Pattern Recognition Letters 111 (2018) 1–8

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Robustified Structure from Motion with rolling-shutter camera using straightness constraint

Yizhen Lao[a,b,*], Omar Ait-Aider[a,b], Helder Araujo[c]

[a] Institut Pascal, Université Clermont Auvergne, Clermont-Ferrand 63000, France
[b] CNRS, Institut Pascal, UMR 6602 Aubière 63170, France
[c] Institute for Systems and Robotics, Dept. of Electrical and Computer Engineering, University of Coimbra, Coimbra 3030-290, Portugal

ARTICLE INFO

Article history:
Received 21 September 2017
Available online 4 April 2018

Keywords:
Rolling shutter
Structure from Motion
Ego-motion estimation
Bundle adjustment

ABSTRACT

We propose a 3-step method for structure and motion computation from two or more images taken by a one or multiple moving rolling shutter cameras. This work is motivated by the realization that existing reconstruction methods using rolling shutter images do not give satisfactory results or even fail in many configurations due to singularities and degenerate configurations. The first contribution consists in decoupling the rotate ego motion from the remaining parameters by adding a constraint on image curves basing on the a priori knowledge that they correspond to world 3D straight lines with unknown directions. Straight lines frequently appear in man-made environments such as urban or indoor scenes. After introducing the parameterization of a curve projected from a 3D straight line observed by a moving camera using three rolling shutter projection models, we show how to linearly extract angular velocity of each camera by using detected curves. Then we develop a linear method to recover the translational velocities and the motion between the cameras using point-matches, after compensating effects of angular velocity on each image. The second contribution consists in a novel point based bundle adjustment for rolling shutter cameras (C-RSBA) which does not consider a static row index during structure and motion optimization contrarily to existing methods. This enables to refine the parameters obtained thanks to the straightness constraint by avoiding degenerate configurations, thus outperforming existing RSBA methods. The approach was evaluated on both synthetic and real data.

1. Introduction

1.1. Motivation

Many modern CMOS cameras are equipped with rolling shutter (RS). In such acquisition mode, pixel rows are exposed sequentially from top to bottom. Therefore, images captured by moving RS cameras can have distortion effects (e.g. Wobble or Skew). RS effects must be considered in real Structure from Motion (SfM) applications where the camera moves fast such as UAVs or vehicles. Recently, more and more SfM methods were specifically designed for RS cameras addressing pose estimation [1,2], 3D reconstruction from stereo rig [3,16,17], bundle adjustment (BA) [10], relative pose problem [6], dense matching [12] and degeneracies [5,11]. Nevertheless, almost all existing works totally rely on detecting and matching point features. To the best of our knowledge, except in [2,14], line features have never been used despite the fact that they are abundant in many man-made environment such as building interiors or urban cityspaces. Furthermore, using straight lines as features offers several advantages such as detection accuracy and the possibility to handle partial occlusions.

When using RS cameras, straight segments can be rendered as curves under different kinematic models. If correctly parameterized, a curve corresponding to the projection of a 3D straight line will carry information about camera ego-motion. We will show that it is then possible to partially recover this ego-motion from curve parameters, thus making the structure and motion computation more consistent.

1.2. Related work

Due to the complexity and the high non-linearity of rolling shutter perspective projection model, strong assumptions which usually do not hold in practice, have been made in existing literatures in order to solve the SfM problem with RS cameras. Some approaches require continuity and “smoothing” of the movement during the shooting, but also between the views thus imposing very high rate frame [10,12] which makes both data transferring

* Corresponding author.
E-mail addresses: lyz91822@gmail.com, yizhen.lao@etu.uca.fr (Y. Lao).

https://doi.org/10.1016/j.patrec.2018.04.004
0167-8655/

and processing very time and memory consuming, not to mention the case where multiple cameras with wide baselines are used. Other approaches consider simplified movements such as pure translation [16], pure rotation [11] or small angular velocity [6]. We believe that a method based on a more general kinematic model and which handles wide baselines would give significant improvement not only in terms of accuracy in pose and motion estimation, but also in terms of automatic data matching performances (namely outliers rejection).

With numerous parameters and highly non-linear projection models, problems of local minima occur more frequently in RS bundle adjustment [10]. Some RS degeneracies were firstly reported in [3] that a pure translational motion nearly parallel to the baseline gives an infinity of solutions due to the coupling between shape and motion parameters. Albl et al. [5] and Ito and Okatani [11] analyzed the case of planar degeneracy which occurs most often for RS SfM and prove that images captured by cameras having parallel read-out directions is a critical motion sequence (CMS) with specific angular velocities as degenerate solutions. They both suggested that it could be avoided by using RS images with different readout directions, which is obviously not a convenient solution for practical applications.

One way to handle problems of degeneracy and local minima mentioned above consists in adding constraints on scene geometry. However, the constraint should be convenient and feasible in practical situations. Straight lines can be used to partially constrain the geometry of a scene. Advantages of using line features in computer vision are well known (vanishing point detection, uncoupling rotation and translation parameters, etc.). In the case of a moving RS camera, straight lines do not project as straight lines anymore but as curves whose shape depends from the motion during image scanning. Thus, motion parameters are hiding in the deviation from those curves to a straight line. This is the basis of the 'Straight line have to be straight' principle used in [7] to remove radial distortion effects.

Rengarajan et al. [14] propose to estimate the angular ego-motion by optimizing a non linear functional which forces image curves to be aligned with vanishing directions. The lines used here are assumed to comply with the so called Manhattan World Assumption (i.e. with orthogonal directions) which is a strong limitation according to the authors themselves.

In summary, SfM with moving RS cameras remains a topic with many open problems. How to use unordered two or more RS images with wide base-lines taken by one moving camera or by multiple cameras, to correctly reconstruct a 3D scene and estimate motion (avoiding degeneracies and without constraining the capture style)?

1.3. Paper contributions and content

We propose a method for the RS SfM problem by introducing a straightness constraint on image curves assuming that they are matched with 3D straight lines.

After introducing the perspective projection model for RS cameras (Section. 2), we show that the parameterization of the projection of a 3D straight line leads to a first, second or third degree polynomial depending on the kinematic model considered during image acquisition (Section. 3).

Thanks to this parameterization, we address the SfM problem for RS cameras in three steps. First, starting from a pair of RS images, on which curves corresponding to 3D straight lines are detected, the rotational part of the velocity is recovered for each image (Section. 4). Second, the SfM problem is solved by compensating effects of rotational speed on each image and then computing the remaining parameters (i.e. the translational velocity of each camera and the motion between them) using the 5×5 essential matrix seen in [6] (Section. 5). Finally, all the parameters are refined using a new BA technique which enables to avoid degeneracy reported in the state-of-the-art. Unlike existing methods, the proposed RS BA does not impose a constant row index on image points during optimization. This makes the projection at each iteration more consistent and thus constrains better structure and motion parameters. (Section. 6).

