THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY | LIBRARY

HKUST SPD - INSTITUTIONAL REPOSITORY

Title	Depth estimation under motion with single pair rolling shutter stereo images
Authors	Wang, Ke; Liu, Chuhao; Wang, Kaixuan; Shen, Shaojie
Source	IEEE Robotics and Automation Letters, v. 6, (2), April 2021, article number 09369005, p. 3160-3167
Version	Accepted Version
DOI	10.1109/LRA.2021.3063695
Publisher	IEEE
Copyright	© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This version is available at HKUST SPD - Institutional Repository (https://repository.ust.hk)

If it is the author's pre-published version, changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published version.

Depth Estimation under Motion with Single Pair Rolling Shutter Stereo Images

Ke Wang, Chuhao Liu, Kaixuan Wang, and Shaojie Shen

***Abstract*—Many methods have been proposed to process stereo matching for rolling shutter image pairs, they treat all pixels from an image pair in an identical way and require additional estimation approaches to estimate motion states of the camera. However, pixels from a rolling shutter image pair naturally have diverse baseline lengths, and motion estimation methods are unstable for one instantaneous image pair input. In this paper, we present a rolling shutter stereo depth estimation pipeline, which can robustly estimate motion states and depth maps by alternating the estimation of the depth maps and refining the motion states from coarse image levels to fine image levels. What is more, we design a novel cost volume building method for rolling shutter image pairs, which adapts depth candidates to the change of baseline lengths for all pixels. We further demonstrate the usability of the proposed method by constructing a new platform, building an outdoor evaluation dataset, and comparing it with baseline methods.**

***Index Terms*—Mapping and Range Sensing**

I. Introduction

BINOCULAR stereo matching techniques have been well studied [1]–[6]. In general, stereo matching approaches are independent of vision odometry systems and depend on one instantaneous image pair. The decoupling property is useful for data fusion, and the stereo setup can continue working when the odometry system has a large drift or loses track. However, stereo matching approaches published so far usually assume cameras with global shutter (GS) sensors, which capture images so that all pixels of the same image are exposed at the same time. For the popular rolling shutter (RS) sensors [7], this exposure assumption is not applicable.

In RS cameras, exposure of rows (scanlines) happens in sequential order, leading to undesired distortion effects when the camera is not static during exposure. There are two main approaches to using RS cameras. The first is to estimate the distortion and synthesize a global shutter image that can be fed to standard vision algorithms [8]–[11], while the second is to include an RS camera model in the vision algorithms [12]–[16]. The former approach usually requires a Manhattan world assumption to search for lines or a small translation assumption to approximate pure rotation, which make it difficult to generalize these methods. Meanwhile, the latter approach keeps the original distorted images, and has led to many RS-aware algorithms for 3D vision pipelines, including RS camera calibration [12], RS structure from motion reconstruction [13], [16] and RS absolute camera pose [14], [15].

Manuscript received: October, 16, 2020; Revised January, 14, 2021; Accepted February, 24, 2021.

This paper was recommended for publication by Editor C. Cadena Lerma upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Hong Kong PhD Fellowship Scheme and HDJI Lab. *(Corresponding author: Ke Wang.)*

The authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China (e-mail:kwangbd@connect.ust.hk; cliuci@connect.ust.hk; kwangap@ust.hk; eeshaojie@ust.hk)

Our codes will be made public on https://github.com/kewangtt/SFDEFRSS.

Digital Object Identifier (DOI): see top of this page.

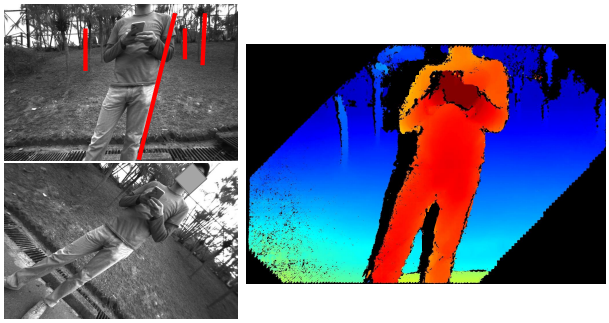


Fig. 1. The left column illustrates an image pair captured from our RS binocular, while the red lines in the first image present the rolling shutter effect produced by camera motion. The right figure shows the estimated depth map from this image pair by our method.

RS stereo depth estimation encounters two main difficulties: 1) High-quality depth estimation requires accurate motion state input to build correct cost volumes, but the deviation of motion states will lead to rapid degradation of depth estimation results. 2) Pixels from a rolling shutter image pair naturally have diverse baseline lengths, which means the general epipolar geometry is represented as a curve when the cameras are moving, and the fundamental matrix is related to the sampling position of scanlines, rather than just determined by intrinsic and extrinsic matrices. Stereo rectification of image pairs (e.g. [17]) is in general not possible as it requires all pixels of an image to have the same pose.

Recently, [18] presented a method to process RS stereo matching; it requires to be fed additional motion parameters and treats pixels with different baseline lengths from an RS image pair in the same way. At the same time, [19] proposed a minimal solver to estimate the motion state of an RS stereo setup from an RS image pair. However, the baseline length of their setup is minimal, and the estimated motion states are unstable in high-speed environments (as tested in our experiments).

In this paper, we utilize two properties to estimate motion states robustly. One is that photometric error based direct alignment frameworks [20]–[23] involve thousands of photometric constraints, which promotes the convergence of motion states. The other is that the camera motion is small during

exposure, which causes the reprojection positions based on the pre-calibrated extrinsic parameters to tend to be located in the convergence range of the photometric-based cost function. Meanwhile, because every pixel has a different baseline length when the RS stereo setup is not static, we propose a novel cost volume building method to adapt the number and values of depth candidates to the change in baseline lengths for all pixels. Our new cost volumes can avoid the overestimation of depth accuracy in a small baseline area and enhance the accuracy of the estimated depth in a large baseline area. An example depth result is shown in Fig. 1. In summary, the contributions of our paper are the following:

- We build an iterative motion estimation pipeline based on photometric error, which can estimate motion states from an instantaneous RS image pair stably.
- We introduce a novel cost volume building method, which increases the accuracy of estimated depths.
- We build a new platform and collect a new dataset for RS stereo depth estimation.

