feature on the i-th image are modeled by interpolating between two consecutive frames:

$$\begin{aligned} \mathbf{r}_{ij} &= \mathbf{r}_i + ah_{ij}(\mathbf{r}_{i+1} - \mathbf{r}_i), \\ \mathbf{t}_{ij} &= \mathbf{t}_i + ah_{ij}(\mathbf{t}_{i+1} - \mathbf{t}_i), \end{aligned} \tag{3}$$

where a is the ratio of the readout time of the camera for one frame and h_{ij} the row number of j-th feature divided by the total number of the rows in the i-th image. The readout time of the camera can be calculated based on the method developed by Meignast *et al.* [29]. In the global shutter camera case, a is set to zero which only considers the camera pose of present frame. With the new $\mathbf{r}_{ij}$ and $\mathbf{t}_{ij}$, the camera pose $\mathbf{P}_{ij}$ for RS projection model is formulated as:

$$\mathbf{P}_{ij} = [R(\mathbf{r}_{ij})|\ \mathbf{t}_{ij}]. \tag{4}$$

We use this camera model to build our bundle adjustment described in Sec. 3.2.

3.2 Bundle Adjustment

We now describe our cost function formulated as the sum of all reprojection errors of all the features. To estimate the optimal

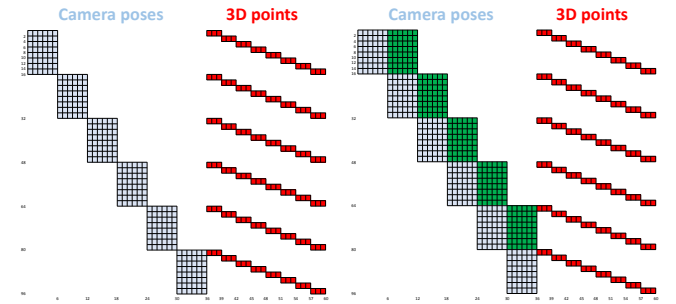


Fig. 3: Example Jacobian matrices with 8 points (24 parameters) and 6 cameras (36 parameters). (a) Jacobian matrix for global shutter BA. (b) Jacobian matrix for rolling shutter BA.

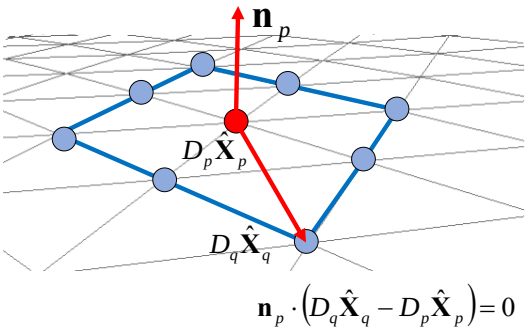


Fig. 4: Illustration of geometric guidance term: the vectors on the surface should be orthogonal to its normal vector.

camera parameters $\tilde{\mathbf{r}}$, $\tilde{\mathbf{t}}$ and the world coordinates $\tilde{\mathbf{X}}$, we solve the following optimization problem.

$$\tilde{\boldsymbol{r}}, \tilde{\boldsymbol{t}}, \tilde{\boldsymbol{X}} = \underset{\boldsymbol{r},\boldsymbol{t},\boldsymbol{X}}{\operatorname{argmin}} \sum_{i=1}^{N_I} \sum_{j=1}^{N_J} \parallel \mathbf{u}_{ij} - \pi(\mathbf{K}\mathbf{P}_{ij}\mathbf{X}_j) \parallel_2, \quad (5)$$

where $\mathbf{u}$, $\mathbf{K}$, $\mathbf{P}$, and $\mathbf{X}$ follow the previously introduced geometric model in Eq. (1) and Eq. (2). $\mathbf{r}$ and $\mathbf{t}$ follow the proposed camera model in Eq. (4). N_I and N_J are the number of images and features, and π is the projection function, that is, $\pi([x, y, z]^\top) = [x/z, y/z]^\top$.

We minimize the reprojection errors on the undistorted image domain warped by the pre-calibrated camera intrinsic matrix $\mathbf{K}$ and distortion parameters (radial and tangential). All components for the camera rotation and translation can be set to zero under small motion condition. We set the initial 3D coordinates for all pixels as the multiplication of their normalized image coordinates $\mathbf{x} = [x, y, 1]^\top$ and a random depth value.

For computational efficiency, we use the analytic Jacobian matrix which has different block structure of the conventional SfM as depicted in Fig. 3. Since our rotations and translations are influenced by two consecutive frames, each residual is related to the extrinsic parameters of two viewpoints. While the general Jacobian matrix only contains the partial derivatives for current camera (sky color), but the proposed method needs them for the next one (green color).

To obtain the feature correspondences, we initially extract features using Harris corner detection [30] and its correspondences using Kanade-Lucas-Tomasi(KLT) tracker [31]. We track all non-reference images relative to the reference frame. This scheme is feasible when the pixel changes in the subsequent frames are small. Features can be inaccurately matched since the features can suffer from slipping on lines or blurry regions, and even be shifted by moving objects or the RS effect. To filter out these outliers, we perform the forwards and backward tracking and reject outlier features with bidirectional error greater than 0.1 pixel.

4 Dense Reconstruction

Although the 3D point cloud obtained from Sec. 3 is geometrically well reconstructed, this is not dense enough to be used for various 3D scene understanding tasks or depth-aware applications [1]. In this section, we present a method to propagate the sparse points and generate dense depth map. Our method covers the rolling shutter effect of commercial cameras, where a local plane sweeping algorithm with a reliable initial depth map accurately reconstructs the dense depth map.

4.1 Depth propagation

Dense depth can be estimated by propagating sparse depths. In this paper, we call the sparse depth as Ground Control Point (GCP) [32]. We formulate the propagation problem as energy minimization for a depth $\mathbf{D}$ on every single pixel point on the basis of depth value of sparse 3D points obtained in Sec. 3. Our energy function consists of three terms: a data term $E_d(\mathbf{D})$, a color smoothness term $E_c(\mathbf{D})$ and a geometric guidance term $E_g(\mathbf{D})$ expressed as:

$$E(\mathbf{D}) = E_d(\mathbf{D}) + \lambda_c E_c(\mathbf{D}) + \lambda_g E_g(\mathbf{D}), \quad (6)$$

where λ_c and λ_g are the weights to balance the three terms. The detailed description of the three terms are as follows.

Data term In Eq. (6), the data term is defined according to the initial sparse depth map:

$$E_d(\mathbf{D}) = \sum_{p \in \mathcal{J}} \left(D_p - \tilde{D}_p\right)^2, \quad (7)$$

where $\mathcal{J}$ is a set of pixels, which has the initial depth value $\tilde{D}_p$ is computed from Sec. 3.

