Rolling-Shutter-Aware Differential SfM and Image Rectification

Bingbing Zhuang, Loong-Fah Cheong, Gim Hee Lee
National University of Singapore
zhuang.bingbing@u.nus.edu, {eleclf,gimhee.lee}@nus.edu.sg

Abstract

In this paper, we develop a modified differential Structure from Motion (SfM) algorithm that can estimate relative pose from two consecutive frames despite of Rolling Shutter (RS) artifacts. In particular, we show that under constant velocity assumption, the errors induced by the rolling shutter effect can be easily rectified by a linear scaling operation on each optical flow. We further propose a 9-point algorithm to recover the relative pose of a rolling shutter camera that undergoes constant acceleration motion. We demonstrate that the dense depth maps recovered from the relative pose of the RS camera can be used in a RS-aware warping for image rectification to recover high-quality Global Shutter (GS) images. Experiments on both synthetic and real RS images show that our RS-aware differential SfM algorithm produces more accurate results on relative pose estimation and 3D reconstruction from images distorted by RS effect compared to standard SfM algorithms that assume a GS camera model. We also demonstrate that our RS-aware warping for image rectification method outperforms state-of-the-art commercial software products, i.e. Adobe After Effects and Apple Imovie, at removing RS artifacts.

1. Introduction

In comparison with its global shutter (GS) counterpart, rolling shutter (RS) cameras are more widely used in commercial products due to its low cost. Despite this, the use of RS cameras in computer vision such as motion/pose estimation is significantly limited compared to the GS cameras. This is largely due to the fact that most existing computer vision algorithms such as epipolar geometry [9] and SfM [26, 6] make use of the global shutter pinhole camera model which does not account for the so-called rolling shutter effect caused by camera motion. Unlike a GS camera where the photo-sensor is exposed fully at the same moment, the photo-sensor of a RS camera is exposed in a scanline-by-scanline fashion due to the exposure/readout modes of the low-cost CMOS sensor. As a result, the image taken from a moving RS camera is distorted as each scanline possesses a different optical center. An example is shown in Fig. 1(a), where the vertical tree trunk in the image captured by a RS camera moving from right to left appears to be slanted.

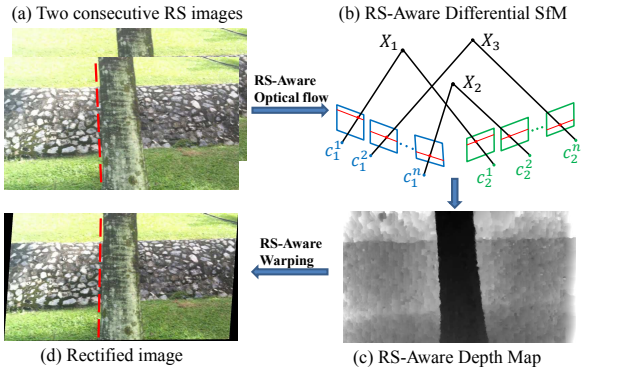

Figure 1: Illustration of our RS-aware differential SfM and image rectification pipeline. See text for more detail.

Due to the price advantage of the RS camera, many researchers began to propose 3D computer vision algorithms that aim to mitigate the RS effect over the recent years. Although several works have successfully demonstrated stereo [21], sparse and dense 3D reconstruction [15, 23] and absolute pose estimation [1, 22, 17] using RS images, most of these works completely bypassed the initial relative pose estimation, e.g. by substituting it with GPS/INS readings. This is because the additional linear and angular velocities of the camera that need to be estimated due to the RS effect significantly increases the number of unknowns, making the problem intractable. Thus, despite the efforts from [5] to solve the RS relative pose estimation problem under discrete motion, the proposed solution is unsuitable for practical use due to the need for high number of image correspondences that prohibits robust estimation with RANSAC.

In this paper, we aim to correct the RS induced inaccuracies in SfM across two consecutive images under continuous motion by using a differential formulation. In contrast to the discrete formulation where 12 additional motion parameters from the velocities of the camera need to be solved, we show that in the differential case, the poses of each scanline can be related to the relative pose of two consecutive images under suitable motion models, thus obviat-

ing the need for additional motion parameters. Specifically, we show that under a constant velocity assumption, the errors induced by the RS effect can be easily rectified by a linear scaling operation on each optical flow, with the scaling factor being dependent on the scanline position of the optical flow vector. To relax the restriction on the motion, we further propose a nonlinear 9-point algorithm for a camera that undergoes constant acceleration motion. We then apply a RS-aware non-linear refinement step that jointly improves the initial structure and motion estimates by minimizing the geometric errors. Besides resulting in tractable algorithms, another advantage of adopting the differential SfM formulation lies in the recovery of dense depth map, which we leverage to design a RS-aware warping to remove RS distortion and recover high-quality GS images. Our algorithm is illustrated in Fig. 1. Fig. 1(a) shows an example of the input RS image pair where a vertical tree trunk appears slanted (red line) from the RS effect, and Fig. 1(d) shows the vertical tree trunk (red line) restored by our RS rectification. Fig. 1(b) illustrates the scanline-dependent camera poses for a moving RS camera. c_i^j denotes the optical center of the scanline j in camera i. Fig. 1(c) shows the RS-aware depth map recovered after motion estimation. Experiments on both synthetic and real RS images validate the utility of our RS-aware differential SfM algorithm. Moreover, we demonstrate that our RS-aware warping produces rectified images that are superior to popular image/video editing software.