To demonstrate the effectiveness of our method, we build a new setup, which consists of two RS cameras and one LiDAR. We take point clouds measured from the LiDAR as a reference to evaluate the estimated depths. We compare the results of our method with the methods from [19], [4] and [18], which are the baseline methods for motion estimation from an RS image pair, global shutter stereo matching and rolling shutter stereo depth estimation, respectively.

II. Related works

Belief propagation (BP) [1] is a pioneering approach that formulates the stereo matching problem as a Markov network and solves it using Bayesian BP. SGM [4] uses a pixelwise, Mutual Information-based matching cost and performs a fast approximation for the global cost function by pathwise optimizations, which dramatically increases the efficiency of stereo matching. Geiger [5] proposed an efficient large-scale stereo matching method, which builds a prior on the disparities by forming a triangulation on a set of support points and reduces the matching ambiguities of the remaining points. Although all these stereo matching methods feature high accuracy, they rely on global shutter stereo rectification to ensure specific search area.

In [12], Oth proposed an algorithm to calibrate the readout time of RS cameras, while [15] focused on their epipolar geometry, defining a 7×7 generalized essential matrix for RS stereo, a formalism that explains the epipolar relationship between two RS images. Schubert [24], [25] combined DSO [20] and an RS camera to compute the pose trajectory and instantaneous motion states of the RS camera.

Saurer proposed a method [18] to build cost volumes from an RS stereo image pair. It back projects the pixels in the first image to sweep planes in the left frame and then projects these space plane points to the second image. The projection process of every pixel requires solving at least a two-order function. Although this stereo matching method can handle the rolling effect to build cost volumes, the accuracy of its result depends on the quality of the input motion states. Saurer [14] also proposed a closed-form solution to solve the motion states of two RS images and the relative pose between them.

Recently, Albl [19] proposed an algorithm to generate a global shutter image from two rolling images. It rotates the orientation of the right camera to increase the contribution of the correspondences for the convergence of motion state estimation. A minimal solver is presented to calculate the motion states of an RS image pair by giving five correspondences. However, the solver is sensitive to the feature extraction and matching noise.

In this paper, we propose a different solution to estimate dense depths from an RS stereo image pair. Our method involves an iterative photometric based motion estimation solver in the depth estimation process, and finally outputs motion states, a dense depth map and the uncertainties of the estimated depth map.

III. Preliminaries

To clarify our approach, we first introduce the RS projection model and the depth estimation model of RS stereo.

Projection model in RS cameras A critical difference between an RS camera and a GS camera is that the former does not possess a single center-of-projection in the general case. Instead, in an RS image, scanlines generally have different projection centers (temporally dynamic), local frames and orientations. Since an RS camera typically has a rapid and constant readout time (∼20 us per line), it is reasonable to assume that the camera undergoes a uniform rotation and a uniform translation during exposure.

We use $\mathbf{v} \in \mathbb{R}^3$ and $\mathbf{w} \in \mathbb{R}^3$ to denote the constant linear velocity and angular velocity per scanline, respectively. Let $\mathbf{P_0} = [\mathbf{R_0}|\mathbf{t_0}]$ represent the pose of the first scanline, the pose of the ith scanline is

$$\mathbf{P}_i = [\mathbf{R}_w(i)\mathbf{R_0} \mid \mathbf{t_0} + i\mathbf{v}], \tag{1a}$$

$$\mathbf{R}_w(i) = exp(i\mathbf{w}^\wedge), \tag{1b}$$

where $exp(\cdot^\wedge)$ transforms a three-dimensional angle-axis vector to a rotation matrix. The translation component in Eq. (1a) is a first-order approximation.

Projection model in RS Stereo As hardware synchronization is easily conducted between two cameras, we assume the ith scanlines in the left and right cameras have an identical exposure start time. We use $\mathbf{P}_i$ and $\mathbf{P}'_i$ to represent the pose of the ith scanlines in the left and right cameras, respectively. Then, we take the frame of the first scanline in the left camera as the world frame; thus $\mathbf{P}_0 = [\mathbf{I}|\mathbf{0}]$. $\mathbf{T}_l^r = [\mathbf{R}_l^r|\mathbf{t}_l^r]$ denotes the pre-calibrated relative pose between the left and right cameras. The transformations from the frame of the ith scanline in the left camera to the frame of the jth scanline in the right camera is

$$\mathbf{T}(i,j) = [\mathbf{R}_l^r\mathbf{R}_w(j)\mathbf{R}_w^T(i)|\mathbf{R}_l^r(j\mathbf{v} - i\mathbf{R}_w(j)\mathbf{R}_w^T(i)\mathbf{v}) + \mathbf{t}_l^r], \tag{2}$$

where $\mathbf{R}^T$ means the transpose of matrix $\mathbf{R}$.

RS stereo depth estimation model Similar to the GS stereo depth estimation model, the RS stereo depth estimation problem can be formulated as a Maximum Posterior Probability

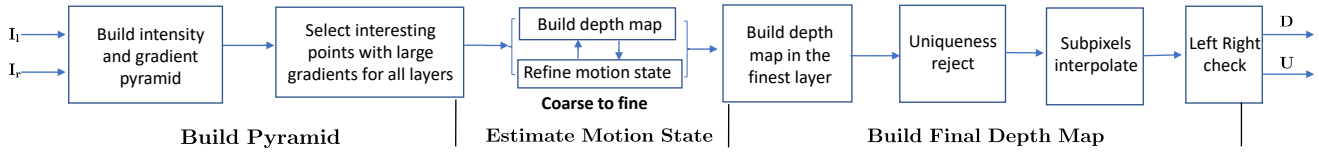


Fig. 2. Overview of our method.

(MAP) estimation problem, but additional motion states $\mathbf{v}$ and $\mathbf{w}$ are required to be estimated. The model is defined as

$$\hat{\mathbf{D}}, \hat{\mathbf{v}}, \hat{\mathbf{w}} = \arg\max_{\mathbf{D},\mathbf{v},\mathbf{w}} P(I_l, I_r|\mathbf{D}, \mathbf{v}, \mathbf{w})P(\mathbf{D}), \tag{3}$$

where variables with $\hat{(\cdot)}$ mean corresponding estimated results, I_l and I_r are an image pair captured from RS stereo cameras, and $\mathbf{D}$ is the depth map between these two images.