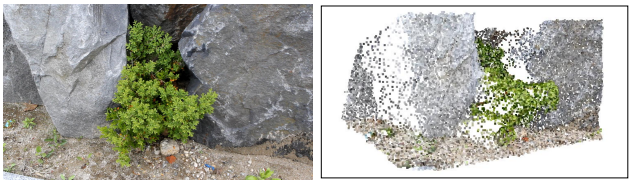
Color smoothness term The color smoothness term is defined as:

$$E_c(\mathbf{D}) = \sum_{p} \left(D_p - \sum_{q \in W_p} \frac{w^c_{pq}}{\sum_q w^c_{pq}} D_q\right)^2, \quad (8)$$

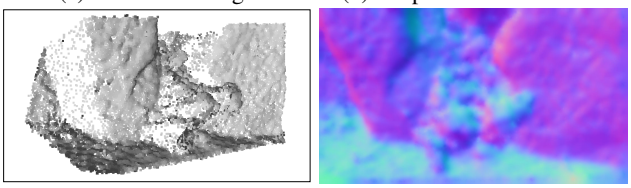
where p is a pixel on the reference image and q is the pixel in the 3×3 window W_p centered at p. The weight term w^c_{pq} is the color affinity which is defined as follows:

$$w^c_{pq} = \exp\left(\sum_{\mathbf{I} \in lab} -\frac{|\mathcal{I}_p - \mathcal{I}_q|}{2\max(\sigma_p^2, \epsilon))}\right), \quad \textit{where, } \sigma_p^2 = \sum_{q \in W_p} (\mathcal{I}_p^2 - \mathcal{I}_q^2), \quad (9)$$

where $\mathcal{I}$ is the intensity vector of the reference image in *lab* color space and ϵ is a maximum bound. This color similarity constraint which was presented in [32] is based on the assumption that each object consists of consistent color variation in the scene. Although this term outputs reliable propagation result over simple depth



(a) Reference image (b) 3D points with its color



(c) 3D points with its normal (d) Normal map

Fig. 5: Normal map estimation.

variations, it often produces geometrically incorrect depth on the slanted surface with complex color variations.

Geometric guidance term To cover the challenging real-world scenes where conventional color smoothness term easily fails, we propose a geometric guidance term that reflects the scene geometry. It provides a geometrical constraint that the normal vector of the object should be orthogonal to the vector of the object surface as shown in Fig. 4. Before incorporating the geometric guidance term in the objective function Eq. (6), we compute a pixel-wise normal map in advance as shown in Fig. 5. The detailed procedures are as follows. Firstly, we determine the normal vectors for each sparse 3D points using local plane fitting. The adjacent 3D points within a 3D sphere with the radius r_d are selected as a input point set for the plane fitting. The sparse unit normal vectors are used for the data term of the normal propagation, and each normal component in xyz is propagated by the color smoothness term in Eq. (8):

$$E_c(\mathbf{N}^x) = \sum_p \left(N_p^x - \sum_{q \in W_p} \frac{w_{pq}^c}{\sum_q w_{pq}^c} N_q^x \right)^2, \tag{10}$$

where N^x is the sparse normal x component and we do the same procedure for the other components in y and z. We obtain normal vectors for each pixel whose magnitude is close to one, but not exactly one. Thus, we normalize the vectors for each pixel to a unit vector. Since the normal vectors of adjacent pixels with high color affinity tend to be similar [33], the color-based propagation produces reliable dense normal map.

Assuming that a depth of the scene is piece-wise smooth, we define the geometric constraint $E_g(\mathbf{D})$ using the pre-computed normal map as shown in Fig. 5:

$$E_g(\mathbf{D}) = \sum_p \sum_{q \in W_p} w_p^g \left(D_p - \frac{\mathbf{n}_p \cdot \hat{\mathbf{X}}_q}{\mathbf{n}_p \cdot \hat{\mathbf{X}}_p} D_q \right)^2, \tag{11}$$

where $\mathbf{n}_p = [n_p^x, n_p^y, n_p^z]^\top$ is the normal vector of a pixel p and $\hat{\mathbf{X}}_p$ is the normalized image coordinates of p. w_p^g is a weight of the consistency of normal directions between neighboring pixels.

$$w_p^g = \frac{1}{N_g} \sum_{q \in W_p} \exp\left(\frac{-(1 - \mathbf{n}_p \cdot \mathbf{n}_q)}{\gamma_g} \right), \tag{12}$$

where γ_g is a parameter which determines the steepness of the exponential function, and N_g is the number of neighboring pixels in the window W_p. If the normal vectors of neighboring pixels are barely correlated with the normal vector of the center pixel, then the optimized depth $\mathbf{D}$ is less affected by the geometric guidance term.

Linear solution Since all three terms are formulated as quadratic equations, the depth that minimizes $E(\mathbf{d})$ can be calculated by solving

$$\nabla E(\mathbf{d}) = 0, \tag{13}$$

where $\mathbf{d}$ is the $M \times 1$ vector of the desired depth values.

For the data term, we set the term as matrix formation from Eq. (7):

$$\nabla E_d(\mathbf{d}) = (\mathbf{d} - \tilde{\mathbf{d}}), \tag{14}$$

where an one dimensional vector $\tilde{\mathbf{d}}$ consists of the initial depth value for the set of pixel $\mathcal{J}$ and zero for the others.

For the color smoothness term, we obtain a matrix form as:

$$\nabla E_c(\mathbf{d}) = (\mathbf{I} - \mathbf{W}^c)\mathbf{d}, \tag{15}$$

where $\mathbf{I}$ is a $M \times M$ identity matrix (M is the number of pixels) and $\mathbf{W}^c$ is the pairwise color similarity (w_{pq}^c) matrix.

For the geometry guidance term, we obtain a matrix form as:

$$\nabla E_g(\mathbf{d}) = \mathbf{W}^g(\mathbf{I} - \mathbf{S})\mathbf{d}, \tag{16}$$

where $\mathbf{W}^g$ is a $M \times M$ normal similarity (w_p^g) matrix and $\mathbf{S}$ is the pairwise element of $s_{pq} = (\mathbf{n}_p \cdot \hat{\mathbf{X}}_q)/(\mathbf{n}_p \cdot \hat{\mathbf{X}}_p)$ in Eq. (11).

Based on the derivatives equal to zero, the $M \times M$ Laplacian matrix $\mathbf{A}$ with weight terms λ_c and λ_g is defined as:

$$\begin{aligned} \mathbf{A}_d &= \tilde{\mathbf{I}}, \\ \mathbf{A}_c &= \mathbf{I} - \mathbf{W}^c, \\ \mathbf{A}_g &= \mathbf{W}^g(\mathbf{I} - \mathbf{S}), \\ \mathbf{A} &= \mathbf{A}_d + \lambda_c \mathbf{A}_c + \lambda_g \mathbf{A}_g, \end{aligned} \tag{17}$$

where a diagonal matrix $\tilde{\mathbf{I}}$ consists of one for the set of pixel $\mathcal{J}$ and zero for the others. Using the integrated matrix, the solution of Eq. (13) is efficiently obtained by solving a linear problem defined as:

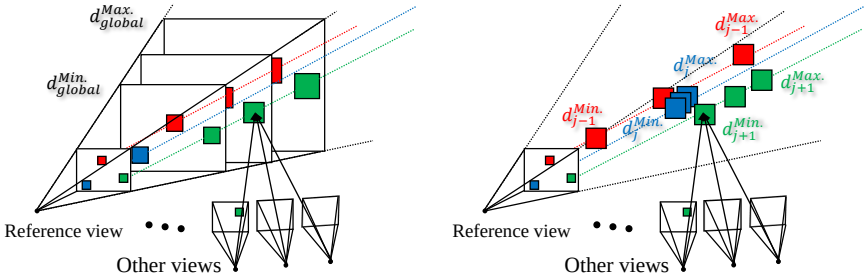
$$\mathbf{A}\mathbf{d} = \tilde{\mathbf{d}}. \tag{18}$$

Eq. (18) can be easily solved with the built-in linear solver, known as the backslash operator in MATLAB™.