Experiments on both real and synthetic data shows that the proposed approach outperforms existing methods and handles cases where other approaches fail (Section. 7). In comparison to the closest existing work, our approach contributions can be summarized as follows:

- Parametric formulation of the projection of a 3D line under general motion model;
- Theoretical analysis of translational and angular velocities effect on the projection of a 3D line;
- Linear solution for the rotational ego-motion estimation without pre-knowledge about 3D straight lines directions or angles between these lines;
- A novel camera-based RS BA (C-RSBA) which handles common degenerate configurations and does not impose a specific capture style.

2. Rolling shutter camera model

In the static scene, a RS camera is equivalent to a global shutter (GS) one. It follows the classical pinhole projection model defined by intrinsic parameter matrix $\mathbf{K}$, rotation $\mathbf{R}$ and translation $\mathbf{T}$ between world and camera coordinate systems [9]:

$$s[\mathbf{m}, \mathbf{1}]^T = \mathbf{K}[\mathbf{R} \quad \mathbf{T}][\mathbf{P}, \mathbf{1}]^T \tag{1}$$

where s is a scale factor, $\mathbf{P} = [X, Y, Z]$ is a 3D point in the world coordinate system and , $\mathbf{m} = [u, v]$ is its image coordinates.

For a moving RS camera, each row will be captured at a different pose during frame exposure. Therefore, for a general camera motion model (with both angular and translational velocities), Eq. (1) becomes:

$$s[\mathbf{m}, \mathbf{1}]^T = \mathbf{K}[\mathbf{R}\delta\mathbf{R_i} \quad \mathbf{T} + \delta\mathbf{T_i}][\mathbf{P}, \mathbf{1}]^T \tag{2}$$

$\mathbf{R_i}$ and $\mathbf{T_i}$ are the rotation and the translation between time t_1 and t_i (t_i is the time of exposure of ith row). Since row-wise scanning speed is constant, we have $t_i = \tau v_i$ where τ is the time delay between two successive image line exposures. Usually τ for consumer cameras is short enough to make assumption that the camera is under uniform motion during one image acquisition. Therefore, rotation and translation can be formulated based on small rotate approximation of Rodrigues and translational velocity formulas:

$$\delta\mathbf{R_i} = \mathbf{I} + \tau v_i[\boldsymbol{\omega}]_\times \qquad \delta\mathbf{T_i} = \tau v_i \mathbf{d} \tag{3}$$

where $\mathbf{I}$ is the 3×3 identity matrix, $\mathbf{d}$ is the translational velocity while $[\boldsymbol{\omega}]_\times$ is the skew-symmetric matrix of $\boldsymbol{\omega}$.

3. Parametrization of 3D straight line projection

3.1. 3D Straight line representation

In this paper we adopt the convenient formulation used in [18] and which represents a 3D straight line in $\mathbf{R}^3$ as a tuple $\mathfrak{L} = <\mathbf{R}, (a, b)>$ with 4 degrees of freedom (DoF) as illustrated in Fig. 1.

3.2. 3D Line projection with a GS camera

With the assumption of a calibrated camera, intrinsic matrix $\mathbf{K}$ is known. Schindler et al. prove that the projection of a 3D line into a GS camera image can be divided into three main steps [18]:

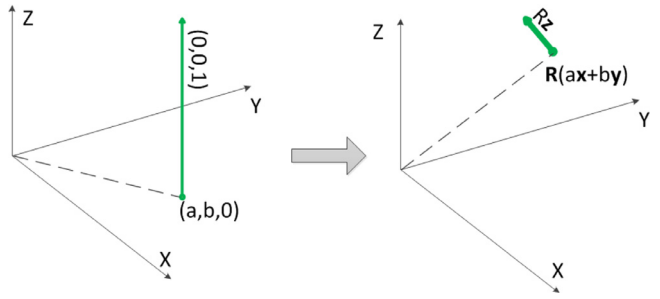

Fig. 1. 3D line representation. The line can be treated as parallel to Z-axis and passing through point $(a, b, 0)$ within XY-Plane (green line shown on left figure) which is then rotated by $\mathbf{R}$ to a new position (shown on the right figure). The final straight line passes through point $\mathbf{R}(ax + by)$, and is heading $\mathbf{R}z$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

• **Transformation into camera coordinate system.** We denote a 3D line in the world coordinate system as $<\mathbf{R_w}, (a_w, b_w)>$ and the transformation between camera coordinate frame and world frame as $\mathbf{R_c^w}$ and $\mathbf{t_c^w}$. The 3D straight line can be expressed in the camera coordinate system as:

$$\begin{aligned} \mathbf{R_c} = \mathbf{R_c^w R_w} \qquad & \mathbf{t_c} = (t_x, t_y, t_z)^T = (\mathbf{R_w})^T \mathbf{t_c^w} \\ & (a_c, b_c) = (a_w - t_x, b_w - t_y) \end{aligned} \tag{4}$$

• **Perspective projection.** The direction $\mathbf{m_{cip}} = [m_x, m_y, m_z]^T$ of a straight line so that $m_x u + m_y v + m_z = 0$ within plane at $z = 1$ in the camera frame can be calculated by the cross product of $\mathbf{R_c z}$ and $\mathbf{R_c}(a_c x + b_c y)$:

$$\mathbf{m_{cip}} = a_c \mathbf{R_{c2}} - b_c \mathbf{R_{c1}} \tag{5}$$

Where $\mathbf{R_{c2}}$ and $\mathbf{R_{c1}}$ are the second and first columns of $\mathbf{R_c}$.

• **Image space.** Image lines can be obtained as: $\mathbf{m_{ci}} = \left(\mathbf{K}^T\right)^{-1} \mathbf{m_{cip}}$. Finally, we can write the projection of a 3D straight line for a GS camera as follows:

$${}^{GS}\mathrm{F}_1 u + {}^{GS}\mathrm{F}_2 v + {}^{GS}\mathrm{F}_3 = 0 \tag{6}$$

3.3. 3D Line projection with uniform RS model

Under the realistic assumption of a uniform motion with both angular and translational velocities, the camera pose for a given row can be denoted by Eq. (3) as:

$$\begin{aligned} \mathbf{R_c} &= ((\mathbf{I} + [\boldsymbol{\omega}]_\times v])\mathbf{R_w^c}))^T \mathbf{R_w} \\ \mathbf{t_c} = (t_x, t_y, t_z)^T &= (\mathbf{R_w})^T (\mathbf{t_c^w} + \mathbf{dv}) \end{aligned} \tag{7}$$

Using the same reasoning than in the previous subsection, the projection of a point belonging to a 3D straight line leads now to the following parametric equation:

$$\begin{aligned} {}^{Unif}F_1 v^3 + {}^{Unif}F_2 v^2 u + {}^{Unif}F_3 v^2 + {}^{Unif}F_4 vu \\ + {}^{Unif}F_5 v + {}^{Unif}F_6 u + {}^{Unif}F_7 = 0 \end{aligned} \tag{8}$$

Seven coefficients are then defined by $\mathbf{K}$, 3D line parameters, camera initial pose ($\mathbf{R}$ and $\mathbf{T}$) and kinematics during image acquisition ($\mathbf{d}$, $\boldsymbol{\omega}$).