Following the classic assumption from [1], the observation noise follows an independent identical negative exponential distribution, the likelihood $P(I_l, I_r|\mathbf{D}, \mathbf{v}, \mathbf{w})$ is defined as

$$P(I_l, I_r|\mathbf{D}, \mathbf{v}, \mathbf{w}) \propto \prod_{\mathbf{s}} exp(-F(\mathbf{s}, d_\mathbf{s}, I_l, I_r, \mathbf{v}, \mathbf{w})), \tag{4}$$

where $F(\cdot)$ is the matching cost of pixel $\mathbf{s}$ with depth $d_\mathbf{s}$, and motion states $\mathbf{v}$ and $\mathbf{w}$. As found in [1], a depth map $\mathbf{D}$ is a Markov Random Field (MRF) [26], $P(\mathbf{D})$ is defined as

$$P(\mathbf{D}) \propto \prod_{\mathbf{s}} \prod_{\mathbf{t}\in\mathcal{N}(\mathbf{s})} exp(-\varphi(d_\mathbf{s}, d_\mathbf{t})), \tag{5}$$

where $\mathcal{N}(\mathbf{s})$ is the neighbor set of pixel $\mathbf{s}$, $d_\mathbf{s}$ and $d_\mathbf{t}$ are the depths of pixels $\mathbf{s}$ and $\mathbf{t}$, and $\varphi(\cdot)$ is a predefined smooth penalty function. Applying a negative log operation for Eq. (3), the resulting minimizing sum form is

$$\hat{\mathbf{D}}, \hat{\mathbf{v}}, \hat{\mathbf{w}} = \arg\min_{\mathbf{D},\mathbf{v},\mathbf{w}} \sum_{\mathbf{s}} \left\{ F(\mathbf{s}, d_\mathbf{s}, I_l, I_r, \mathbf{v}, \mathbf{w}) + \sum_{\mathbf{t}\in\mathcal{N}(\mathbf{s})} \varphi(d_\mathbf{s}, d_\mathbf{t}) \right\}, \tag{6}$$

where $F(\cdot)$ and $\varphi(\cdot)$ are also called the data term and smooth term.

IV. Approach

In traditional GS stereo depth estimation, Eq. (6) is solved by constructing a cost volume and running dynamic programming in it. However, for RS stereo depth estimation, the matching cost $F(\cdot)$ contains three variables, $d_\mathbf{s}$, $\mathbf{v}$, and $\mathbf{w}$. Thus, $F(\cdot)$ includes seven degrees of freedom, volumes cannot be computed directly as its super large solution space. Hence, we decouple Eq. (6) to estimate the motion states of cameras from one RS image pair and build a depth map from this pair with estimated motion states.

A. Overview of the pipeline

As shown in Fig. 2, the pipeline is composed of three stages: constructing an image pyramid, estimating motion states, and building the final depth map. Firstly, we construct an intensities and gradients pyramid from an image pair and select interesting points with large gradients for all image levels. Secondly, we estimate motion states by alternating between building the depth map and refining motion states from coarse image levels to fine image levels. Finally, we build the final depth map in the finest level of the pyramid and refine it by uniqueness rejection, sub-pixel interpolation and a left-right check. Apart from a depth map, our pipeline also generates an uncertainty map with respect to the estimated depth map.

B. Estimate motion states from an RS image pair

The correspondences from an RS image pair only make weak contributions to the convergence of motion state estimation and are prone to being disturbed by the feature extraction noise. To deal with these problems, we adopt a photometric-based approach to estimate motion states, which does not require extraction or matching of features and can involve thousands of direct photometric constraints in every iteration. The large number of constraints promotes the convergence of motion estimation. Similar to other direct-based methods [20]–[23], our approach requires proper initializations (motion states and depths of interesting points), but we avoid the initialization problem by alternating between estimating motion states and building depth maps from coarse to fine.

Motion estimation model In order to simplify the symbol system, we ignore the image level and optimization iteration index symbols, and the calculation process of every iteration in all image levels is similar. Following the definitions in Sect.III, we use $\mathbf{v}$ and $\mathbf{w}$ to denote motion states. Let $d_k \in \mathbf{D} = \{d_k|k = 1\cdots N\}$ represent the depths of the selected interesting pixels $\mathbf{p}_k = \{\mathbf{p}_k|k = 1\cdots N\}$ in any one left image level. Let $\boldsymbol{\mathcal{X}} = \{\mathbf{v}, \mathbf{w}, \mathbf{D}\}$, and we can refine $\boldsymbol{\mathcal{X}}$ by minimizing the sum of the photometric error

$$\hat{\boldsymbol{\mathcal{X}}} = \arg\min_{\boldsymbol{\mathcal{X}}} \sum_{k=1}^{N} \sum_{\mathbf{p}\in\mathcal{N}(\mathbf{p}_k)} |e(\mathbf{p}, \boldsymbol{\mathcal{X}})|_\gamma, \tag{7}$$

where $|\cdot|_\gamma$ donates the Huber norm of a value, N is the number of selected pixels in this image layer, $\mathcal{N}(\mathbf{p}_k)$ presents the eight adjacent points of $\mathbf{p}_k$ and itself. The photometric error

$$e(\mathbf{p}, \boldsymbol{\mathcal{X}}) = I_r(\pi(\mathbf{p}, \mathbf{v}, \mathbf{w}, d_k, v, v')) - I_l(\mathbf{p}) \tag{8}$$

measures the intensity difference between the pixel $\mathbf{p}$ at $\mathbf{I}_l$ and its reprojection pixel at $\mathbf{I}_r$, and v and v' are the scanline indices of $\mathbf{p}$ and its reprojection position, respectively. The reprojection function

$$\pi(\mathbf{p}, \mathbf{v}, \mathbf{w}, d_k, v, v') = \left| \mathbf{K}\mathbf{T}(v, v') \begin{bmatrix} \mathbf{K}^- \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} d_k \\ 1 \end{bmatrix} \right|_n, \tag{9}$$

where $|\cdot|_n$ means the normalization of a space point (x, y, z) and will return a two-dimensional coordinate $(x/z, y/z)$, $\mathbf{K}$ is

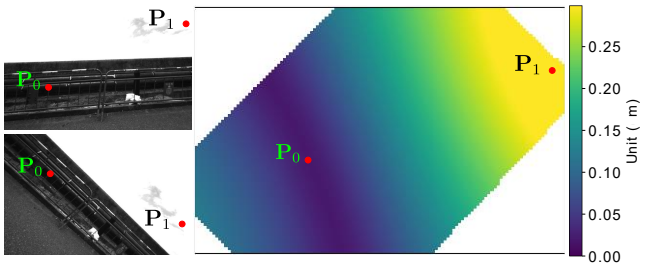


Fig. 3. Left column illustrates an RS image pair captured at a speed of 13 m/s, and the right image shows the baseline lengths of all pixels in the image pair. $\mathbf{p}_0$ and $\mathbf{p}_1$ are two individual pixels with a 3 cm baseline length and a 30 cm baseline length respectively.