4.2 Rolling Shutter Plane Sweeping

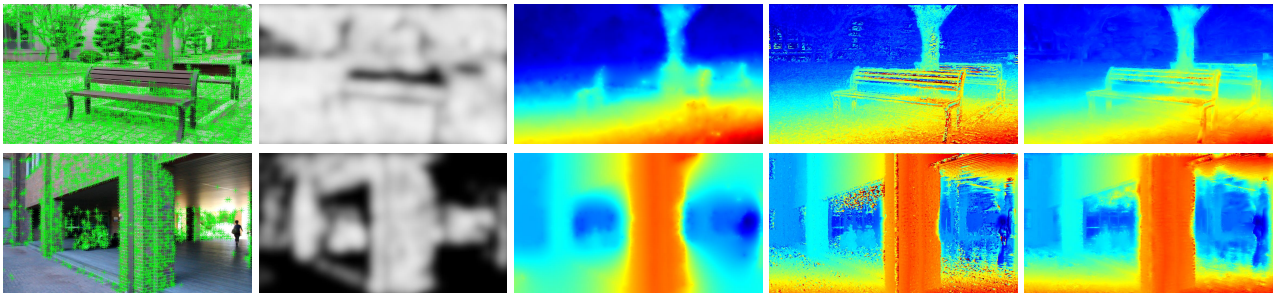
To achieve a high quality 3D geometry from sparse matching points, a depth propagation requires reliable seeds points covering over the image. In this work, the GCP acts as the reliable seed points for the depth propagation, it often fails to infer accurate depth on texture-less regions [32]. For this reason, we present a new plane sweeping algorithm that incorporates the rolling shutter camera model and sweeps a certain depth range provided by the propagated depth. The proposed method further improves the high-fidelity depth resulted from Sec. 4.1 while effectively handling the rolling shutter effect.

The plane sweeping algorithm [34], [25], [10] back-projects the image set onto a successive virtual planes (usually perpendicular to the z-axis of the reference image) in the 3D space. Then, it measures the photo consistency of all back-projected points for every virtual planes to determine the depth of each pixel. In the warping process of the conventional sweeping approach,



(a) Conventional (Full range sweeping) (b) Proposed (Local sweeping)

Fig. 6: Illustration of the global and local plane sweeping. (a) Conventional method sweeps the full depth range $[d^{Min}_{global}, d^{Max}_{global}]$. (b) Proposed method sweeps a specified depth range for each pixels $[d^{Min}_{j}, d^{Max}_{j}]$ $(j = 1, ..., N_T)$. N_T is the total number of pixels. Blue, green and red square has high, medium and low confidence respectively, so the depth range is short, medium and long.



(a) (b) (c) (d) (e)

Fig. 7: Rolling shutter plane sweeping. (a) Input reference image with extracted features. (b) GCP-based confidence map $\mathbf{C}$. (c) Initial depth map $\mathbf{D}$. (d) Depth map from winner-take-all on plane sweeping algorithm. (e) Final depth map with guided filtering.

the plane-induced homography is used on all pixels. However, the image registration using the global homography often causes geometrical errors when applied to a rolling shutter image (Note that each scanline has a different camera pose). Thus, we propose the per-pixel nature of plane sweeping algorithm that takes into account rolling shutter effect. The plane-induced homography $\mathbf{H}^m_{ij} \in \mathbb{R}^{3\times3}$ that warps the pixel j on the i-th image to the m-th virtual plane $(m = 1, ..., M)$ as follows:

$$\mathbf{H}^m_{ij} = \mathbf{K}\left(R(\mathbf{r}_{ij}) - \frac{\mathbf{t}_{ij}\mathbf{n}_z^\intercal}{d^m_j}\right)\mathbf{K}^{-1}, \tag{19}$$

where $\mathbf{r}_{ij}$ and $\mathbf{t}_{ij}$ are the rotation and translation vector of the pixel j on the i-th image. The vector $\mathbf{n}_z^\intercal = (0, 0, 1)^\intercal$ is the normal of a plane and d^m_j is m-th depth hypotheses of the pixel j. The function $R(\cdot)$ is the transformation of the rotation vector $\mathbf{r}$ into rotation matrix formation (Eq. (2)) and $\mathbf{K}$ is the intrinsic matrix (Eq. (1)).

The pixel-wise homography enables to handle the rolling shutter artifact, but full range sweeping in Fig. 6(a) produces quantized reconstruction results with a finite number of plane hypotheses, especially for the ground plane perpendicular to the sweeping direction. A greater number of plane hypotheses can alleviate the quantized artifacts, but it significantly increases the computational cost. One remarkable solution in [25] is the multiple plane sweeps where the directions are aligned to the surface normals of the scene. This is effective to reduce the computation time, but it would require many plane hypotheses when the scene includes multiple directions of surface normals. To address the issues, we incorporate a scheme to set the per-pixel depth range into the rolling shutter homography as shown in Fig. 6(b). The strategy allows the matching algorithm to sweep planes only in those regions with high prior probability according to the propagated depth from Sec. 4.1. We specify a pixel-wise depth range $[d^{\text{min}}_j, d^{\text{max}}_j]$ that is proportional to each pixel's unique confidence value C_j around the initial propagated depth as follows:

$$\begin{aligned} d^{\text{min}}_j &= (1 - \lambda_d \exp(-C_j/\gamma_d))D_j; \\ d^{\text{max}}_j &= (1 + \lambda_d \exp(-C_j/\gamma_d))D_j, \end{aligned} \tag{20}$$

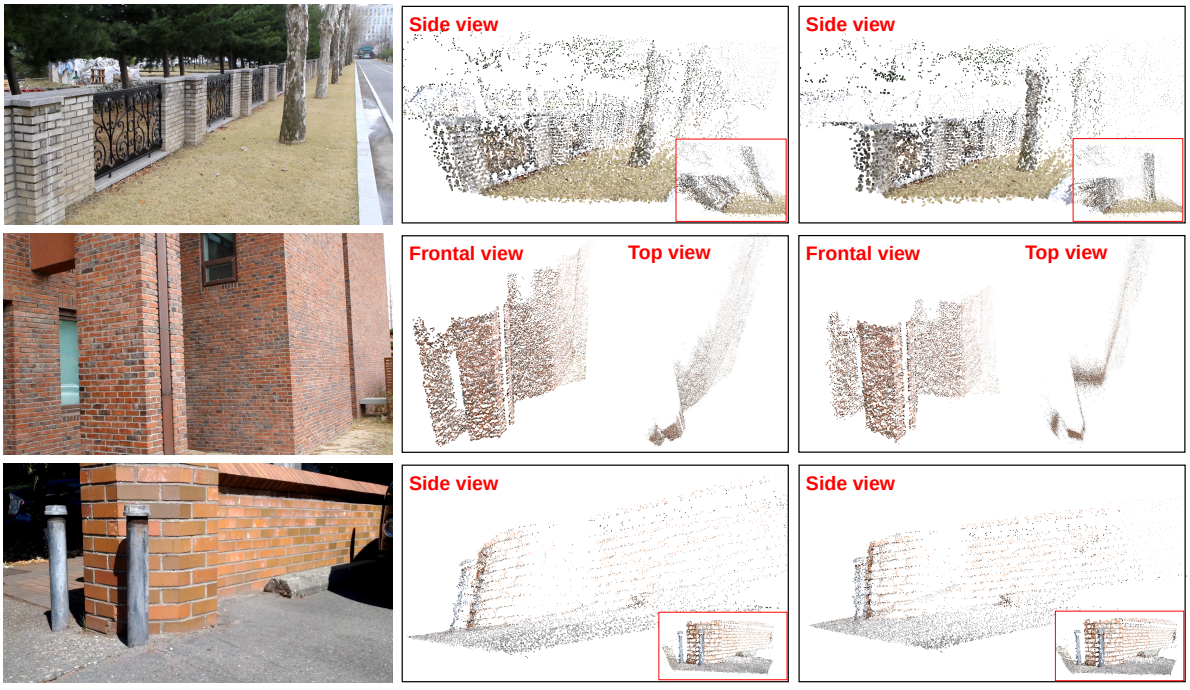
where λ_d $(0 \leq \lambda_d \leq 1)$ controls the confidence weight and γ_d controls the steepness of the exponential function. The confidence map C is defined by using a simple Gaussian filtering on the GCP binary map (One is for the GCP and zero for the rest). This is a reasonable process because the accuracy of the GCP is high, whereas the points far away from the GCP has exponentially decreasing accuracy. We uniformly quantize the depth d^m_j within a pixel-wise specified depth range $[d^{\text{min}}_j, d^{\text{max}}_j]$ with M labels. The number of labels is adjustable.