2. Related works

Meingast *et al.* [18] was one of the pioneers to study the geometric model of a rolling shutter camera. Following this work, many 3D computer vision algorithms have been proposed in the context of RS cameras. Saurer *et al.* [23] demonstrated large-scale sparse to dense 3D reconstruction using images taken from a RS camera mounted on a moving car. In another work [21], Saurer *et al.* showed stereo results from a pair of RS images. [13] showed high-quality 3D reconstruction from a RS video under small motion. Hedborg *et al.* [11] proposed a bundle adjustment algorithm for RS cameras. In [15], a large-scale bundle adjustment with a generalized camera model is proposed and applied to 3D reconstructions from images collected with a rig of RS cameras. Several works [1, 22, 17] were introduced to solve the absolute pose estimation problem using RS cameras. All these efforts demonstrated the potential of applying 3D algorithms to RS cameras. However, most of them avoided the initial relative pose estimation problem by taking the information directly from other sensors such as GPS/INS, relying on the global shutter model to initialize the relative pose or assuming known 3D structure.

Recently, Dai *et al.* [5] presented the first work to solve the relative pose estimation problem for RS cameras. They tackled the discrete two-frame relative pose estimation problem by introducing the concept of generalized essential matrix to account for camera velocities. However, 44 point correspondences are needed to linearly solve for the full motion. This makes the algorithm intractable when robust estimation via the RANSAC framework is desired for real-world data. In contrast, we look at the differential motion for two-frame pose estimation where we show that the RS effect can be compensated in a tractable way. This model permits a simpler derivation that can be viewed as an extension of conventional optical flow-based differential pose estimation algorithms [16, 24, 27] designed for GS cameras. Another favorable point for the optical flow-based differential formulation is that unlike the region-based discrete feature descriptors used in the aforementioned correspondence-based methods, the brightness constancy assumption used to compute optical flow is not affected by RS distortion as observed in [2].

Several other research attempted to rectify distortions in images caused by the RS effect. Forssén *et al.* [20] reduced the RS distortion by compensating for 3D camera rotation, which is assumed to be the dominant motion for hand-held cameras. Some later works [14, 8] further exploited the gyroscope on mobile devices to improve the measurement of camera rotation. Grundmann *et al.* [7] proposed the use of homography mixtures to remove the RS artifacts. Baker *et al.* [2] posed the rectification as a temporal super-resolution problem to remove RS wobble. Nonetheless, the distortion is modeled only in the 2D image plane. To the best of our knowledge, our rectification method is the first that is based on full motion estimation and 3D reconstruction. This enables us to perform RS-aware warping which returns high-quality rectified images as shown in Sec. 6.

3. GS Differential Epipolar Constraint

In this section, we give a brief description of the *differential epipolar constraint* that relates the infinitesimal motion between two *global shutter* camera frames. Since this section does not contain our contributions, we give only the necessary details to follow the rest of this paper. More details of the algorithm can be found in [16, 27]. Let us denote the linear and angular velocities of the camera by $\boldsymbol{v} = [v_x, v_y, v_z]^T$ and $\boldsymbol{w} = [w_x, w_y, w_z]^T$. The velocity field $\boldsymbol{u}$ on the image plane induced by $(\boldsymbol{v}, \boldsymbol{w})$ is given by:

$$\boldsymbol{u} = \frac{\boldsymbol{A}\boldsymbol{v}}{Z} + \boldsymbol{B}\boldsymbol{w}, \tag{1}$$

where

$$\boldsymbol{A} = \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}. \tag{2}$$

(x, y) is the normalized image coordinate and Z is the corresponding depth of each pixel. In practice, $\boldsymbol{u}$ is approximated by optical flow under brightness constancy assumption. Given a pair of image position and optical flow vector

$(\boldsymbol{x}, \boldsymbol{u})$, Z can be eliminated from Eq. (1) to yield the differential epipolar constraint [1]:

$$\boldsymbol{u}^T \hat{\boldsymbol{v}} \boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{s} \boldsymbol{x} = 0, \tag{3}$$

where $\boldsymbol{s} = \frac{1}{2}(\hat{\boldsymbol{v}}\hat{\boldsymbol{w}} + \hat{\boldsymbol{w}}\hat{\boldsymbol{v}})$ is a symmetric matrix. $\hat{\boldsymbol{v}}$ and $\hat{\boldsymbol{w}}$ represent the skew-symmetric matrices associated with $\boldsymbol{v}$ and $\boldsymbol{w}$ respectively. The space of all the matrices having the same form as $\boldsymbol{s}$ is called the *symmetric epipolar space*. The 9 unknowns from $\boldsymbol{v}$ and $\boldsymbol{s}$ (3 + 6 respectively) can be solved linearly from at least 8 measurements $(\boldsymbol{x}_j, \boldsymbol{u}_j), \forall\ j = 1, ..., 8$. The solution returned by the linear algorithm is then projected onto the symmetric epipolar space, followed by the recovery of $(\boldsymbol{v},\boldsymbol{w})$ as described in [16]. Note that $\boldsymbol{v}$ can only be recovered up to scale.

It is well to remember here that in actual computation, assuming small motion between two frames, all the instantaneous velocity terms will be approximated by displacement over time. Removing the common factor of time, the optical flow vector $\boldsymbol{u}$ now indicates the displacement of pixel over two consecutive frames, and the camera velocities $(\boldsymbol{v}, \boldsymbol{w})$ indicate the relative pose of the two camera positions. The requisite mapping between the relative orientation $\boldsymbol{R}$ and the angular velocity $\boldsymbol{w}$ is given by the well-known mapping $\boldsymbol{R} = \exp(\boldsymbol{w}) \simeq \boldsymbol{I} + \hat{\boldsymbol{w}}$. Henceforth, we will call $(\boldsymbol{v}, \boldsymbol{w})$ relative motion/pose for the rest of this paper. We utilize Deepflow [25] to compute the optical flow for all our experiments due to its robust performance in practice.

4. RS Differential Epiplor Constraint

4.1. Constant Velocity Motion

The differential epipolar constraint shown in the previous section works only on images taken with a GS camera and would fail if the images were taken with a RS camera. The main difference between the RS and GS camera is that we can no longer regard each image as having one single camera pose. Instead we have to introduce a new camera pose for each scanline on a RS image as shown in Fig. 1(b) due to the readout and delay times as illustrated in Fig. 2.