From the uniform model of Eq. (8), one can derive two simpler models: a linear RS model and a rotate-only model, which assume pure translation and pure rotation during image acquisition. By forcing translational velocity $\mathbf{d}$ and angular velocity $\boldsymbol{\omega}$ to be equal to 0 respectively, we will both obtain a hyperbolic curve. The parameterizations of a 3D straight line projection with different RS models are summarized in Table 1.

4. Extraction of angular velocities from curves

4.1. 16-Curves linear solution for uniform RS model

For a single RS image, if we assume the camera frame as world coordinate system, we obtain $\mathbf{R_w^c} = \mathbf{I}$ and $\mathbf{t_w^c} = [0, 0, 0]^T$. Then, based on Eq. (8), $\boldsymbol{\omega}$ can be denoted by seven coefficients of cubic curves (details in supplemental material):

$$\begin{bmatrix} C_1 & \cdots & C_{17} \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 & \cdots & \mathbf{W}_{17} \end{bmatrix}^T = 0 \tag{9}$$

where C_i are 17 auxiliary variables determined by $\mathbf{K}$ and cubic curve coefficients ${}^{unif}\mathrm{F}_1$ to ${}^{unif}\mathrm{F}_7$ while $\mathbf{W}_i$ are 17 vectors consisted by components of $\boldsymbol{\omega}$. Finally, this equation can be solved linearly by SVD with at least 16 detected curves.

4.2. Comparison of the three RS models

Some existed works assumed that only angular velocities play a main role for hand-held devices [14,15] and vehicles [8]. We try to give a qualitative and quantitative analysis of RS effects on 3D line projection. Although linear RS model will introduce a hyperbolic curve, its second order coefficients ${}^{Lin}F_1 = \mathbf{K_{22}}^{-T}(\mathbf{R_{w21}R_{w2}}^T - \mathbf{R_{w22}R_{w1}}^T)\mathbf{d}$, ${}^{Lin}F_2 = \mathbf{K_{11}}^{-T}(\mathbf{R_{w11}R_{w2}}^T - \mathbf{R_{w12}R_{w1}}^T)\mathbf{d}$ are provable much smaller compared to ${}^{Lin}F_3 = \mathbf{K_{22}}^{-T}(a_w\mathbf{R_{w22}} - b_w\mathbf{R_{w21}}) + \frac{\mathbf{K_{31}}^{-T}}{\mathbf{K_{11}}^{-T}}{}^{Lin}F_2 + \frac{\mathbf{K_{32}}^{-T}}{\mathbf{K_{22}}^{-T}}{}^{Lin}F_1 + (\mathbf{R_{w31}R_{w2}}^T - \mathbf{R_{w32}R_{w1}}^T)\mathbf{d}$ and ${}^{Lin}F_4 = \mathbf{K_{11}}^{-T}(a_w\mathbf{R_{w12}} - b_w\mathbf{R_{w11}})$ and can approximately be ignored in practice. The simulated experiment shown in Fig. 2 confirmed that even with high translational speeds, ${}^{Lin}F_1$, ${}^{Lin}F_2$ are relatively low, and projected curves (blue) are close to a straight line as for GS case (green). In practice, the effect of these last two parameters on the curves is even covered by the curve detection noise. Therefore, we chose to extract angular velocity based on the rotate-only RS model instead of the uniform model, which needs much more detected curves. However, the 16-curves method still can be used in very specific applications where the translational speed is very high in comparison to scan speed, and where curve detection can be achieved with a very high accuracy (sub-pixellic).

4.3. Practical 4-curves linear solution

It is provable that if we denote 3D line structural parameters as $a_c\mathbf{R_{w2}} - b_c\mathbf{R_{w1}} = [s_1, s_2, s_3]^T$, for five hyperbolic coefficients of each curve, we can formulate a group of equations:

$$\begin{aligned} F_1 = \mathbf{K_{22}}^{-T}(s_1\omega_3 - s_3\omega_1) \qquad & F_2 = \mathbf{K_{11}}^{-T}(s_3\omega_2 - s_2\omega_3) \\ F_3 = \mathbf{K_{22}}^{-T}s_2 + \mathbf{K_{31}}^{-T}(s_3\omega_2 - s_3\omega_3) & \\ + \mathbf{K_{32}}^{-T}(s_1\omega_3 - s_3\omega_1) + (s_2\omega_1 - s_1\omega_2) & \\ F_4 = \mathbf{K_{11}}^{-T}s_1 \qquad F_5 = \mathbf{K_{11}}^{-T}s_1 + \mathbf{K_{32}}^{-T}s_2 + s_3 & \end{aligned} \tag{10}$$

where s_1, s_2 and s_3 are different for each curve. There are six unknowns inside Eq. (10), therefore, with more curves we can extract $\boldsymbol{\omega}$ from curve coefficients.

Using five equations such as Eq. (10), s_1, s_2, s_3 and ω_3 can be substituted by ω_1 and ω_2 (more details are given in supplemental materials), we can obtain bivariate cubic polynomial. Where new coefficients C_1 to C_8 are only determined by $\mathbf{K}$ and coefficients F_1 to F_5. Now, by giving four curves, we have:

$$\begin{vmatrix} C_1^1 & \cdots & C_8^1 \\ \vdots & \ddots & \vdots \\ C_1^4 & \cdots & C_8^4 \end{vmatrix} [{\omega_1}^3, \omega_2{\omega_1}^2, {\omega_1}^2, \omega_2{}^2, \omega_1\omega_2, \omega_1, \omega_2, 1]^T = \mathbf{0} \tag{11}$$

Again, we substitute ω_2 by ω_1 and 32 coefficients C_i^j in Eq. (11) (details in supplemental materials), we obtain the follow-

Table 1
Parametric representation of 3D straight line projection with different RS models.