the camera's intrinsic matrix, and $\mathbf{T}(v, v')$ has been defined in Eq.(2) based on motion states $\mathbf{v}$ and $\mathbf{w}$ and is a 3 by 4 matrix. Because we do not know the reprojection position before we finish this reprojection operation, v' is an unknown parameter in Eq.(9). This v' should satisfy

$$v' \approx V(\pi(\mathbf{p}, \mathbf{v}, \mathbf{w}, d_k, v, v'))_v, \tag{10}$$

where $V(\cdot)$ maps the reprojection position to the original lens-distorted coordinates. In short, the scanline of using the relative pose should be the same as the scanline of the reprojection position. In the stereo case, we pre-compute an initial pose map by giving zero motion, and there is an initial v'_k for every v_k. With this initial pose map, v'_k is refined by iterating the update $v'_k \leftarrow V(\pi(\mathbf{p}, \mathbf{v}, \mathbf{w}, d_k, v_k, v'_k))_v$ for a few times. Compared with the reprojection operation in [18], which requires solving at least a quadratic function (5-order function for uniform motion), our reprojection method has higher efficiency. The general strategy to minimize Eq. (7) is the Levenberg-Marquardt (LM) algorithm [27].

The initial motion states for Eq. (7) in the nth image level originate from the estimation results of the previous $(n+1)th$ image level (larger level is coarser). Given the latest motion states, we use our depth estimation method described in Sect.IV-C to generate the initial depth map for Eq. (7). For the coarsest image level, we take zero motion as the initial motion states.

C. Depth Estimation for RS Stereo

Given motion states $\mathbf{w}$ and $\mathbf{v}$, the data term $F(\cdot)$ described in Sect.III has one freedom, which makes building cost volumes tractable. Our depth estimation contains three steps, building cost volumes based on dynamic baselines, involving smooth terms, and refining the depth map.

Dynamic baselines All pixels have an identical baseline length in GS stereo. However, different pixels have diverse baseline lengths in RS stereo when the RS stereo setup is not static. Fig. 3 illustrates an RS image pair and its baseline map, this image pair is captured with a 13 m/s velocity. The static baseline length of the RS stereo cameras is around 15 cm. Due to the motion of the cameras, the baseline lengths in the top right areas increase to around 30 cm and reduce to about 3 cm in the bottom left areas. Given a correspondence $\mathbf{p}$ and

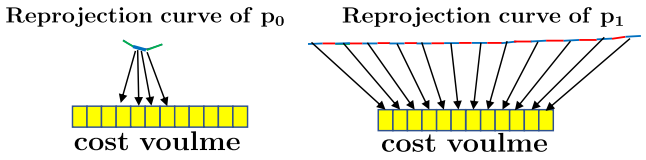


Fig. 4. This figure illustrates the constructed cost volumes of two individual pixels $\mathbf{p}_0$ and $\mathbf{p}_1$ from an RS image pair with a fixed number of depth candidates. The top two curves are the reprojection curves of $\mathbf{p}_0$ and $\mathbf{p}_1$ respectively. A colour segment in these two curves means a pixel in the right image. In the left column, many matching costs correspond to the same reprojection position. In right column, the adjacent matching costs do not correspond to the adjacent reprojection positions.

$\mathbf{p}'$ with scanline indices $v_\mathbf{p}$ and $v'_\mathbf{p}$, the instantaneous baseline length of this correspondence is computed by

$$[\mathbf{R}_\mathbf{p}|\mathbf{t}_\mathbf{p}] = \mathbf{T}(v_\mathbf{p}, v'_\mathbf{p}) \tag{11a}$$

$$b_\mathbf{p} = h(\mathbf{v}, \mathbf{w}, v_\mathbf{p}, v'_\mathbf{p}) = \left|\mathbf{R}_\mathbf{p}^{-1}\mathbf{t}_\mathbf{p}\right|_2, \tag{11b}$$

where $|\cdot|_2$ means the L2 norm and $\mathbf{T}(\cdot)$ is defined in Eq.(2).

Depth candidates For traditional GS depth estimation methods, all pixels have the same depth candidates, which are calculated by

$$d_m = f\beta x_m^{-1} \qquad x_m \in \{1, 2, \cdots, M\}, \tag{12}$$

where f is the focal length of the stereo setup, β represents the baseline length of the stereo system, d_m denotes the computed depth candidate, and M is the number of depth candidates. This definition of depth candidates is reasonable for GS stereo systems, where every depth candidate corresponds to a unique pixel (a unique disparity) in the right image and adjacent depths correspond to adjacent pixels.

However, the definition in Eq. (12) is ineffective in RS stereo depth estimation. Fig. 4 illustrates that the reprojection positions of $\mathbf{p}_0$ with depth candidates defined in Eq. (12) are located on a very short curve in the right image, and the number of pixels that this curve passes through is far smaller than M. Conversely, the reprojection positions of $\mathbf{p}_1$ are spread over a long curve. A colour segment in these two curves means a pixel in the right image; multiple matching costs correspond to the same reprojection position in the cost volume of $\mathbf{p}_0$, and the adjacent matching costs do not correspond to the adjacent reprojection position for $\mathbf{p}_1$. The former shrinking case will overestimate the accuracy of the depth at $\mathbf{p}_0$, and the latter dilating case will lose much information and underestimate the accuracy of depth at $\mathbf{p}_1$.