Based on the pixel-wise homography matrix, the pixel j on the images I_i whose image coordinates are represented as $\mathbf{u}_{ij}$ can be warped to the reference image (1^{st} frame) domain as

$$I^m_i(\mathbf{u}_{1j}) = I_i(\pi(\mathbf{H}^m_{ij}\mathbf{u}_{ij})), \tag{21}$$

where $\pi(\cdot)$ is the normalization function defined in Sec. 3.2. We warp all non-reference images into the reference image domain and compute a matching cost with all possible depth values. Then, we build the cost volume using the intensity variance which yields effective results for small motion case proposed in [10]:

$$C^m(\mathbf{u}_{1j}) = \text{VAR}(I^m_i(\mathbf{u}_{1j})), (i = 1, ..., N_I), \tag{22}$$



(a) Reference images (b) SfSM without RS handling (c) SfSM with RS handling

Fig. 8: SfSM result with/without RS handling - *Grass*, *Building1*, *Wall*.

TABLE 1: Mean reprojection error w.r.t the ratio of readout time a (unit: Pixel). Bold font indicates the most accurate results.

Canon EOS 60D					Microsoft Kinect2					Google Nexus				
Dataset	$a=0$	0.3	0.5	0.7	*Dataset*	$a=0$	0.1	0.3	0.5	*Dataset*	$a=0$	0.3	0.5	0.7
Faucet	0.142	0.112	**0.103**	0.115	*Plant2*	0.164	0.149	**0.122**	0.128	*Wall*	0.151	0.137	**0.119**	0.125
Flower1	0.082	0.066	**0.064**	0.086	*Flower2*	0.101	0.093	**0.078**	0.081	*Stone*	0.091	0.080	**0.079**	0.106
Brick	0.062	0.057	**0.055**	0.062	*Bridge*	0.204	0.188	**0.160**	0.172	*Hydrant*	0.115	0.105	**0.103**	0.113
Rocks	0.087	0.069	**0.064**	0.084	*Haetae*	0.104	0.097	**0.086**	0.101	*Trash*	0.152	0.131	**0.117**	0.126
Bear	0.237	0.232	**0.229**	0.260	*Table*	0.113	0.106	**0.101**	0.126	*Stair*	0.143	0.131	**0.117**	0.144

where VAR(·) is the pixel-wise variance for all images. Finally, we apply winner-takes-all (WTA) method to obtain depth map.

5 Experimental Results

We evaluate our method under two different perspectives. First of all, we demonstrate the effectiveness of each module of our framework by quantitative and qualitative evaluation in Sec. 5.1. Second, we quantitatively and qualitatively compare our depth map results with those obtained from the state-of-the-art methods [8], [9], [10] in Sec. 5.2. All those quantitative evaluations are done by comparing the depth with that from the Microsoft Kinect2 which is valid for being used as ground truth [37]. To show its generality of camera, we use the images from Canon 60D camera (DSLR) and the author-provided datasets[1] taken with a Google Nexus (mobile phone). We capture various indoor and outdoor scenes using the video capturing mode for 1 second (30 frames). With these datasets, we show that the proposed framework is applicable to various depth-aware photographic editing on mobile phone in Sec. 5.3.

With 30 frames of 960×540 resolution images, our implementation takes about 5 minutes in total. We implement all steps in MATLAB™, except the plane sweeping step coded in C++. A machine equipped with an Intel i7 3.40GHz CPU and 16GB RAM

1. http://yf.io/p/tiny/

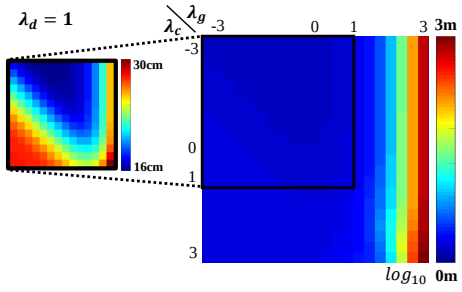
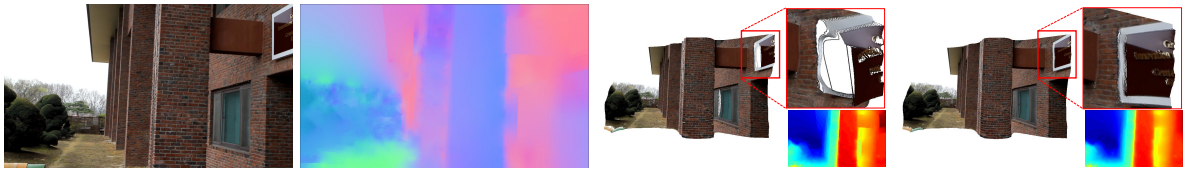


Fig. 9: RMSE map in accordance with the color weight λ_c and geometry weight λ_g. Both weights are changed from 10^{-3} to 10^3 and data weigh λ_d is fixed. The depth result is shown in Fig. 12.

was used for computation. The SfSM step (feature extraction, tracking, and bundle adjustment) requires about 8 seconds. For the dense reconstruction part with 128 labels takes 5 minutes including the depth propagation (4 seconds) and guided filtering (7 seconds). We expect that parallelized computing using GPU makes the overall process more efficient, especially for the plane sweeping part. We set an initial depth value as 300, the maximum bound ϵ as 0.001. The steepness of geometric guidance weight γ_g and the depth confidence γ_d are fixed as 0.001 and 0.005, respectively. The radius r_d is calculated as the initial depth divided



(a) Reference image (b) Estimated normal map (c) Conventional (d) Proposed

Fig. 10: Depth propagation result comparison with/without the proposed geometric guidance constraint; Real-world dataset. (a) Reference images. (b) Estimated normal map. (c) Dense 3D point clouds without guidance term. (d) Dense 3D point clouds with guidance term.