Camera model	Projection equation	Curve type	Parameters
GS camera	${}^{GS}F_1 u + {}^{GS}F_2 v + {}^{GS}F_3 = 0$	Straight line	**R, t**
Linear RS camera	${}^{Lin}F_1 v^2 + {}^{Lin}F_2 vu + {}^{Lin}F_3 v + {}^{Lin}F_4 u + {}^{Lin}F_5 = 0$	Hyperbolic curve	**R, t, d**
Rotate-only RS camera	${}^{Rot}F_1 v^2 + {}^{Rot}F_2 vu + {}^{Rot}F_3 v^{Rot} + F_4 u + {}^{Rot}F_5 = 0$	Hyperbolic curve	**R, t, ω**
Uniform RS camera	${}^{Unif}F_1 v^3 + {}^{Unif}F_2 v^2 u + {}^{Unif}F_3 v^2 + {}^{Unif}F_4 vu + {}^{Unif}F_5 v + {}^{Unif}F_6 u + {}^{Unif}F_7 = 0$	Cubic curve	**R, t, d, ω**

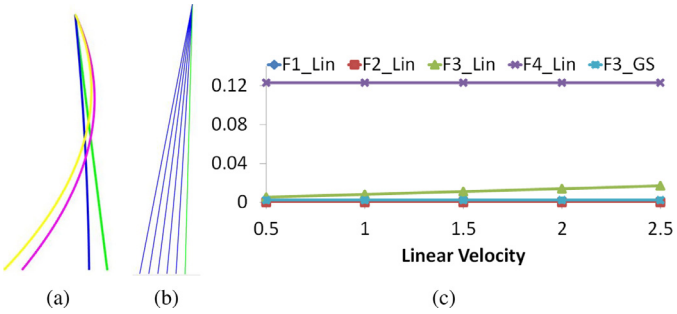

Fig. 2. Projections of 3D straight lines with different RS camera kinematics. (a) A simulated 3D straight line projected as different forms of curves in case of non ego-motion (green), translation-only (blue), rotate-only (pink) and uniform ego-motion (yellow). Assuming the depth from 3D straight line to camera as 1 unit length, blue curves in (b) are the projected 3D line with translational velocities from 0.5 to 2.5 unit/s, while green line is for the GS case. The variations of ${}^{Lin}F_1$, ${}^{Lin}F_2$, ${}^{Lin}F_3$ and ${}^{Lin}F_4$ and the constant value of ${}^{GS}F_3$ are shown in (c). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ing quartic equation:

$$(H_1, H_2, H_3, H_4, H_5)({\omega_1}^4, {\omega_1}^3, {\omega_1}^2, \omega_1, 1)^T = 0 \tag{12}$$

Thus, parameter ω_1 can be recovered by solving the Eq. (12) as a linear non-homogeneous system with the unknown vector $[\omega_1^4, \omega_1^3, \omega_1^2, \omega_1]^T$. Finally, ω_2 and ω_3 are recovered by substitution.

4.4. Straight line selection strategy

Curve pixels are detected and fitted in the way described in [14]. In order to distinguish curves which corresponds to actual 3D straight lines and 3D curves, we perform a filtering procedure by fitting curve pixels to cubic curve. The curves with big fitting errors will be discarded. A more complex process based on a RANSAC-like prediction verification will be used to discard curves which do not correspond to actual 3D straight lines.

5. SfM using extracted angular velocities

After extracting angular velocities for each image, we still need to recover both motion between cameras and the translational velocities for each camera. This is achieved as follows: first each image is rectified by compensating the angular velocities computed in the previous section. This results in a new image pair, which looks like if each camera undergoes pure translational motion during acquisition. Thus, the epipolar geometry between image pair is computed along with the translational velocities of the cameras using the linear RS model. The advantage of using linear RS model in SfM is to avoid planar degenerate solutions described [5,11]. This degeneracy is caused by the fact that using angular velocities as unknown parameters will collapse into specific values during non-linear optimization. Thus, using linear RS model by fixing angular velocities will avoid this degeneracy.

In order to compensate effects of $\boldsymbol{\omega}$, we perform an inverse mapping to all point-matches between images:

$$\mathbf{x}_{compensate} = \mathbf{K}\mathbf{R}(\mathbf{v})^{-1}\mathbf{K}^{-1}\mathbf{x}_{orginal} \tag{13}$$

This procedure maps original points $\mathbf{x}_{orginal}$ (matches between images) to $\mathbf{x}_{compensate}$. Now, the corrected images can be regarded as linear RS images. Rotation $\mathbf{R(v)}$ is calculated by using Eq. (3).

After the compensation, the relative pose problem of linear RS cameras can be solved by using the 5×5 essential matrix with point matches $[u, v]^T \leftrightarrow [u', v']^T$ proposed by Dai et al. [6]:

$$\left|v'^2 \quad v'u' \quad v' \quad u' \quad 1\right|\mathbf{E}_{5\times5}\left|v^2 \quad vu \quad v \quad u \quad 1\right|^T = 0 \tag{14}$$

With at least 20 point correspondences, $\mathbf{E}_{5\times5}$ can be computed as usual using a DLT (Direct Linear Transform) Algorithm. Then, the relative pose **[R, t]** and the translational velocities are extracted linearly from $\mathbf{E}_{5\times5}$. Finally, 3D points are reconstructed by triangulation [4].

6. Camera-based RS bundle adjustment

Considering a sequence of two or more images where the presented method is applied on each pair, a BA can be performed to

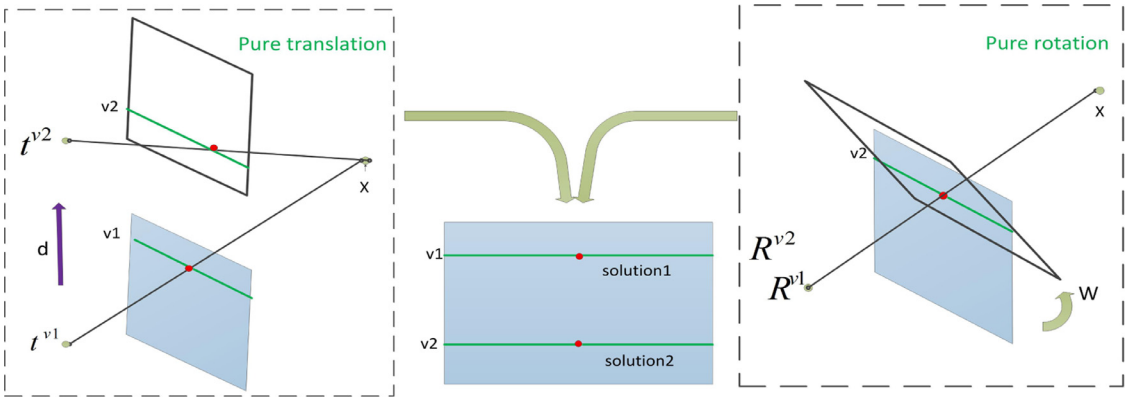

Fig. 3. Two examples of double-projections pattern. On the left, a RS camera is under pure translation heading to $[0; 1; 0]^T$ rapidly. Besides, a example of pure-rotation with axis (1,0,0) shown on the right. A 3D point **X** will be observed twice at row v_1 and v_2 if the speeds are big enough.