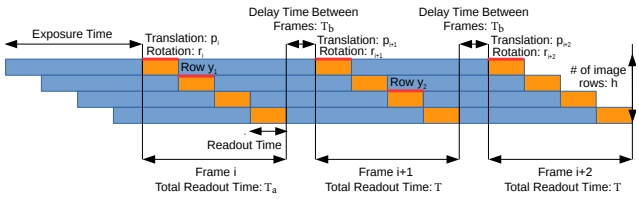

Figure 2: Illustration of exposure, readout and delay times in a rolling shutter camera.

Consider three consecutive image frames i, i+1 and i+2. Let $\{\boldsymbol{p}_i,\ \boldsymbol{p}_{i+1},\ \boldsymbol{p}_{i+2}\}$ and $\{\boldsymbol{r}_i,\ \boldsymbol{r}_{i+1},\ \boldsymbol{r}_{i+2}\} \in so(3)$ represent the translation and rotation of the first scanlines on the respective images as shown in Fig. 2. Frame i is set as the reference frame, i.e. $(\boldsymbol{p}_i, \boldsymbol{r}_i) = (\boldsymbol{0}, \boldsymbol{0})$. Now consider an optical flow which maps an image point from (x_1, y_1) in frame i to (x_2, y_2) in frame $i + 1$. Assuming constant instantaneous velocity of the camera across these three frames under small motion, we can compute the translation and rotation $(\boldsymbol{p}_1, \boldsymbol{r}_1)$ of scanline y_1 on frame i as a linear interpolation between the poses of the first scanlines from frames i and $i + 1$:

$$\boldsymbol{p}_1 = \boldsymbol{p}_i + \frac{\gamma y_1}{h}(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i), \tag{4a}$$

$$\boldsymbol{r}_1 = \boldsymbol{r}_i + \frac{\gamma y_1}{h}(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i). \tag{4b}$$

h is the total number of scanlines in the image. $\gamma = \frac{T_a}{T_a + T_b}$ is the readout time ratio which can be obtained a priori from calibration [18]. Similar interpolation for the pose $(\boldsymbol{p}_2, \boldsymbol{r}_2)$ of the scanline y_2 on frame $i + 1$ can be done between the first scanlines from frames $i + 1$ and $i + 2$:

$$\boldsymbol{p}_2 = \boldsymbol{p}_{i+1} + \frac{\gamma y_2}{h}(\boldsymbol{p}_{i+2} - \boldsymbol{p}_{i+1}), \tag{5a}$$

$$\boldsymbol{r}_2 = \boldsymbol{r}_{i+1} + \frac{\gamma y_2}{h}(\boldsymbol{r}_{i+2} - \boldsymbol{r}_{i+1}). \tag{5b}$$

Now we can obtain the relative motion $(\boldsymbol{p}_{21}, \boldsymbol{r}_{21})$ between the two scanlines y_2 and y_1 by taking the difference of Eq. (5) and (4), and setting $(\boldsymbol{p}_{i+2} - \boldsymbol{p}_{i+1}) = (\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)$ and $(\boldsymbol{r}_{i+2} - \boldsymbol{r}_{i+1}) = (\boldsymbol{r}_{i+1} - \boldsymbol{r}_i)$ due to the constant velocity assumption:

$$\begin{aligned} \boldsymbol{p}_{21} &= \underbrace{\left(1 + \frac{\gamma}{h}(y_2 - y_1)\right)}_{\alpha}(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i) \\ \Rightarrow \boldsymbol{p}_{21} &= \alpha(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i), \end{aligned} \tag{6a}$$

$$\begin{aligned} \boldsymbol{r}_{21} &= \underbrace{\left(1 + \frac{\gamma}{h}(y_2 - y_1)\right)}_{\alpha}(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i) \\ \Rightarrow \boldsymbol{r}_{21} &= \alpha(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i). \end{aligned} \tag{6b}$$

α is the dimensionless scaling factor of the relative pose made up of γ, h, y_1 and y_2. It was mentioned in the previous section that under small motion, $(\boldsymbol{v}, \boldsymbol{w})$ in the differential epipolar constraint can be regarded as the relative pose of the camera in practice. We can thus substitute $(\boldsymbol{v}, \boldsymbol{w})$ from Eq. (3) with the relative pose $(\boldsymbol{p}_{21}, \boldsymbol{r}_{21})$ from Eq. (6). Consequently, we get the rolling shutter differential epipolar constraint

$$\frac{\boldsymbol{u}^T}{\alpha} \hat{\boldsymbol{v}}_g \boldsymbol{x} - \boldsymbol{x}^T \boldsymbol{s}_g \boldsymbol{x} = 0, \tag{7}$$

where $\boldsymbol{s}_g = \frac{1}{2}(\hat{\boldsymbol{v}}_g\hat{\boldsymbol{w}}_g + \hat{\boldsymbol{w}}_g\hat{\boldsymbol{v}}_g)$, $\boldsymbol{v}_g = (\boldsymbol{p}_{i+1} - \boldsymbol{p}_i)$ and $\boldsymbol{w}_g = (\boldsymbol{r}_{i+1} - \boldsymbol{r}_i)$. $\boldsymbol{v}_g$ and $\boldsymbol{w}_g$ describe the relative pose between the first scanlines of two rolling shutter frames, and can be taken to be the same as $\boldsymbol{v}$ and $\boldsymbol{w}$ from the global shutter case. It can be seen from Eq. (7) that our differential epipolar constraint for rolling shutter cameras differs

[1] Note that our version is slightly different from [16] by the sign in the second term due to the difference on how we define the motion.

from the differential epipolar constraint for global shutter cameras (Eq. (3)) by just the scaling factor α on the optical flow vector $\boldsymbol{u}$. Here, we can make the interpretation that the rolling shutter optical flow vector $\boldsymbol{u}$ when scaled by α is equivalent to the global shutter optical flow vector. Collecting all optical flow vectors, and rectifying each of them with its own α (dependent on the scanlines involved in the optical flow), we can now solve for the RS relative motion using conventional linear 8-point algorithm [16].