Building cost volumes based on dynamic baselines To deal with the dynamic baseline problem in building cost volumes, we assign different depth candidates for pixels by

$$\mathcal{M}_\mathbf{p} = \dot{b_\mathbf{p}}\beta^{-1}M, \tag{13a}$$

$$\bar{d}_m = f_u\beta\mathcal{M}_\mathbf{p}(x_m M)^{-1} \quad x_m \in \{1, 2, \cdots, \mathcal{M}_\mathbf{p}\}, \tag{13b}$$

where $\mathcal{M}_\mathbf{p}$ denotes the new number of depth candidates for the pixel $\mathbf{p}$, $\bar{d}_m$ represents the mth new depth candidate, and $\dot{b}_\mathbf{p}$ is the approximation value for the real baseline of pixel $\mathbf{p}$, and is calculated by

$$v'_\mathbf{p} \approx V(\pi(\mathbf{p}, \mathbf{v}, \mathbf{w}, d_{\frac{M}{2}}, v_\mathbf{p}, v'_\mathbf{p}))_v, \tag{14a}$$

$$\dot{b}_\mathbf{p} = h(\mathbf{v}, \mathbf{w}, v_\mathbf{p}, v'_\mathbf{p}), \tag{14b}$$

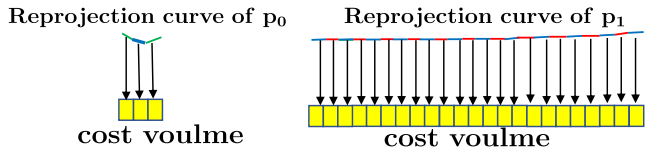


Fig. 5. This figure illustrates the constructed cost volumes of two individual pixels $\mathbf{p}_0$ and $\mathbf{p}_1$ with a dynamic number of depth candidates. A colour segment in these two curves means a pixel in right image. A matching cost corresponds to a unique reprojection position, and the adjacent matching costs correspond to the adjacent reprojection positions.

where $d_{\frac{M}{2}}$ is the median value of the original depth candidates, and functions $h(\cdot)$ and $V(\cdot)$ have been defined in Eq.(11b) and Eq.(10). With the new depth candidates, the reprojection positions of $\mathbf{p}_0$ and $\mathbf{p}_1$ are illustrated in Fig.5. The new reprojection positions are more uniform, and every depth candidate corresponds to a pixel in the right image.

The matching cost of pixel $\mathbf{p}$ for depth d is computed by

$$\begin{aligned} c_{\mathbf{p}}(d) &= F(\mathbf{p}, d, I_l, I_r, \mathbf{v}, \mathbf{w}) \\ &= \rho(I_l(\mathbf{p}), I_r(\pi(\mathbf{p}, \mathbf{v}, \mathbf{w}, d, v_p, v'_{\mathbf{p}}))), \end{aligned} \tag{15}$$

where $c_{\mathbf{p}}(d)$ denotes the matching cost and $\rho(\cdot)$ is a distance function to measure the difference between $I_l(\mathbf{p})$ and $I_r(\pi(\mathbf{p}, \mathbf{v}, \mathbf{w}, d, v_p, v'_{\mathbf{p}}))$. Fig. 6(a) shows the whole constructed cost volumes with dynamic depth steps. To utilize the high-efficiency dynamic programming algorithm in the next step, we reshape the cost volumes to the same height by a cost interpolating operation; the interpolation result is shown in Fig. 6(b). The height of the interpolated volume is the maximum depth candidate number $\mathcal{M}_{max}$ with respect to the pixel with a maximum baseline $\dot{b}_{max}$. Given a new depth d_t, the closest depths of d_t in the depth candidates of $\mathbf{p}$ are d_l and d_r, and the matching cost of $\mathbf{p}$ with respect to d_t is

$$c_{\mathbf{p}}(d_t) = \frac{(d_t^{-1} - d_r^{-1})c_{\mathbf{p}}(d_l) + (d_l^{-1} - d_t^{-1})c_{\mathbf{p}}(d_r)}{d_l^{-1} - d_r^{-1}}, \tag{16}$$

where $c_{\mathbf{p}}(d_t)$ is the interpolated matching cost.

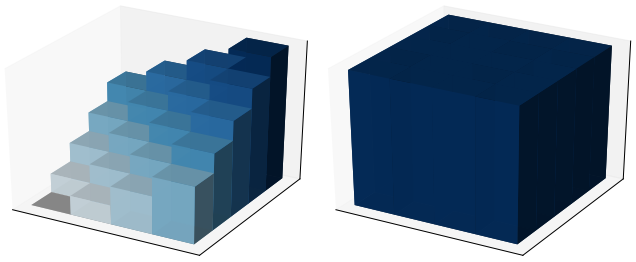
Incorporating smooth constrains The depths mentioned in the previous sections are defined in the local frame of every pixel. Due to the rapid scan time, the motion between adjacent scanlines is very small. Thus, the depth domain is smooth and also satisfies the MRF assumption. We adopt the same smooth constraints described in SGM, and replace the judging conditions from the changing value of disparities to those of our depth candidates indexes.

Uniqueness constraints Rejecting low-confidence estimated depth results is a useful step in depth estimation methods. The rejection process for pixel $\mathbf{p}$ with minimal final cost index ξ is

$$J[c_{\mathbf{p}}(d_\xi) < \tau c_{\mathbf{p}}(d_k)] \qquad k > \xi + r \ or \ k < \xi - r, \tag{17}$$

where $J[\cdot]$ is a judging function, and τ and r are two parameters for controlling the magnitude of the uniqueness constraint. The depth d_ξ will be accepted when $J[\cdot]$ returns true for all d_k. In the RS stereo case, r should be different for different baselines; the r of pixel $\mathbf{p}$ is calculated by

$$r_{\mathbf{p}} = \left\lfloor r \frac{\dot{b}_{max}}{\dot{b}_{\mathbf{p}}} \right\rfloor, \tag{18}$$



(a) (b)

Fig. 6. (a) shows the constructed volumes with dynamic depth candidates and (b) illustrates the interpolated result from (a).

where $\lfloor \cdot \rfloor$ is the floor operation.

Depth uncertainty map We find that the accuracy of estimated depths are proportional to their baselines. Thus, the baseline maps can be regarded as the uncertainties of the estimated depths.

Finally, we remove the undesired RS effects contained in the estimated depth map by reprojecting all pixels to the frame of the first scanline at the left image. Poses of all pixels have been defined in Eq. (1); thus, undistortion is easily performed based on estimated motion states and depths.

V. Experiments

We evaluate our method on a real dataset, which contains ten sequences and is collected from a four-camera setup (as shown in Fig.7). The camera setup contains two uEye UI-3881LE cameras (RS cameras) by IDS with Lensagon BK5M3920 lens by Lensation, two GS cameras, a Velodyne LiDAR and a DJI-A3 controller (IMU of 400 HZ). Hardware triggers from the IMU synchronize the four cameras. The resolutions of the two RS cameras are up to 3088×2076, and the extrinsic and intrinsic of all cameras are pre-calibrated by the Kilar-calibration tool [12], [28]. In our setup, the right RS camera is rotated 45 degrees from the y-axis to the x-axis and the baseline of the two RS cameras is around 15 cm.