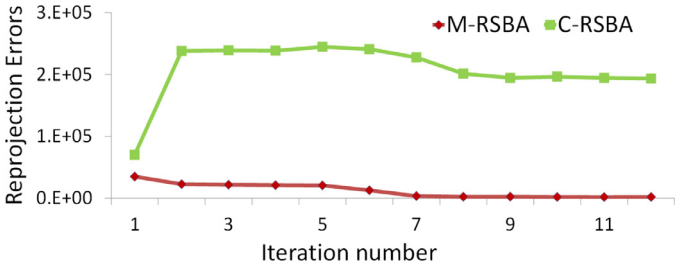

Fig. 4. Reprojection Errors Comparison. Results obtained with M-RSBA using multiple RS views with parallel read-out directions. For each iteration during optimization.

refine together camera poses, camera velocities and scene structure. Parameters obtained using the method described above for each pair are combined and used as starting points for an iterative minimization of a nonlinear cost function based on re-projection error. Assuming that l 3D points are observed on k images, projection errors will lead to the following non-linear error function:

$$\varepsilon(\mathbf{R}, \mathbf{t}, \boldsymbol{\omega}^j, \mathbf{d}^j) = \sum_{j=1}^{k} \sum_{i=1}^{l} \left| \tilde{\mathbf{m}}_\mathbf{i}^\mathbf{j} - \mathbf{m}_\mathbf{i}^\mathbf{j} \right|^2 \tag{15}$$

where $\tilde{\mathbf{m}}_\mathbf{i}^\mathbf{j}$ is the ith measurement on the jth image, while $\mathbf{m}_\mathbf{i}^\mathbf{j}$ is its respected reprojection point. $\boldsymbol{\omega}^\mathbf{j}$ and $\mathbf{d}^\mathbf{j}$ are rotate and translational speeds of the jth camera respectively.

M-RSBA. To the best of our knowledge, all existing works [5,10] used row index $\tilde{\mathbf{v}}_i^j$ of measurements $\tilde{\mathbf{m}}_\mathbf{i}^\mathbf{j}$ to calculate reprojection points $\mathbf{m}_\mathbf{i}^\mathbf{j}$. This is called measurement-based projection ($\mathfrak{p}^m$):

$$s[\mathbf{m}_\mathbf{i}^\mathbf{j}, \mathbf{1}]^T = \mathfrak{p}^m = \mathbf{K}[\mathbf{R}^\mathbf{j}\delta\mathbf{R}_\mathbf{i}^\mathbf{j}(\tilde{\mathbf{v}}_\mathbf{i}^\mathbf{j}) \quad \mathbf{T}^\mathbf{j} + \delta\mathbf{T}_\mathbf{i}^\mathbf{j}(\tilde{\mathbf{v}}_\mathbf{i}^\mathbf{j})][\mathbf{P}, \mathbf{1}]^T \tag{16}$$

$\delta\mathbf{R}_\mathbf{i}^\mathbf{j}(\tilde{\mathbf{v}}_i^j)$ and $\delta\mathbf{T}_\mathbf{i}^\mathbf{j}(\tilde{\mathbf{v}}_i^j)$ are obtained by using Eq. (3) based on measurements $\tilde{\mathbf{v}}_i^j$. This method uses image measurements as preknowledge to calculate reprojection points and makes exposure-delay of each point fixed. However, during parameter optimization, exposure-delays should change at each iteration in order to maintain structure and motion consistency according to row indexes. Thus, two drawbacks of M-RSBA appear: **i)** It cannot simulate true projection during optimization, which leads to loss of accuracy. **ii)** It brings risks of degeneracy as shown in [5].

C-RSBA. Alternatively, we propose a novelty approach to calculate reprojection points based on a pure RS camera model [$\mathbf{R}^j$, $\mathbf{T}^j$, $\boldsymbol{\omega}^j$, $\mathbf{d}^j$], which do not use fixed measurement indexes. We called it camera-based RS projection ($\mathfrak{p}^c$):

$$\mathbf{m}_i^j = \begin{pmatrix} u_i^j \\ v_i^j \end{pmatrix} = \mathfrak{p}^c = \begin{pmatrix} \frac{\mathbf{R}^{(1)}\mathbf{P}+v\hat{\mathbf{R}}^{(1)}\mathbf{P}+\mathbf{T}^{(1)}+v\mathbf{d}^{(1)}}{\mathbf{R}^{(3)}\mathbf{P}+v\hat{\mathbf{R}}^{(3)}\mathbf{P}+\mathbf{T}^{(3)}+v\mathbf{d}^{(3)}} \\ \frac{-b\pm\sqrt{-4ac+b^2}}{2a} \end{pmatrix} \tag{17}$$

where $\hat{\mathbf{R}}^{(k)}$, $\mathbf{R}^{(k)}$, $\mathbf{T}^{(k)}$ and $\mathbf{d}^{(k)}$ are the k^{th} row of $\hat{\mathbf{R}} = [\boldsymbol{\omega}^j]_\times \mathbf{R}_i^j$, $\mathbf{R}_i^j$, $\mathbf{T}_i^j$ and $\mathbf{d}^j$ respectively. a, b and c are three auxiliary variables defined as: $a = \hat{\mathbf{R}}^{(3)}\mathbf{X} + \mathbf{d}^{(3)}$, $b = \hat{\mathbf{R}}^{(3)}\mathbf{X} + \mathbf{T}^{(3)} - \hat{\mathbf{R}}^{(2)}\mathbf{X} - \mathbf{d}^{(2)}$ and $c = -\hat{\mathbf{R}}^{(2)}\mathbf{X} - \mathbf{T}^{(2)}$.

Double-projections pattern. The quadratic equation in Eq.-20 yields two theoretical solutions v_1 and v_2 named as double-projections pattern (shown in Fig. 3). In common and practical configurations, there are usually one solution located within image range while another one far away from image range. We analyze two typical cases in Fig. 3. A 3D point with 10 unit depth projected as two points within image range requires translation speed at least 500 unit/s and angular speed 50 rad/s which is huge for real applications.