4.2. Constant Acceleration Motion

Despite the simplicity of compensating for the RS effect by scaling the measured optical flow vector, the constant velocity assumption can be too restrictive for real image sequences captured by a moving RS camera. To enhance the generality of our model, we relax the constant velocity assumption to the more realistic constant acceleration motion. More specifically, we assume constant direction of translational and rotational velocity, but allow its magnitude to either increase or decrease gradually. Experimental results on real data show that this relaxation on motion assumption improves the performance significantly.

The constant acceleration model slightly complicates the interpolation for the pose of each scanline, compared to the constant velocity model. We show only the derivations for the translation of the scanlines since similar derivations apply to the rotation. Suppose the initial translational velocity of the camera at $\boldsymbol{p}_i$ is $\boldsymbol{\mathcal{V}}$ and it maintains a constant acceleration $\boldsymbol{a}$ such that at time t the velocity increases or decreases to $\boldsymbol{\mathcal{V}} + \boldsymbol{a}t$, and the translation $\boldsymbol{p}(t)$ is

$$\boldsymbol{p}(t) = \boldsymbol{p}_i + \int_0^t (\boldsymbol{\mathcal{V}} + \boldsymbol{a}t')dt' = \boldsymbol{p}_i + \boldsymbol{\mathcal{V}}t + \frac{1}{2}\boldsymbol{a}t^2. \quad (8)$$

Let us re-parameterize $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{a}$ as $\boldsymbol{\mathcal{V}} = \frac{\Delta \boldsymbol{p}}{\Delta t}$ and $\boldsymbol{a} = k\frac{\boldsymbol{\mathcal{V}}}{\Delta t}$, where $\Delta\boldsymbol{p}$ is an auxiliary variable introduced to represent a translation, Δt is the time period between two first scanlines, and k is a scalar factor that needs to be estimated. Putting $\boldsymbol{\mathcal{V}}$, $\boldsymbol{a}$ back into Eq. (8) and let $t = \Delta t$, we get the translation for the first scanline of frame $i+1$ as

$$\boldsymbol{p}_{i+1} = \boldsymbol{p}_i + (1 + \frac{1}{2}k)\Delta\boldsymbol{p}. \quad (9)$$

Denoting the time stamp of scanline y_1 (or y_2) on image i (or $i+1$) by t_{y_1} (or t_{y_2}), we have

$$t_{y_1} = \frac{\gamma y_1}{h}\Delta t, \quad t_{y_2} = (1 + \frac{\gamma y_2}{h})\Delta t. \quad (10)$$

Substituting the two time instances in Eq. (10) into Eq. (8) and eliminating $\Delta\boldsymbol{p}$ by Eq. (9) gives rise to the translations of scanline y_1 and y_2:

$$\begin{aligned}\boldsymbol{p}_1 &= \boldsymbol{p}_i + \frac{\gamma y_1}{h}\Delta\boldsymbol{p} + \frac{1}{2}k(\frac{\gamma y_1}{h})^2\Delta\boldsymbol{p} \quad (11a)\\ &= \boldsymbol{p}_i + \underbrace{(\frac{\gamma y_1}{h} + \frac{1}{2}k(\frac{\gamma y_1}{h})^2)(\frac{2}{2+k})}_{\beta_1(k)}(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i),\end{aligned}$$

$$\begin{aligned}\boldsymbol{p}_2 &= \boldsymbol{p}_i + (1 + \frac{\gamma y_2}{h})\Delta\boldsymbol{p} + \frac{1}{2}k(1 + \frac{\gamma y_2}{h})^2\Delta\boldsymbol{p} \quad (11b)\\ &= \boldsymbol{p}_i + \underbrace{(1 + \frac{\gamma y_2}{h} + \frac{1}{2}k(1 + \frac{\gamma y_2}{h})^2)(\frac{2}{2+k})}_{\beta_2(k)}(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i).\end{aligned}$$

Similar to Eq. (6), we get the relative translation and rotation between scanline y_2 and y_1 as follows:

$$\boldsymbol{p}_{21} = \beta(k)(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i), \quad \boldsymbol{r}_{21} = \beta(k)(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i), \quad (12)$$

where $\beta(k) = \beta_2(k) - \beta_1(k)$. Making use of the small motion assumption, we plug $(\boldsymbol{p}_{21}, \boldsymbol{r}_{21})$ into Eq. (3) and the RS differential epipolar constraint can now be written as

$$\boldsymbol{u}^T\hat{\boldsymbol{v}}_g\boldsymbol{x} - \beta(k)\boldsymbol{x}^T\boldsymbol{s}_g\boldsymbol{x} = 0. \quad (13)$$

It is easy to verify that Eq. (13) reduces to Eq. (7) when the acceleration vanishes , i.e. $k = 0$ (constant velocity).

In comparison to the constant velocity model, we have one additional unknown motion parameter k to be estimated, making Eq. (13) a polynomial equation. In what follows, we show that Eq. (13) can be solved by a 9-point algorithm with the hidden variable resultant method [4]. Rewriting $\boldsymbol{v}_g$ as $[v_x, v_y, v_z]^T$ and the symmetrical matrix $\boldsymbol{s}_g$ as

$$\boldsymbol{s}_g = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_2 & s_4 & s_5 \\ s_3 & s_5 & s_6 \end{bmatrix},$$

Eq. (13) can be rearranged to

$$z(k)\boldsymbol{e} = 0, \quad (14)$$

where $z(k)$ is a 1×9 vector made up of the known variables $\gamma, h, \boldsymbol{x}$ and $\boldsymbol{u}$, and the unknown variable k. $\boldsymbol{e}$ is a 9×1 unknown vector as follows:

$$\mathbf{e} = [v_x,\ v_y,\ v_z,\ s_1,\ s_2,\ s_3,\ s_4,\ s_5,\ s_6]^T. \quad (15)$$