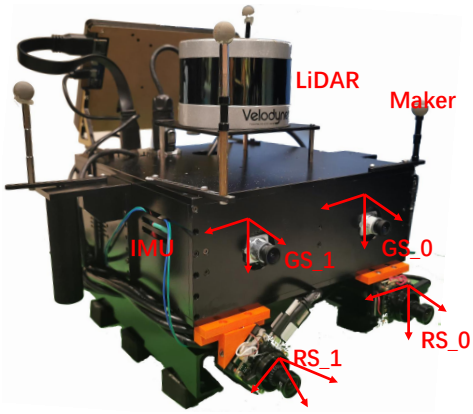


Fig. 7. Camera setup used to acquire our dataset. There are four cameras: two global-shutter cameras and two rolling shutter cameras. All cameras are hardware-synchronized with the IMU, and the transformations between all frames are pre-calibrated. Outdoor structure ground-truth is recorded by a LiDAR.

The total readout time for all scanlines (2048 scanlines) is approximately 0.06 s. There is an approximately 0.6 m displacement during the readout process when the speed of the cameras is 10 m/s (36 km/h). The line exposure time used to

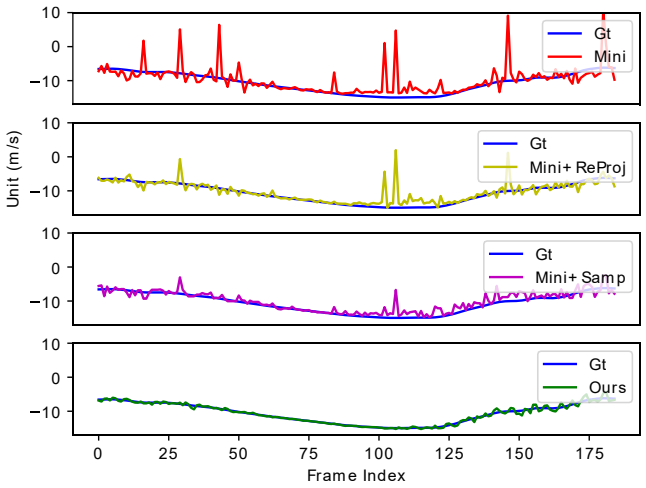


Fig. 8. The estimated velocities from four methods for a sequence: the minimal solver proposed in [19] (Mini), the combination of the minimal solver and refinement based on reprojection error (Mini+ReProj), the minimal solver with a refinement based on Sampson error (Mini+Samp) and our method. The blue curve (Gt) means the ground truth of the velocities.

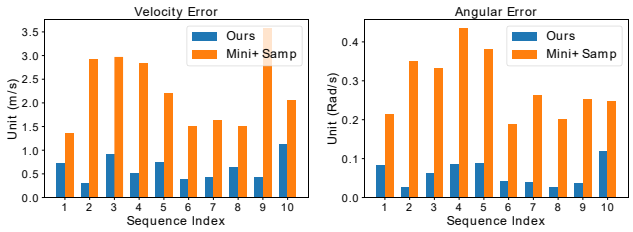
collect data is about 0.5 ms. Our evaluation contains two parts; one for motion states estimation and the other for depth map estimation. We utilize A-LOAM[1], which is a public LiDAR odometry project, to build the ground truth of the motion states and depth maps.

A. Evaluation for Estimated Motion States

We compare our method with the baseline motion estimation method from [19], for which the code is not publicly available. We build their algorithm ourselves. We utilize the automatic generator of Grobner solvers provided by Kukelova [29] to generate a closed-form motion states solver that gives up to 20 real solutions with respect to a 5-correspondence input. We sample 200 5-correspondence subgroups for every image pair and select the ten solutions with the most inliers as candidate solutions of the closed-form solver. Then,we build two motion state refinements: one the same as in [19], namely, refining motion states by minimizing Sampson error, and the other by minimizing reprojection errors. Therefore, we have three sets of solutions from [19]. In order to avoid redundant references, we use **Mini**, **Mini+ReProj** and **Mini+Samp** to indicate the solutions from the closed-form solver, the solutions with reprojection error refinement, and the solutions with Sampson error based refinement, respectively.

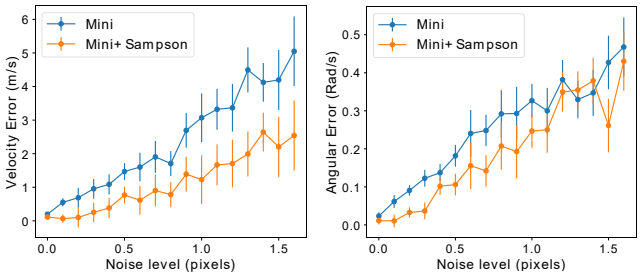
There are two error metrics for the estimated motion states, the L2 norm of the velocity residuals and the L2 norm of the rotation residuals. For the candidate solutions from the above three variants, we take the best one as the final evaluation result. Fig. 9 illustrates the average performance of our method and Mini+Samp in 10 different sequences. The average accuracy of motion states from our method is consistently better than that from Mini+Samp. Fig. 8 shows the velocity details of the estimated results from the first sequence. The blue curve in Fig. 8 means the ground truth of velocities. It can be seen that the results from our method (green) are smoother than those from the other three methods.

[1]https://github.com/HKUST-Aerial-Robotics/A-LOAM



(a) (b)

Fig. 9. Illustration of the average accuracy of estimated motion states from Mini+Samp and our method in ten sequences.



(a) (b)

Fig. 10. Accuracy of motion estimation versus feature noise levels.

To quantitatively analyze why the solver from [19] is unstable, we simulate feature correspondences with different noise levels. The simulation results are illustrated in Fig. 10. The results of motion estimation quickly worsen with increasing correspondences noise, especially when the noise level is over 0.8 pixels.

B. Evaluation for Estimated Depth Maps

We compare our method with that from [18] and SGM. As the method in [18] requires a precomputed motion state, we build two variants, one that takes motion estimation results from Mini+Samp and feeds it to the method from [18] and another that takes our motion estimation results as the input of [18]'s method. To avoid redundant references, we use **MiniSamp+RSS** and **OurMS+RSS** to indicate their estimated depths, respectively. The implementation configurations for stereo matching are identical for all methods, 8-path cost aggregation, $P1 = 10$, and $P2 = 120$.