(a) Image & GT (b) Conventional (c) Proposed (a') Image & GT (b') Conventional (c') Proposed

Max. Depth: 3.178*m* RMSE: 3.348*cm* RMSE: 2.619*cm* Max. Depth: 28.24*m* RMSE: 73.26*cm* RMSE: 58.45*cm*

Fig. 11: Depth propagation result comparison with/without the proposed geometric guidance constraint; Synthetic (4D Light Field Benchmark [36]) dataset. (a) Reference images & Ground truth depth maps. (b) Dense 3D point clouds (Top) and depth difference maps (Bottom) without guidance term. (c) with guidance term.

by 20. The weighting parameters λ_c, λ_g and λ_d are set to 1. Fig. 9 shows the root-mean-square-error (RMSE) according to the color and geometry weight changes with the fixed data weight. When the data weight λ_d is 1, the optimal parameters of λ_c and λ_g are 10^{-3} and $10^{-1.5}$, respectively. We observe that the weights of color and geometry were between 10^{-3} and 10, they had little effect on the results.

5.1 Evaluation on each proposed module

RS bundle adjustment Our bundle adjustment is designed to optimize the 3D structure and camera poses while removing RS-induced artifacts. Thus, we perform both quantitative and qualitative evaluations to demonstrate the RS handling capability of the proposed method. Firstly, the gap between the quality of reconstruction results with and without RS handling is shown in Fig. 8. The results with RS-handling are obtained with Canon EOS 60D, Kinect2 RGB, and Google Nexus using known readout time ratio a of 0.5, 0.3, and 0.5, respectively. In contrast, we set $a = 0$ for the cases without RS-handling, which is a typical setting for global shutter cameras. To validate the importance of correct readout time ratio, we quantitatively compare the mean reprojection errors of the bundle adjustment results with different a values in Table 1. For each camera, five scenes are chosen and four different a values, including 0 and the correct one, are given to our bundle adjustment. As it reveals, the reprojection error decreases when the readout time ratio approaches its appropriate value and finally the error is the lowest with its optimal. We can see that our RS handling method makes the 3D geometry more realistic, while its reprojection errors are significantly reduced.

Depth propagation Fig. 10 shows the results of propagation methods with and without geometric guidance term. In Fig. 10(c), the result using only the color smoothness term contains severe artifacts on the slanted surface with multiple colors due to the lack of geometric information for an unknown depth. In contrast, the geometric guidance term assists in preserving the slanted structures as shown in Fig. 10(d). The geometric guidance term provides a more reliable 3D scene since it constrains the 3D structure along with surface normal directions.

We also quantitatively compare the conventional methods with the proposed propagation method in Fig. 11. To purely investigate the effects of geometric guidance term, we utilize the synthetic lightfield dataset [36] that provides the multiple RGB images with narrow baselines and ground truth depth map in Fig. 11(a). We extract features using Harris corner detection and remove the outliers using the bidirectional error as explained in Fig. 3.2. Then, we obtain sparse 3D points by multiplying the ground truth depth values and the normalized image coordinate of the features. Finally, we obtain the depth results of conventional and proposed method by propagating the extracted 3D points. Fig. 11(b) and (c) show the dense 3D point cloud, the depth difference map and the root-mean-square-error (RMSE). Both qualitative and quantitative evaluation shows that the proposed propagation reduces errors in texture-less regions such as the checker board Fig. 11(a) and background of Fig. 11(a').

RS plane sweeping To evaluate our dense matching method, we

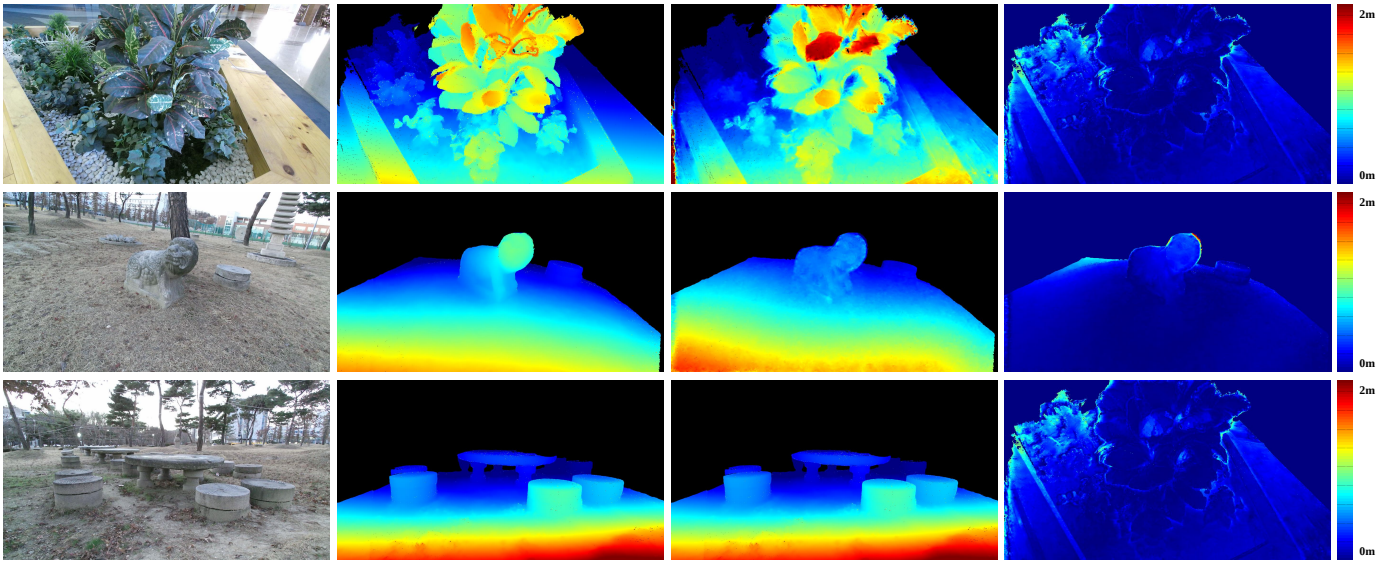


Fig. 12: Result comparison with the depth from Kinect fusion - *Plant2*, *Haetae*, *Table*. (a) Reference images from Kinect RGB sensor. (b) Depth maps from Kinect depth sensor. (c) Our depth maps. (d) Depth difference maps between (b) and (c).

TABLE 2: The percentages of depth error from Kinect2 w.r.t the ratio of readout time a.

	R10 (%)				R20 (%)				RMSE (cm)			
Dataset	$a = 0$	0.1	0.3	0.5	$a = 0$	0.1	0.3	0.5	$a = 0$	0.1	0.3	0.5
Plant2	0.678	0.758	0.842	**0.855**	0.822	0.906	**0.941**	0.934	20.67	16.08	**12.77**	13.12
Flower2	0.762	0.839	**0.861**	0.783	0.878	0.955	**0.969**	0.942	16.06	11.29	**10.29**	13.28
Bridge	0.897	0.900	**0.903**	0.718	0.969	0.975	**0.985**	0.978	20.36	19.44	**18.64**	26.79
Haetae	0.885	0.891	**0.911**	0.893	0.964	0.971	**0.985**	0.980	19.12	19.70	**17.66**	18.09
Table	0.874	0.876	**0.880**	0.855	0.971	0.974	**0.976**	0.963	18.94	18.17	**17.88**	20.01

compare our depth with Kinect depth. We capture small motion video clip using Kinect2 and generate dense mesh using Kinect fusion [13] at the same time. Then, we obtain depth map using our method with the video clip in Fig. 12(c) and Kinect depth in RGB perspective in Fig. 12(b) (The black colors in Fig. 12(b), (c) represent unmeasured depth pixels). The Kinect depth is aligned from the mesh using ray tracing with the known intrinsic and extrinsic parameters of the Kinect RGB and depth sensor.