We need 9 image position and optical flow vectors $(\boldsymbol{x}, \boldsymbol{u})$ to determine the 10 unknown variables k and $\boldsymbol{e}$ up to a scale. Each point yields one constraint in the form of Eq. (14). Collecting these constraints from all the points, we get a polynomial system:

$$\boldsymbol{\mathcal{Z}}(k)\boldsymbol{e} = 0, \quad (16)$$

where $\boldsymbol{\mathcal{Z}}(k) = [z_1(k)^T, z_2(k)^T, ..., z_9(k)^T]^T$ is a 9×9 matrix. For Eq. (16) to have a non-trivial solution, $\boldsymbol{\mathcal{Z}}(k)$ must be rank-deficient which implies a vanishing determinant:

$$det(\boldsymbol{\mathcal{Z}}(k)) = 0. \quad (17)$$

Eq. (17) yields a 6-degree univariate polynomial in terms of the unknown k which can be solved by the technique of Companion matrix [4] or Sturm bracketing [19]. Next, the Singular Value Decomposition (SVD) is applied to $\boldsymbol{\mathcal{Z}}(k)$, and the singular vector associated with the least singular value is taken to be $\boldsymbol{e}$. Following [16], we extract $(\boldsymbol{v}_g, \boldsymbol{w}_g)$ from $\boldsymbol{e}$ by a projection onto the symmetric epipolar space. The minimal solver takes less than 0.02s using our unoptimized MATLAB code.

4.3. RS-Aware Non-Linear Refinement

It is clear that the above algorithm minimizes the algebraic errors and thus yields a biased solution. To obtain more accurate solution, this should be followed by one more step of non-linear refinement that minimizes the geometric errors. In the same spirit of re-projection error in the discrete case and combining Eq. (1) and (12), we write the differential re-projection error and non-linear refinement as

$$\underset{k,\boldsymbol{v}_g,\boldsymbol{w}_g,\boldsymbol{Z}}{\operatorname{argmin}} \sum_{i\in O}^{N} ||\boldsymbol{u}_i - \beta^i(k)(\frac{\boldsymbol{A}_i\boldsymbol{v}_g}{Z_i} + \boldsymbol{B}_i\boldsymbol{w}_g)||_2^2, \quad (18)$$

which minimizes the errors between the measured and predicted optical flows for all points in the pixel set O over the estimated parameters k, $\boldsymbol{v}_g$, $\boldsymbol{w}_g$, $\boldsymbol{Z} = \{Z_1, Z_2, ...Z_N\}$. $\boldsymbol{Z}$ is the depths associated with all the image points in O. N is the total number of points. Note that in the case of constant velocity model, k is kept fixed as zero in this step. Also note that (18) reduces to the traditional non-linear refinement for GS model [3, 28, 12] when the readout time ratio γ is set as 0. RANSAC is used to obtain a robust initial estimate. For each RANSAC iteration, we apply our minimal solver to obtain k, $\boldsymbol{v}_g$, $\boldsymbol{w}_g$ and then compute the optimal depth for each pixel by minimizing (18) over $\boldsymbol{Z}$; the inlier set is identified by checking the resultant differential re-projection error on each pixel. The threshold is set as 0.001 on the normalized image plane for all experiments. We then minimize (18) for all points in the largest inlier set from RANSAC to improve the initial estimates by block coordinate descent over k, $\boldsymbol{v}_g$, $\boldsymbol{w}_g$ and $\boldsymbol{Z}$, whereby each subproblem block admits a closed-form solution. Finally, $\boldsymbol{Z}$ is recovered for all pixels which gives the dense depth map.

5. RS-Aware Warping For Image Rectification

Having obtained the camera pose for each scanline and the depth map of the first RS image frame, a natural extension is to take advantage of these information to rectify the image distortion caused by the RS effect. From Eq. (11a) we know that the relative poses $(\boldsymbol{p}_{1i}, \boldsymbol{r}_{1i})$ between the first and other scanlines in the same image are as follow:

$$\boldsymbol{p}_{1i} = \boldsymbol{p_1} - \boldsymbol{p_i} = \beta_1(k)\boldsymbol{v}_g, \quad (19a)$$

$$\boldsymbol{r}_{1i} = \boldsymbol{r_1} - \boldsymbol{r_i} = \beta_1(k)\boldsymbol{w}_g. \quad (19b)$$

Combining the pose of each scanline with the depth map, warping can be done by back-projecting each pixel on each scanline into the 3D space, which gives the point cloud, followed by a projection onto the image plane that corresponds to the first scanline. Alternatively, the warping displacement can be computed from Eq. (1) by small motion approximation as $\boldsymbol{u}_w = \beta_1(k)(\frac{\boldsymbol{A}\boldsymbol{v}_g}{Z} + \boldsymbol{B}\boldsymbol{w}_g)$.

Since the camera positions of each scanline within the same image are fairly close to that of the first scanline, the displacement caused by the warping is small compared to the optical flow between two consecutive frames. Thus warping-induced gaps are negligible and we do not need to use any pixel from the next frame (i.e. image $i+1$) for the rectification. This in turn means that the warping introduces no ghosting artifacts caused by misalignment, allowing the resulting image to retain the sharpness of the original image while removing the geometric RS distortion, as shown by the experimental results in Sec. 6.

6. Experiments

In this section, we show the experimental results of our proposed algorithm on both synthetic and real image data.