Thus, since only one solution is consistent with the pose in practice. We propose to always select the solution that is nearer

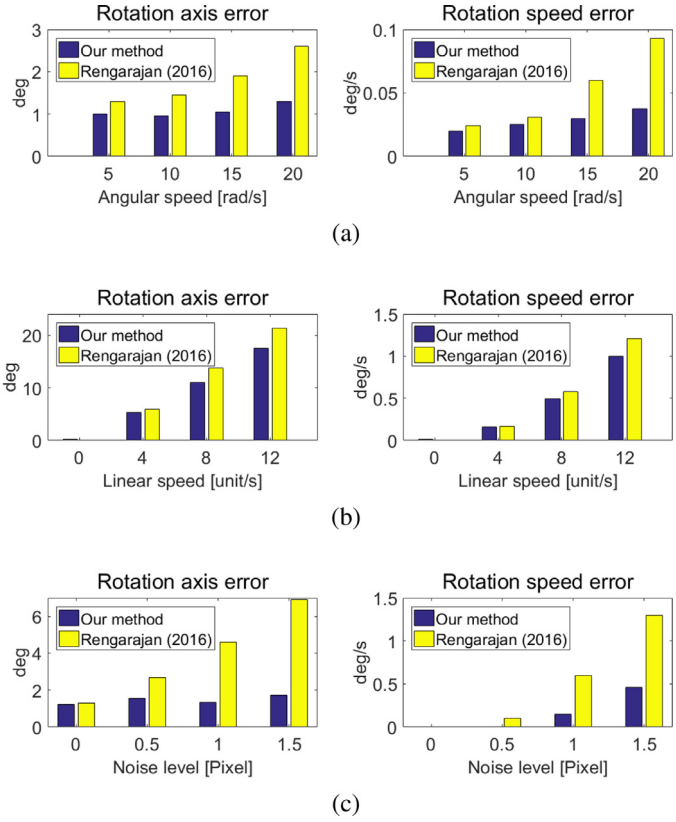

Fig. 5. Evaluation of ego-rotation computation: (a) effects of an increasing angular velocity. (b) Effects of an increasing translational velocity. (c) Effects of image noise level.

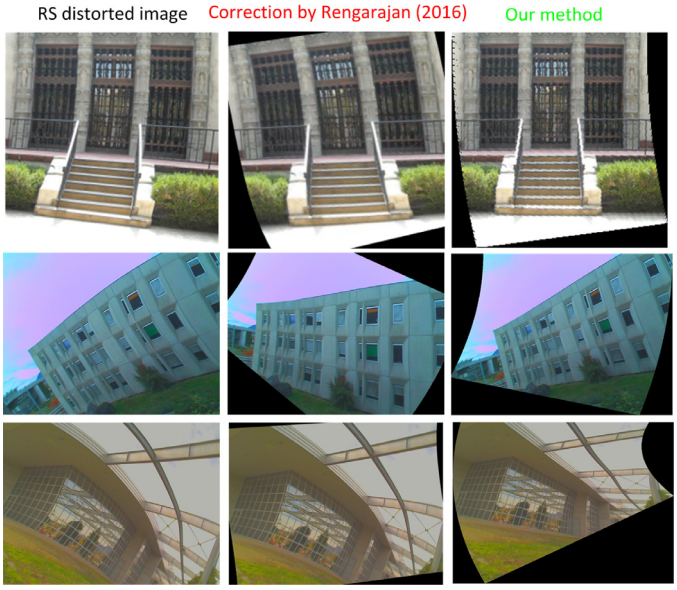

Fig. 6. Comparison of image rectification by compensating effects of ego-rotation: Synthetic RS image benchmark [13] (first row). Self-capture real RS images of urban scene (second and third rows).

to image by comparing reprojection values obtained by using v_1 and v_2 respectively during bundle adjustment (at each optimization round). This selection provides a solution that maintains point projection within the camera field of view.

Advantages of C-RSBA. Albl et al. [5] investigated the mechanism of planar degeneracy, which often raised during RSBA us-

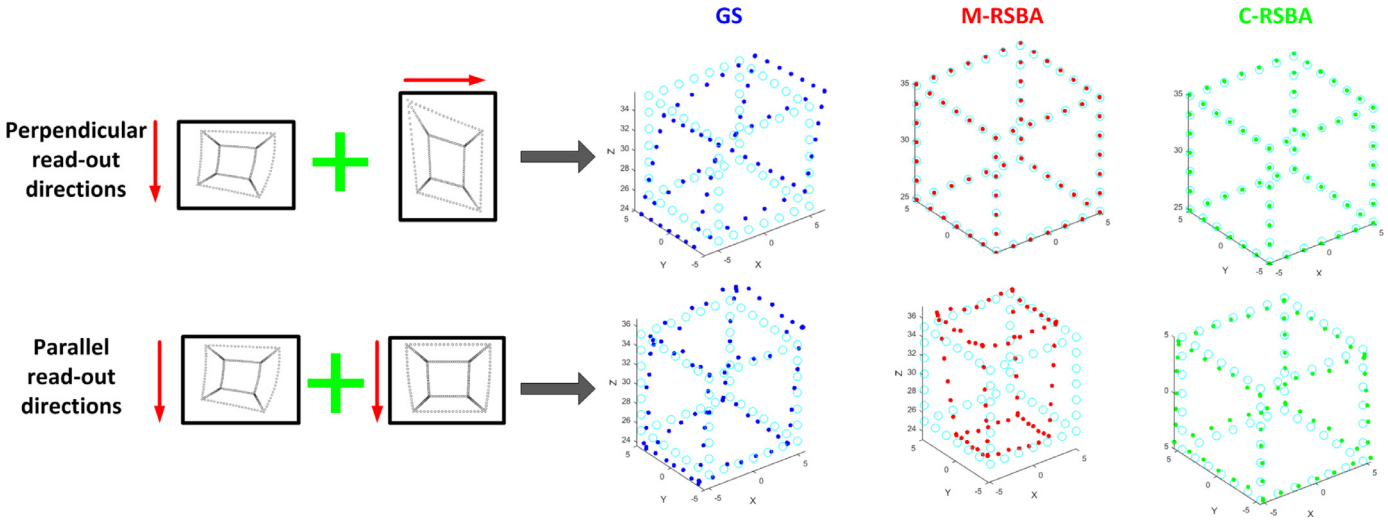

Fig. 7. Reconstruction results of GSBA (blue), M-RSBA (red) and C-RSBA (green) by using images with parallel and perpendicular read-out directions in comparison to ground-truth (cyan). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ing measurements-based projection. Multiple RS views with parallel read-out directions will collapse into solutions for which cameras have $\boldsymbol{\omega} = [-1\,0\,0]^T$ and 3D points located on $y = 0$ plane. In contrast, we found out that by using camera-based method to calculate reprojection errors, C-RSBA survives from degeneracy. The theoretical explanation for why M-RSBA suffers from degeneracy while C-RSBA survive from reprojection error standpoint is given below:

Proposition 1. *When RSBA collapses towards a planar degenerate solution, reprojection error calculated by* $\mathfrak{p}^m$ *gradually descends to 0 while errors using* $\mathfrak{p}^c$ *become huge (Fig. 4).*

We assume a RS camera with referenced pose $[\mathbf{I}, \mathbf{0}]$ *and ego-motion close to planar critical configuration as* $\boldsymbol{\omega} = [-1, 0, 0]^T$, $\mathbf{d} = \mathbf{0}$ *and 3D points close to* $\mathbf{X} = [X, 0, Z]^T$.