Due to the scale ambiguity of our results, the scale of each depth map is adjusted to the scale of the depth map from the Kinect using the average measured depth value. As shown in Fig. 12, the scale-adjusted depth maps from our method are similar enough to the depth maps from the Kinect fusion. For more detailed analysis, we report root-mean-square-error (RMSE) and a robustness measure with respect to the ratio of readout time a in Table 2. A robustness measure frequently used in a Middlebury stereo evaluation system [38]. Specifically, R10 and R20 respectively denote the percentage of pixels that have a distance error of less than 10% and 20% of the maximum depth value in the scene except for the unmeasured regions. Table 2 shows that the accuracy of depth result is highest when the appropriate readout time is set and the performance gets worse when the readout time is far from its proper value.

5.2 Comparison to state-of-the-art

Qualitative evaluation Fig. 13 shows the depth maps and the corresponding point clouds obtained using our method, our previous method [9] and the state-of-the-art methods [8], [10]. We use the datasets provided in [8] and their results are brought from their website for fair comparison. The depth maps from [8], [10] seem plausible, but the corresponding 3D point clouds are clearly slanted, as seen in Fig. 13(b),(c), due to their disregard of the RS effect. Our previous method [9] achieves the desired effect, but the depth maps tend to be too smooth to distinguish the object boundary. On the other hand, our extended method produces geometrically satisfying results where the depth discontinuity is well preserved.

Quantitative evaluation We also quantitatively compare these methods, except [8] as its implementation is unreleased by the authors, by exploiting a robustness measure and RMSE as presented in Table 3. For each dataset captured with the Kinect2 sensor, 30 color images are given as input to the methods, and the results are compared to the ground truth depth map. The readout time ratio 0.3 is used for both our previous and extended methods. The results from [10] are obtained using the author-provided code[2].

The experiment shows that our method outperforms all the competing methods, and even our previous method produces more accurate results than the method [10] when compared using the ground truth depth data. This gives us two messages. Firstly, the rolling shutter handling is required to geometrically well reconstruct the 3D structure of the scene. Although the method [10] performs the self-calibration of camera parameters in addition to the reconstruction and outputs good looking results, the actual depth error is larger than those of our methods due to the lack of handling the RS effect. The experiment also shows that the

2. https://sites.google.com/site/hyowoncv/ha_cvpr16

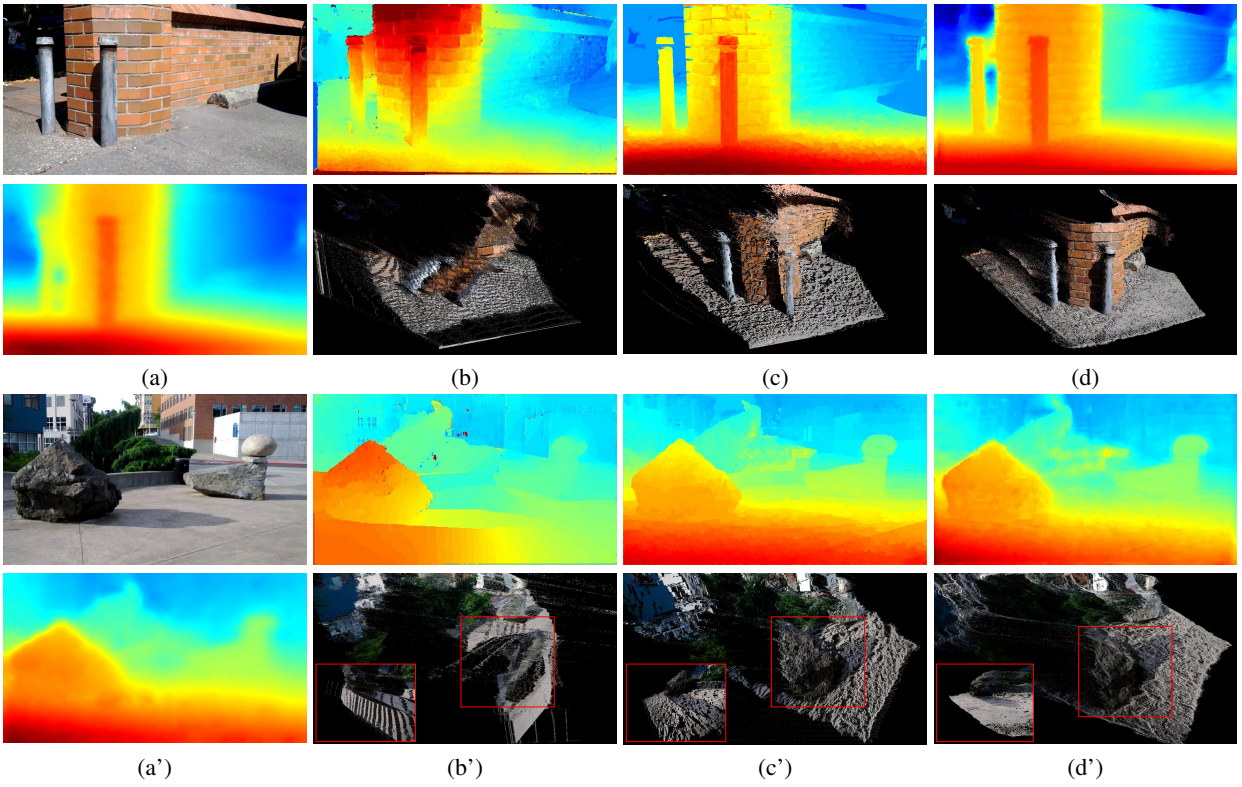


Fig. 13: Result comparison using depth map (Top) and dense 3D (Bottom) - *Wall*, *Stone'*. (a) Reference images (Top) and depth maps from our previous work [9] (Bottom). (b) Results from Yu and Gallup [8]. (c) Results from Ha *et al.* [10]. (d) Our results.

TABLE 3: Quantitative comparison with state-of-the-art. Red: Best, Blue: Second best. (Full PS*: Full depth range plane sweeping with RS handling.)

	R10 (%)				R20 (%)				RMSE (cm)			
Dataset	Im [9]	Ha [10]	Full PS*	Ours	Im [9]	Ha [10]	Full PS*	Ours	Im [9]	Ha [10]	Full PS*	Ours
Plant2	0.815	0.785	0.794	**0.842**	0.927	0.910	0.901	**0.941**	14.61	15.99	15.08	**12.77**
Flower2	0.786	0.658	0.810	**0.861**	0.943	0.910	0.930	**0.969**	11.60	19.24	13.44	**10.29**
Bridge	0.873	0.466	0.859	**0.897**	0.949	0.772	0.959	**0.985**	19.86	50.48	19.82	**18.64**
Haetae	0.893	0.737	0.895	**0.911**	0.974	0.955	0.977	**0.985**	18.80	28.58	17.90	**17.66**
Table	0.874	0.859	0.863	**0.880**	0.953	0.902	0.959	**0.976**	18.06	20.25	19.22	**17.88**

accuracy of our extended method is higher than that of our previous method. Thus, we can conclude that our method that cooperates with our previous pipeline improves the quality of the results. The results of [10] for the *Bridge* dataset provoke a large RMSE (about 50 cm). Since the method [10] sometimes fails to estimate the camera parameters, the estimated depth in this case could not be accurate enough to compare.