6.1. Synthetic Data

We generate synthetic but realistic RS images by making use of two textured 3D mesh—the *Old Town* and *Castle* provided by [21] for our experiments. To simulate the RS effect, we first use the 3D Renderer software [10] to render the GS images according to the pose of scanlines. From these GS images, we extract the pixel values for each scanline to generate the RS images. As such, we can fully control all the ground truth camera and motion parameters including the readout time ratio γ_G, camera relative translation $\boldsymbol{v}_G$ and rotation $\boldsymbol{R}_G = \exp(\boldsymbol{w}_G)$, and acceleration parameter k_G in the case of constant acceleration motion. The image size is set as 900×900 with a 810 pixels focal length. Examples of the rendered RS images from both datasets are shown in the first row of Fig. 3. For the relative motion estimate $(\boldsymbol{v}_g, \boldsymbol{w}_g)$, we measure the translational error as $\cos^{-1}(\boldsymbol{v}_g^T\boldsymbol{v}_G/(\|\boldsymbol{v}_g\|\,\|\boldsymbol{v}_G\|))$ and the rotational error as the norm of the Euler angles from $\boldsymbol{R}_g\boldsymbol{R}_G^T$, where $\boldsymbol{R}_g = \exp(\boldsymbol{w}_g)$. Since the translation is ambiguous in its magnitude, the amount of translation is always represented as the ratio between the absolute translation magnitude and average scene depth in the rest of this paper. We term this ratio as normalized translation.

Quantitative Evaluation: We compare the accuracy of our RS-aware motion estimation to the conventional GS-based model. We avoid forward motion which is well-known to be challenging for SfM even for traditional GS cameras. WLOG, all the motions that we synthesize have equal vertical and horizontal translation components, and equal yaw, pitch and roll rotation components. To fully understand the behavior of our proposed algorithm, we investigate the performance under various settings. To get statistically meaningful result, all the errors are obtained from an average of 100 trials, each with 300 iterations of RANSAC. Both the results from the minimal solver and the non-linear refinement are reported to study their respective contribution to the performance.

We plot the translational and rotational error under the constant velocity motion in Fig. 5. We first investigate how

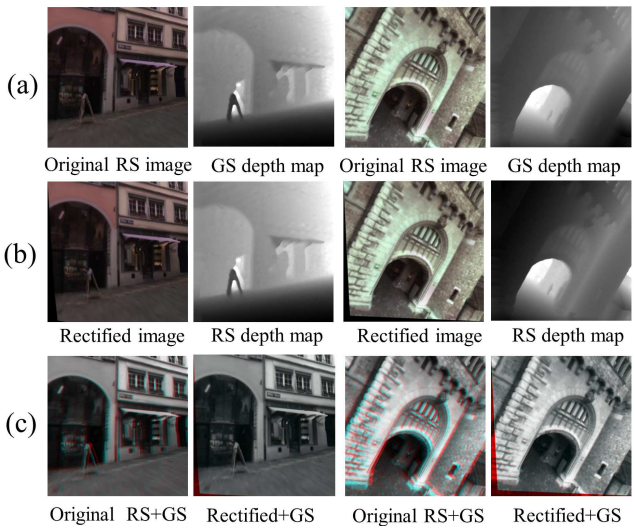

Figure 3: An example of the experimental results on the *Old Town* and *Castle* data. (a)-(b): The original RS images, estimated depth maps by GS & RS, and rectified images. (c) Overlaying the original RS and the rectified images on the ground truth GS images.

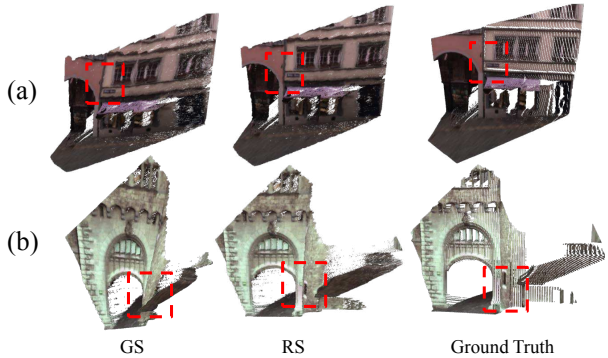

Figure 4: Visualization of the reconstructed 3D point clouds.

the value of the readout time ratio γ would affect the performance in Fig. 5(a)-(b) by increasing γ from 0.1 to 1, while the normalized translation and magnitude of $\boldsymbol{w}$ are fixed at 0.025 and 3° respectively. We can see that the accuracy of the RS model (both minimal solver and non-linear refinement) is insensitive to the variation of γ, while the GS model tends to give higher errors with increasing γ. This result is expected because a larger readout time ratio leads to larger RS distortion in the image. Next, we fixed the value of γ to 0.8 for the following two settings: (1) We fix the magnitude of $\boldsymbol{w}$ to 3° and increase the normalized translation from 0.02 to 0.06 as shown in Fig. 5(c)-(d). (2) The normalized translation is fixed as 0.025 and the magnitude of $\boldsymbol{w}$ is increased from 0.5° to 4.5° as shown in Fig. 5(e)-(f). Overall, the accuracies of the RS and GS model have a common trend determined by the type of motion. However, the RS model has higher accuracies in general, especially in the challenging cases where the rotation is relatively large compared to translation. This implies that our RS-aware algorithm has compensated for the RS effect in pose estimation. We note that in some cases, especially in *Old Town*, the non-linear refinement gives marginal improvement or even increases the error slightly. We reckon this is because the RANSAC criterion we used is exactly the individual term that forms the non-linear cost function, and it can happen that the initial solution is already close to a local minimum, hence the effect of non-linear refinement can become dubious given that Eq.(1) is only an approximation for small discrete motion in practice, as mentioned in Sec.3.

Similarly, we conduct quantitative evaluations under the constant acceleration motion. To save space, only the results from *Old Town* are reported here. See *supplementary material* for the similar results from *Castle*. First, we investigate how the variation of acceleration by increasing k from -0.2 to 0.2 would influence the performance of both the GS and RS model in Fig. 6(a). We can see that the accuracy of the GS model degrades dramatically under large acceleration, while the RS model maintains almost consistent accuracies regardless of the amount of acceleration. For Fig. 6(b)-(d), we fix k to 0.1 and set other motion or camera parameters to be the same as that for the constant velocity motion. As can be observed, the RS model in general yields higher accuracies than the GS model, especially for the translation. For example, the GS model gives significantly larger error ($> 50^\circ$) on translation under strong rotation as shown in Fig. 6(d). We observe that the non-linear refinement tends to improve the translation estimate but degrade the rotation estimate for the GS model. For the RS model, the impact is marginal. We observe larger improvement when the RANSAC threshold is increased, but this leads to a drop of overall accuracy. See our *supplementary material* for more analyses on the quantitative results.