We take the mapping results of A-LOAM as the reference values for the estimated depth maps and the mapping results are very sparse compared with the estimated depth maps. We generate the ground truths of an image pair captured at time t with three steps: 1) find the global pose of the LiDAR at time t and transform the mapping results to the local coordinate of the LiDAR at time t; 2) transform point clouds from the local LiDAR coordinate to the camera coordinate by a pre-calibrated extrinsic and remove these points out of the camera view; and 3) remove the occluded points manually, as the sparse mapping results and inevitable sensor noise mean we cannot remove the occlusion area automatically.

We use two metrics to evaluate the depth maps, absolute depth error and the fill rate of the depth maps. The former is

defined as follows:

$$\left[u_i, v_i, 1\right]^{\mathbf{T}} = \mathbf{K}\left[\frac{p_{i_x}}{p_{i_z}}, \frac{p_{i_y}}{p_{i_z}}, 1\right]^{\mathbf{T}} \tag{19a}$$

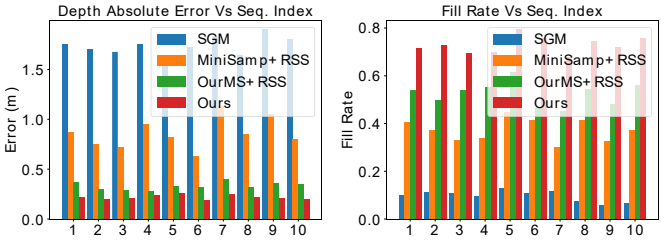
$$e_i = \left|p_{i_z} - y(\mathbf{D_j}, u_i, v_i)\right|_1, \tag{19b}$$

where $|\cdot|_1$ represents the L1 norm, $\mathbf{D_j}$ is the depth map of the jth RS image pair, and p_i is the ith point in the reference point cloud of jth image pair. The function $y(\mathbf{D_j}, u_{x_{ij}}, v_{y_{ij}})$ means the bilinear interpolation in the coordinate $(u_{x_{ij}}, v_{y_{ij}})$ of the depth map $\mathbf{D_j}$. The latter metric, the fill rate, means the ratio of valid estimated depths, and is defined as

$$th_i = max(p_{i_z} * 0.05, 0.15) \tag{20a}$$

$$n_j = \sum_{i=0}^{N} |e_i < th_i|_c \qquad r_j = \frac{n_j}{N}, \tag{20b}$$

where N is the number of points in the reference point cloud. $|e_i < th_i|_c$ means a conditional operation: if condition $e_i < th_i$ is true, the operation will return 1, and otherwise return 0. The calculated result r_j presents the fill rate of the jth image pair.

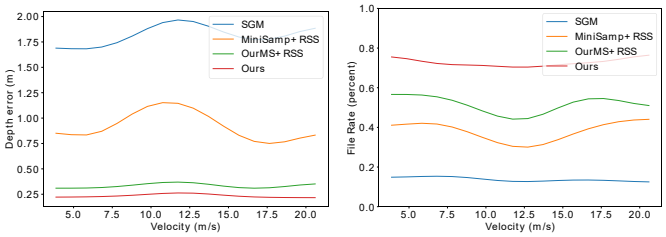


(a) (b)

Fig. 11. Illustration of the average depth errors and average fill rates for SGM, MiniSamp+RSS, OurMS+RSS and our method in ten sequences.

Fig.12 illustrates the change of depth error and fill rate along with the increase of velocities for SGM, MiniSamp+RSS, OurMS+RSS and our method. It can be seen that our method is consistently better than the other three methods. Fig.11 illustrates the depth errors and fill rates of the above four methods in different sequences. We can see that the results from OurMS+RSS show obvious improvements in the accuracy of the estimated depth, and these improvements come from better motion states. Compared with OurMS+RSS, the results from our method show better depth accuracy and fill rate, which is obtained from our novel cost volume building method. The average depth errors and fill rates are summarized in Table I, which again demonstrates that our method with dynamic baselines is better than the other methods.

Fig.13 shows five qualitative results. The first column in this figure shows the image pairs, while the second, third, fifth, and sixth column show the estimated depths from SGM, MiniSamp+RSS, OurMS+RSS and our method, respectively. The fourth column and seventh column illustrate the undistorted point clouds generated from the estimated depths for MiniSamp+RSS and our method, respectively. The last column illustrates the reference point cloud with respect to the left image, and it contains the occluded parts. We can see that the quality of the estimated depths from MiniSamp+RSS goes from good to poor in the five rows, all results from SGM are always very poor, and the depths estimated from our method are the best in all five rows.



(a) (b)

Fig. 12. Illustration of the change of depth accuracy and the fill rates of depth maps with increasing velocity for SGM, MiniSamp+RSS, OurMS+RSS and our method.

TABLE I
DEPTH EVALUATION

Method	Average depth error (m)	Average fill rate (per.)
SGM	1.739265	0.136424
MiniSamp+RSS	0.85359	0.390751
OurMS+RSS	0.32369	0.55322
Ours	**0.186341**	**0.767058**

C. Runtime analysis

Almost all our modules are built by C++ and run on a CPU, which takes much time to build the enormous cost volumes. The total time for processing an image pair is about 3 seconds. According to the runtime analysis in [18], a CUDA implementation can dramatically reduce the time cost and should be considered to solve this problem.

VI. CONCLUSION

In this paper, we present an algorithm to estimate a depth map from an instantaneous RS image pair. Compared with previous methods, our method estimates the instantaneous motion states and a depth map of the image pair stably. In future work, we plan to transfer our CPU implementation to a CUDA implementation and investigate how to combine the online extrinsic refinement with our method to increase the accuracy of depth estimation.

REFERENCES

[1] J. Sun, H. Y. Shum, and N. N. Zheng, "Stereo matching using belief propagation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2351, no. 7, pp. 510–524, 2002.

[2] Q. Yang, L. Wang, and R. Yang, "Real-time Global Stereo Matching Using Hierarchical Belief Propagation," in *Procedings of the British Machine Vision Conference 2006*. British Machine Vision Association, 2006, pp. 101.1–101.10.

[3] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[4] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, feb 2008.

[5] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6492 LNCS, no. PART 1, 2011, pp. 25–38.