Lastly, we compare the Full PS (full range plane sweeping with RS handling) method to the proposed local plane sweeping method that sweeps the specified depth range. RMSE and robustness measure in Table 3 represents that the proposed method produces more accurate depth results. We observe that the proposed sweeping eliminates ambiguities found by searching over the full range of depths.

5.3 Applications

Fig. 14 shows that our method is applied to various real-world scenes (captured by Canon EOS 60D). Exploiting the depth results, we also demonstrate that our depth is a useful input for the computer vision applications. One of them in the computer vision field is a digital refocusing, which changes a level of focus after taking a photo [2], [39], [17], [8], [1], [40]. With a depth map, we can add a synthetic blur by applying different amounts of blur depending on the pixels' depth as shown in Fig. 15. For realistic digital refocusing, an accurate depth map is necessary. To visualize the improvement of the application, we synthetically render defocused images based on depth maps from the proposed method and [8]. Another interesting application is image stylization, which photographically changes an image. When depth information is given, users can easily changes objects' color on a certain depth range and produce visually pleasing image as shown in Fig. 15.

6 Discussion

Limitation & Future work We have shown that our system produces high quality dense reconstruction results and has the potential to benefit a number of computer vision applications. Nevertheless, there are several issues that can be addressed in future work. We first expect to analytically examine the feasible range of rotation angle and the baseline for SfSM. Although our method is designed for a narrow baseline setting, it does not guarantee its performance with extremely narrow baselines, for example, a thousandth of the closest scene depth. Moreover, a video clip with a small baseline, but large rotation, cannot be an adequate input to our method since large rotation breaks the

underlying assumption of the small angle approximation in our formulation. Lastly, our depth results are not in the metric scale because the estimated camera poses are relative and up to a scale factor. The metric reconstruction can be another interesting future work.

As a limitation of our work, we observe that reversed signs of depths and camera poses are sometimes resulted from the bundle adjustment. It's because our bundle adjustment only reduces its reprojection error based on its cost function regardless of geometrical correctness. We believe this undesired effect can be overcome with a proper constraint that imposes a penalty for negative depth, or with reliable pose initialization (e.g. based on inertial sensors).

Conclusion We have presented an accurate dense 3D reconstruction method from small motion clip, which effectively handles rolling shutter effect. This paper mainly introduce three contributions. Firstly, rolling shutter bundle adjustment, which jointly estimates accurate scene geometry and camera trajectory from small motion clip, is proposed. Secondly, a depth propagation method with geometric guidance constraint which provides initial depth hypothesis. Lastly, a rolling shutter plane sweeping algorithm which sweeps the plane around depth search space near the hypothesis.

Through the proposed RS handling procedure, we have shown that our method is very practical and generic, so that it can be applied to both global shutter and rolling shutter cameras. Moreover, a large variety set of experiments have been conducted, highlighting the high-quality depth maps and 3D meshes obtained with our method. These results have been compared against the existing methods with various measures, bringing to light the strong improvements offered by our approach. Finally, we also have demonstrated that our results depth are applicable for various depth-aware applications.

REFERENCES

[1] C. Hernández, "Lens blur in the new google camera app," http://googleresearch.blogspot.kr/2014/04/lens-blur-in-new-google-camera-app.html.

[2] "The lytro camera," http://www.lytro.com/.

[3] "HTC One (m8)," http://www.htc.com/us/smartphones/htc-one-m8/.

[4] "Huawei p9," http://consumer.huawei.com/en/mobile-phones/p9/index.html.

[5] "Microsoft kinect," https://developer.microsoft.com/en-us/windows/kinect.

[6] "Intel realsense," http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-shortrange.html.

[7] S. Suwajanakorn, C. Hernandez, and S. M. Seitz, "Depth from focus with your mobile phone," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[8] F. Yu and D. Gallup, "3d reconstruction from accidental motion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] S. Im, H. Ha, G. Choe, H.-G. Jeon, K. Joo, and I. S. Kweon, "High quality structure from small motion for rolling shutter cameras," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[10] H. Ha, S. Im, J. Park, H.-G. Jeon, and I. S. Kweon, "High-quality depth from uncalibrated small motion clip," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] P.-E. Forssén and E. Ringaby, "Rectifying rolling shutter video from hand-held devices," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] J. Hedborg, P.-E. Forssén, M. Felsberg, and E. Ringaby, "Rolling shutter bundle adjustment," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera," in *Proc. of the 24th Annual ACM Symposium on User Interface Software and Technology*, 2011.

[14] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[15] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[16] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[17] N. Joshi and C. L. Zitnick, "Micro-baseline stereo," *Microsoft Research Technical Report MSR-TR-2014-73*, 2014.

[18] S. Im, H. Ha, F. Rameau, H.-G. Jeon, G. Choe, and I. S. Kweon, "All-around depth from small motion with a spherical panoramic camera," in *European Conference on Computer Vision (ECCV)*, 2016.

[19] L. Magerand, A. Bartoli, O. Ait-Aider, and D. Pizarro, "Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching," in *European Conference on Computer Vision (ECCV)*, 2012.

[20] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[21] K. Shoemake, "Animating rotation with quaternion curves," in *ACM SIGGRAPH*, 1985.

[22] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 15, no. 4, pp. 353–363, 1993.

[23] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal on Computer Vision (IJCV)*, vol. 47, no. 1-3, pp. 7–42, 2002.

[24] R. T. Collins, "A space-sweep approach to true multi-image matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996.

[25] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[26] O. Saurer, K. Koser, J.-Y. Bouguet, and M. Pollefeys, "Rolling shutter stereo," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[27] O. Saurer, M. Pollefeys, and G. H. Lee, "Sparse to dense 3d reconstruction from rolling shutter images," 2016.

[28] M. L. Boas, *Mathematical Methods in the Physical*. John Wiley & Sons., Inc, 2006.

[29] M. Meingast, C. Geyer, and S. Sastry, "Geometric models of rolling-shutter cameras," *arXiv preprint cs/0503076*, 2005.

[30] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, 1988.

[31] C. Tomasi and T. Kanade, *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[32] L. Wang and R. Yang, "Global stereo matching leveraged by sparse ground control points," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[33] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical engineering*, vol. 19, no. 1, pp. 191 139–191 139, 1980.

[34] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton *et al.*, "Towards urban 3d reconstruction from video," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, 2006.

[35] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European conference on computer vision*. Springer, 2010, pp. 1–14.