For qualitative results, two examples under constant acceleration motion are shown in Fig.3&4. Fig. 3(a)&(b) show the original synthetic RS images, the estimated depth maps using the GS model, and the estimated depth maps and rectified images using our RS model. In Fig. 3(c), we compare the original RS images and rectified images to the ground truth GS images, which are rendered according to the poses of the first scanlines, via overlaying. The red and blue color regions indicate high differences. Compared to the original RS images, one can see that the rectified images from our RS-aware warping are closer to the ground truth GS images, except in the few regions near the image edges where the optical flow computation may not be reliable. In Fig. 4, we show the point clouds reconstructed by the GS model, our RS model, and the ground truth respectively. As highlighted by the boxes, the point clouds returned by the GS model are distorted compared to the ground truth. In comparison, our RS model successfully rectifies these artifacts to obtain visually more appealing results.

6.2. Real data

In this section, we show the results of applying the proposed RS algorithm to images collected by real RS cameras.

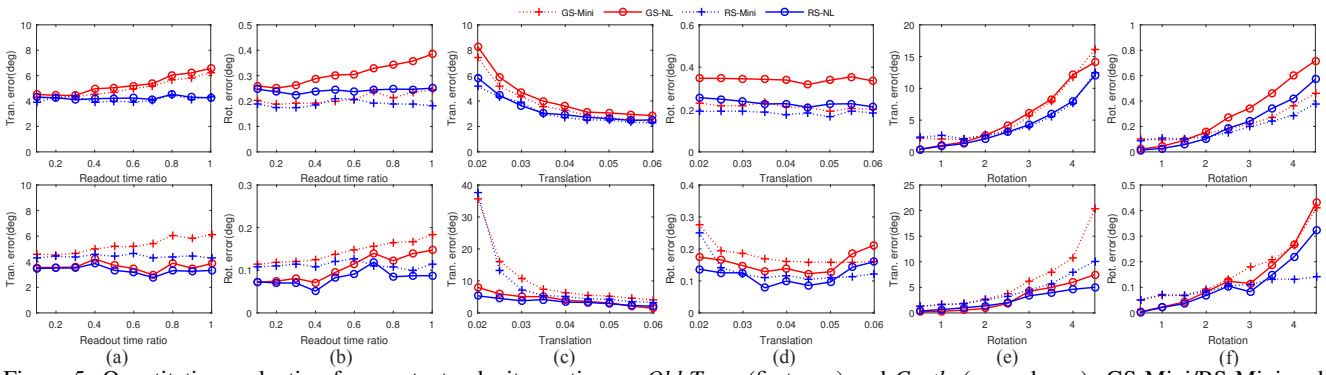

Figure 5: Quantitative evaluation for constant velocity motion on *Old Town* (first row) and *Castle* (second row). GS-Mini/RS-Mini and GS-NL/RS-NL stand for the results from the minimal solver and non-linear refinement respectively using GS/RS model.

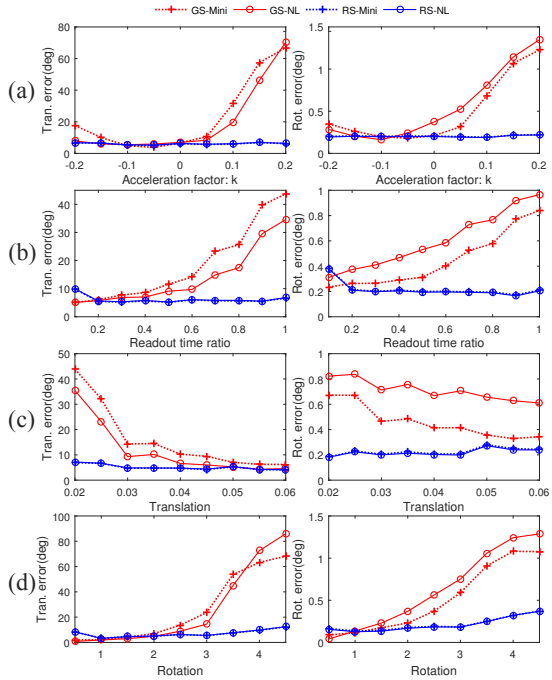

Figure 6: Quantitative evaluation for constant acceleration motion on *Old Town*. The legend is the same as in Fig.5.

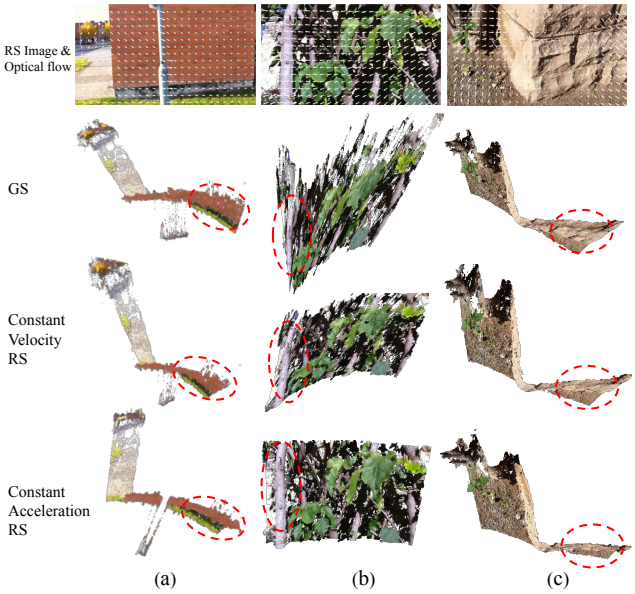

Figure 7: SfM results on real image data. Top row: original RS images. Bottom 3 rows: reconstructed 3D point clouds by the GS model and our RS models with constant velocity and acceleration.