[6] D. Hernandez-Juarez, A. Chacon, A. Espinosa, D. Vazquez, J. C. Moure, and A. M. Lopez, "Embedded real-time stereo estimation via Semi-Global Matching on the GPU," *Procedia Computer Science*, vol. 80, pp. 143–153, 2016.

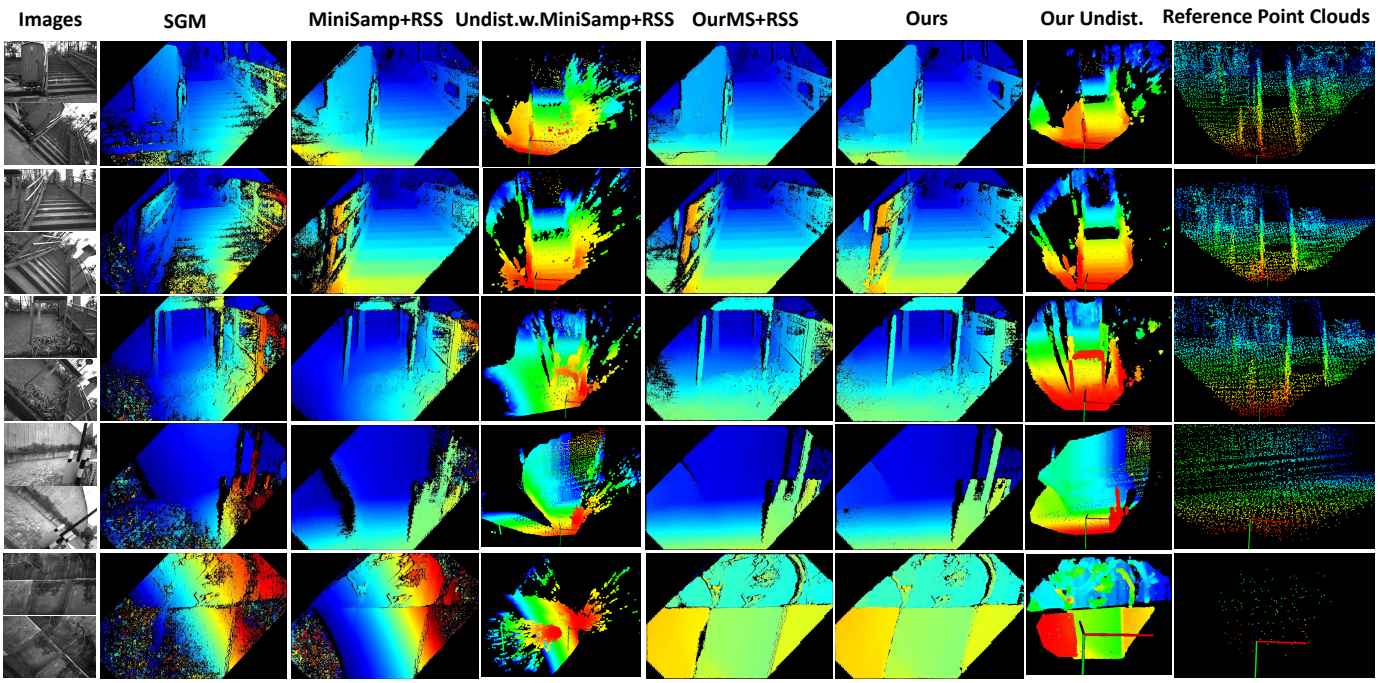


Fig. 13. The first column in this figure shows the left image of image pairs, the second column, third column, fifth column and sixth column show the estimated depths from SGM, MiniSamp+RSS, OurMS+RSS and our method, respectively. The fourth column and seventh column illustrate the undistorted point clouds generated from estimated depths for MiniSamp+RSS and our method, respectively. The last column illustrates the reference point cloud with respect to the left image, and it contains the occluded parts.

[7] M. Meingast, C. Geyer, and S. Sastry, "Geometric Models of Rolling-Shutter Cameras," 2005.

[8] P. Purkait, C. Zach, and A. Leonardis, "Rolling shutter correction in manhattan world," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 882–890.

[9] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," in *2012 IEEE international conference on computational photography (ICCP)*. IEEE, 2012, pp. 1–8.

[10] V. Rengarajan, A. N. Rajagopalan, and R. Aravind, "From bows to arrows: Rolling shutter rectification of urban scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2773–2781.

[11] Y. Lao and O. Ait-Aider, "A robust method for strong rolling shutter effects correction using lines with automatic feature selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4795–4803.

[12] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1360–1367, 2013.

[13] J. Hedborg, E. Ringaby, P.-E. Forssén, and M. Felsberg, "Structure and motion estimation from rolling shutter video," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011, pp. 17–23.

[14] O. Saurer, M. Pollefeys, and G. H. Lee, "A minimal solution to the rolling shutter pose estimation problem," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 1328–1334, 2015.

[15] Y. Dai, H. Li, and L. Kneip, "Rolling Shutter Camera Relative Pose: Generalized Epipolar Geometry," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 4132–4140, 2016.

[16] O. Saurer, M. Pollefeys, and G. H. Lee, "Sparse to Dense 3D Reconstruction from Rolling Shutter Images," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, no. June, pp. 3337–3345, 2016.

[17] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, no. March, pp. 16–22, 2000.

[18] O. Saurer, K. Koser, J. Y. Bouguet, and M. Pollefeys, "Rolling shutter stereo," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 465–472, 2013.

[19] C. Albl, Z. Kukelova, V. Larsson, M. Polic, T. Pajdla, and K. Schindler, "From two rolling shutters to one global shutter," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2505–2513.

[20] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, jul 2016.

[21] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision(ECCV)*. Springer, 2014, pp. 834–849.

[22] H. Alismail, B. Browning, and S. Lucey, "Photometric bundle adjustment for vision-based SLAM," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 324–341.

[23] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2013, pp. 1449–1456.

[24] D. Schubert, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "Direct sparse odometry with rolling shutter," *Lecture Notes in Computer Science*, vol. 11212 LNCS, pp. 699–714, 2018.

[25] D. Schubert, N. Demmel, L. von Stumberg, V. Usenko, and D. Cremers, "Rolling-Shutter Modelling for Direct Visual-Inertial Odometry," nov 2019.

[26] W. Gilks, *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, dec 1995.

[27] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, 2006.

[28] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 1280–1286.

[29] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *European Conference on Computer Vision*. Springer, 2008, pp. 302–315.