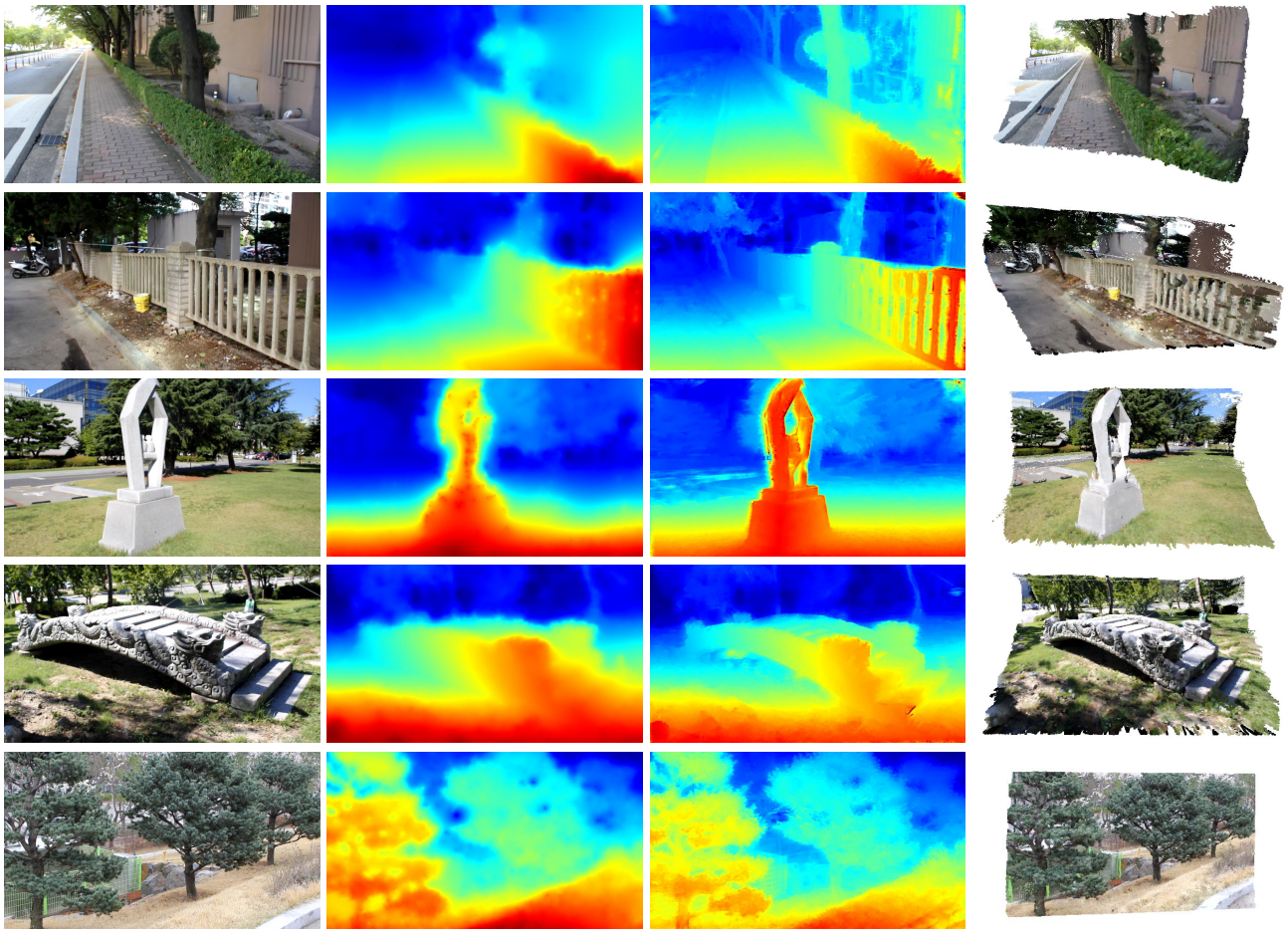
[36] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 19–34.

[37] M. Reynolds, J. Doboš, L. Peel, T. Weyrich, and G. J. Brostow, "Capturing time-of-flight data with confidence," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[38] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal on Computer Vision (IJCV)*, 2002.

[39] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar, "Picam: An ultra-thin high performance monolithic camera array," *ACM Transactions on Graphics (TOG)*, 2013.

[40] J. T. Barron, A. Adams, Y. Shih, and C. Hernández, "Fast bilateral-space stereo for synthetic defocus," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.



(a) (b) (c) (d)

Fig. 14: Our final results. (a) Reference images. (b) Depth maps from our previous work [9]. (c) Our depth maps. (d) Dense reconstruction based on (c).

(a) Input images (b) Application results (c) Input images (d) Application results

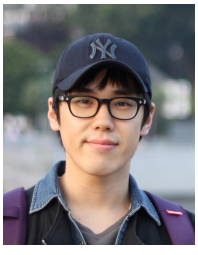
Fig. 15: Depth-aware applications using our depth results: Refocusing (Top), Stylization (Middle), Depth-aware colorization (Bottom).

Sunghoon Im received the BS degree in Electronic Engineering from Sogang University in 2014, and the MS degree in Electrical Engineering from KAIST in 2016. He is currently working toward the Ph.D. degree in Electrical Engineering at KAIST. His research interests include 3D reconstruction and computational imaging. He is a recipient of the Samsung HumanTech Paper Award and the Qualcomm Innovation Award. He is a student member of the IEEE.

Hyowon Ha received the BS and the MS degrees in electrical engineering from KAIST, Korea, in 2010 and 2012, respectively. He is currently working toward the PhD degree in Robotics and Computer Vision Lab in the School of Electrical Engineering, KAIST. His research interests include 3D reconstruction, camera calibration, and structured-light. He received the Samsung HumanTech Paper Award and the Qualcomm Innovation Award in 2016. He is a student member of the IEEE.

Gyeongmin Choe received the BS degree in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 2012. He is currently working toward the PhD degree in the field of computer vision at Department of electrical engineering, KAIST. His research interests include 3D computer vision such as various depth sensors, shape from shading and 3D geometry enhancement. He is a silver prize (2nd place) recipient of the Samsung HumanTech Paper award in 2007 and 2014. He is a student member of the IEEE.

Hae-Gon Jeon received the BS degree in Electrical and Electronic Engineering from Yonsei University in 2011, and the MS degree in Electrical Engineering from KAIST in 2013. From Aug. 2013 to Jan. 2015, he worked as a researcher at the Personal Plug and Play DigiCar Center. He is currently working toward the Ph.D. degree in Electrical Engineering at KAIST. His research interests include computational imaging and 3D reconstruction. He is a recipient of the Samsung HumanTech Paper Award and the Qualcomm Innovation Award. He is a student member of the IEEE.



Kyungdon Joo received the B.E. degree in School of Electrical and Computer Engineering from the University of Seoul in 2012, and the M.S. degree in Robotics Program from KAIST in 2014. He is currently working toward the Ph.D. degree at KAIST, South Korea. His research interests include robust computer vision, geometry and machine learning. He was a member of "Team KAIST," which won the first place in DARPA Robotics Challenge Finals 2015. He is a student member of the IEEE.

In So Kweon received the BS and MS degrees in mechanical design and production engineering from Seoul National University, Seoul, Korea, in 1981 and 1983, respectively, and the PhD degree in robotics from the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, in 1990. He worked for the Toshiba R&D Center, Japan, and joined the Department of Automation and Design Engineering, KAIST, Seoul, Korea, in 1992, where he is now a professor with the Department of Electrical Engineering. He is a recipient of the best student paper runner-up award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09). His research interests are in camera and 3D sensor fusion, color modeling and analysis, visual tracking, and visual SLAM. He was the program co-chair for the Asian Conference on Computer Vision (ACCV '07) and was the general chair for the ACCV '12. He is also on the editorial board of the International Journal of Computer Vision. He is a member of the IEEE and the KROS.