Lemma 1. *Reprojection error by using* $\mathfrak{p}^m$ *is,*

$$\mathbf{e} = [e_u \quad e_v]^T = \tilde{\mathbf{m}} - \mathfrak{p}^m(\mathbf{C}, \mathbf{X}, \tilde{v}) = [\tilde{u} - \frac{X}{Z} \quad 0]^T$$

Simultaneously, $[X, 0, Z]^T$ *be further optimized to make* e_u *also reduced to 0. Finally, overall error* $\mathbf{e}$ *will descend to be 0.*

Lemma 2. *Reprojection error by using* $\mathfrak{p}^c$ *is,*

$$e = [e_u \quad e_v]^T = \tilde{\mathbf{m}} - \mathfrak{p}^{\mathbf{c}}(\mathbf{C}, \mathbf{X}) = [\tilde{u} - \frac{X}{Z} \quad \tilde{v}]^T$$

The overall reprojection becomes $\left|v_i^j\right|$, *which is even larger than reprojection error of the start point.*

Through Lemmas 1 and 2, one can observe that planar degenerate solution is a perfect minimum for cost function of M-RSBA while being a plateau for C-RSBA. This explains how C-RSBA successfully avoid planar degeneracy. An example of reprojection errors of M-RSBA and C-RSBA when configurations are slipping towards planar degeneracy (shown in Fig. 4) illustrates Proposition 1.

Thus, without constraints on camera motions such as perpendicular read-out directions among RS views [5], C-RSBA can successfully avoid planar degeneracy.

7. Experiments

7.1. Angular velocity extraction with synthetic images

A virtual scene composed of several sets of aligned 3D points has been constructed. Images corresponding to random angular velocities during acquisition were generated using the following virtual camera parameters: focal length = 1 unit, resolution = 640×480 pixel and scan speed = 7.5×10^{-5}s/row. Then values of $\boldsymbol{\omega}$ were computed from deformed edges using the linear approach described in Section. 4.3. While ground-truths are available, we evaluated the recovered $\boldsymbol{\omega}$ from rotate axis $\mathbf{a}$ and rotate speed ω basing on the following distances: $\mathbf{a_{error}} = subspace(\mathbf{a}, \mathbf{a_{GT}})$ and $\boldsymbol{\omega_{error}} = |\boldsymbol{\omega} - \boldsymbol{\omega_{GT}}|$. We repeated each experiment 100 times to get representative statistical results.

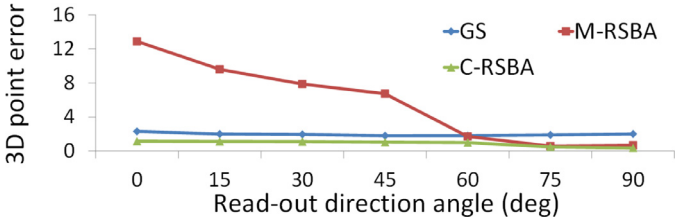

Fig. 8. Reconstruction errors of GSBA, M-RSBA and C-RSBA with read-out direction angles varying from 0 to 90°. M-RSBA only provides better results than GSBA when read-out direction angles are big (higher than 60°). C-RSBA obtains accurate and stable reconstructions independently from the read-out direction angles.

We compare our method with the relative single-image based RS ego-motion estimation method by Rengarajan et al. [14].[1] We draw attention to the fact that the latter method requires 3D scene, which comply Manhattan World (orthogonal 3D lines) while ours handles more general cases without pre-knowledge about directions of scene 3D lines.

Accuracy vs Angular velocities. Experiments were carried out with $|\boldsymbol{\omega}|$ varying from 5 to 20 rad/s. The results in Fig. 5(a) show that we can stably estimate rotate axis and speed. The proposed method performs better than the method of Rengarajan et al. [14].

Accuracy vs Linear velocities. In this experiment, we increased translational velocity $\mathbf{d}$ from 0 to 12 unit/s (1 unit = scene depth) which is too high and rarely occur in practice. The results in Fig. 5(b) confirms our analysis in Section. 4.2 that $\boldsymbol{\omega}$ play much more important role for RS effects than $\mathbf{d}$, which confirms that the used model is relevant to realistic computer vision applications.

[1] We self-implement the method and used in simulated experiments. The results in synthetic and real RS images are supplied by the authors upon request.

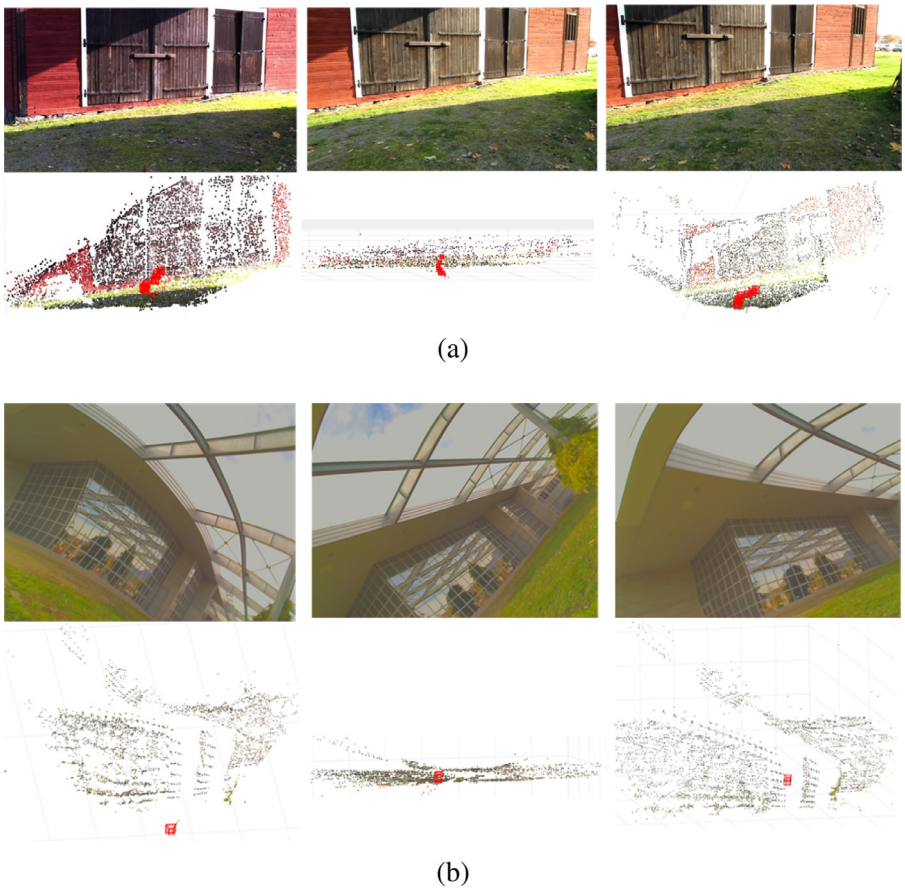

Fig. 9. SfM with similar read-out directions. Reconstruction results are obtained by GSBA (left), M-RSBA (middle) and C-RSBA (right). Obviously, M-RSBA surfers from planar degeneracy, while significant deformations can also be observed in GSBA reconstructions. C-RSBA provides correct reconstructed 3D scene.

Accuracy vs Pixel noise. We fixed the RS cameras under 1 unit/s translational speed and 5 rad/s rotate speed. Then we added random Gaussian noise to the projected pixels from 0 to 1.5 pixel. The results in Fig. 5(c) demonstrate that the proposed approach is more robust than [14].

7.2. Angular velocity extraction and compensation on real images

We evaluated angular velocity extraction on both synthetic and real RS images. Fig. 6 shows that the proposed first step linear approach can successfully obtain angular velocities of RS cameras. After $\boldsymbol{\omega}$ compensation, distorted RS images become only affected by remaining translational velocities. From the results, there are still significant curvature left in images rectified of method by Rengarajan et al. [14], while our method obtains better visual corrections. This demonstrate the effectiveness of angular velocity extraction algorithm compared superior to state-of-the-art work.

7.3. Evaluation of bundle adjustment

Since the angles between read-out directions among image sequence have significant impact on final reconstruction quality, we designed a simulation experiment to evaluate GSBA, M-RSBA (initialized by GSBA) and C-RSBA (initialized by the proposed linear two-step method). Three cameras are generated randomly on a sphere with a radius of 1 unit and heading to a cubical scene with varying average scanning angles from 0 to 90°. In Fig. 7, a deformed 3D cube is being reconstructed by GSBA in both parallel and perpendicular read-out directions cases. M-RSBA obtains correct reconstruction using images with perpendicular read-out directions but fails in parallel one, which is a commoner configuration in practical applications. The proposed C-RSBA reconstructs a correct 3D scene in both parallel and perpendicular cases.

In order to draw a quantitative conclusion, we used the sum of distances between reconstructed 3D points and ground-truth 3D points as a criteria to evaluate SfM performances. Results in Fig. 8 show that M-RSBA achieves better reconstruction than GSBA when read-out direction angles are bigger than 60 °, while C-RSBA obtains higher-accuracy and is more stable with close read-out directions (below 30 °).

Finally, we compare GSBA, M-RSBA and C-RSBA on two real RS image sequences. The first data set [10] captured by an iPhone4 camera for the facade of warehouse and a road along wall. The second dataset is a real complex building captured by a Logitech camera with strong RS effects. All RS images are captured with small read-out direction angles. The results shown in Fig. 9 confirmed our prediction in Section. 6 and coincide the simulation experiments. GSBA gives distorted reconstruction since RS effects presence. One can observe that the more strong distortion in RS image, the more deformations after SfM for GSBA. It is important to realize that M-RSBA cannot handle the case where input RS images with small scanning direction angles (less than 60 °). Strong deformations (flattened scenes) were observed in the 3D scene reconstructed with M-RSBA.

Quite the contrary, C-RSBA provides significantly better reconstructions than GSBA and M-RSBA, which collapse into degeneracy. This experiment shows that C-RSBA is feasible independently from image capture mode.

8. Conclusions

A 3-step method which solves RS SfM was presented. Unlike with existing methods, a general motion model is assumed and no a-priori knowledge on the 3D lines is needed. Moreover, the first two steps of the proposed solution are linear and work with fewer matches than previous methods. We also provide a novelty C-RSBA refinement method, which can successfully avoid planar degeneracy without any constraint on read out direction as in existing approaches. Note that image capture style with similar read-out directions are extremely natural and common in real applications while requirements of two distinct read-out directions will extensively limit the application range. Experiments with both real and synthetic data prove that the proposed method outperforms existing ones and can handle degeneracies pointed out in the literature. We believe that this work will help to take an extra step toward the use of RS cameras in SfM applications. Finally, since it can handle very strong RS effects, the proposed method can also be seen as a monocular ego-speed measurement technique.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2018.04.004.

References

[1] O. Ait-Aider, N. Andreff, J.M. Lavest, P. Martinet, Simultaneous object pose and velocity computation using a single view from a rolling shutter camera, in: European Conference on Computer Vision, Springer, 2006, pp. 56–68.
[2] O. Ait-Aider, A. Bartoli, N. Andreff, Kinematics from lines in a single rolling shutter image, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–6.
[3] O. Ait-Aider, F. Berry, Structure and kinematics triangulation with a rolling shutter stereo rig., in: Proceedings of the IEEE International Conference on Computer Vision, 2009. 18351840 doi: 10.1109/ICCV.2009.5459408.
[4] C. Albl, Z. Kukelova, T. Pajdla, Rolling shutter absolute pose problem with known vertical direction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3355–3363.
[5] C. Albl, A. Sugimoto, T. Pajdla, Degeneracies in rolling shutter sfm, in: European Conference on Computer Vision, Springer, 2016, pp. 36–51.
[6] Y. Dai, H. Li, L. Kneip, Rolling shutter camera relative pose: generalized epipolar geometry, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4132–4140.
[7] F. Devernay, O. Faugeras, Straight lines have to be straight, Mach. Vis. Appl. 13 (1) (2001) 14–24.
[8] G. Duchamp, O. Ait-Aider, E. Royer, J.-M. Lavest, A rolling shutter compliant method for localisation and reconstruction., in: VISAPP (3), 2015, pp. 277–284.
[9] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.
[10] J. Hedborg, P.-E. Forssen, M. Felsberg, E. Ringaby, Rolling shutter bundle adjustment, Computer Vision and Pattern Recognition (CVPR), 2012.
[11] E. Ito, T. Okatani, Self-calibration-based approach to critical motion sequences of rolling-shutter structure from motion, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2, 2017.
[12] J.H. Kim, C. Cadena, I. Reid, Direct semi-dense slam for rolling shutter cameras, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1308–1315, doi:10.1109/ICRA.2016.7487263.
[13] V. Rengarajan, Y. Balaji, A. Rajagopalan, Unrolling the shutter: cnn to correct motion distortions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2291–2299.
[14] V. Rengarajan, A.N. Rajagopalan, 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.
[15] E. Ringaby, P.-E. Forssén, Efficient video rectification and stabilisation for cell–phones, Int. J. Comput. Vis. 96 (3) (2012) 335–352.
[16] O. Saurer, K. Koser, J.-Y. Bouguet, M. Pollefeys, Rolling shutter stereo, IEEE Int. Conf. Comput. Vis. (ICCV) (2013).
[17] O. Saurer, M. Pollefeys, G. Hee Lee, Sparse to dense 3d reconstruction from rolling shutter images, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[18] G. Schindler, P. Krishnamurthy, F. Dellaert, Line-based structure from motion for urban environments, in: Third International Symposium on 3D Data Processing, Visualization, and Transmission, IEEE, 2006, pp. 846–853.