First, we show the results on pairs of consecutive images from the public RS images dataset released by [11]. The sequence was collected by an Iphone 4 camera at 1280×720 resolution with 96% readout time ratio. Despite having a GS camera that is rigidly mounted near the Iphone for ground truth comparison over long trajectories as shown in [11], the accuracy is insufficient for the images from the GS camera to be used as ground truth for two-frame differential relative pose estimation. Instead, we rely on the visual quality of the reconstructed point clouds to evaluate our algorithms. We show the point clouds of three different scenes by the GS and our RS models—both constant velocity and acceleration in Fig. 7. More results are shown in *supplementary material*. As highlighted by the red ellipses in Fig. 7(a), we can see from the top-down view that the wall is significantly skewed under the GS model. This distortion is corrected to a certain extent and almost completely removed by our constant velocity and acceleration RS models respectively. Similar performance of our RS models can also be observed in the examples shown in Fig. 7(b) and Fig. 7(c) from front and top-down view respectively.

The RS effect from the above mentioned dataset is significant enough to introduce bias in SfM algorithm, but it is not strong enough to generate noticeable image distortions. To demonstrate our image rectification algorithm, we collected a few image sequences with an Iphone 4 camera under larger motions that lead to obvious RS distortions on the images. We compare the results of our proposed method with those of Rolling Shutter Repair in two image/video editing software products—Adobe After Effect and Apple Imovie on pairs of the collected images, as shown in Fig. 8. We feed the image sequences along with camera parameters into the software products. We tried different advanced settings provided by After Effect to get the best result for

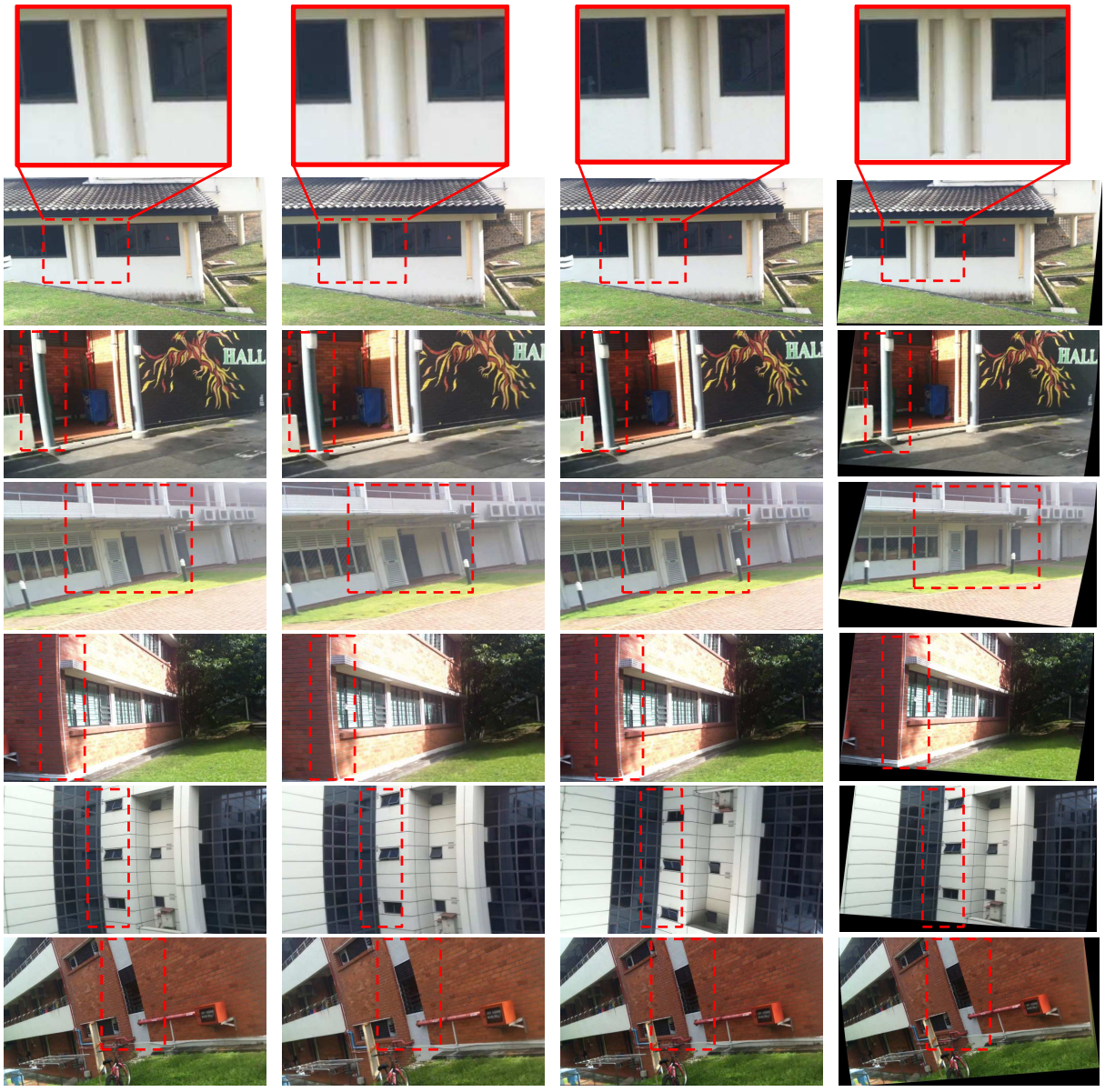

Figure 8: Comparison of image rectification results on real image data with noticeable RS distortion. The red boxes highlight the superior performance of our proposed method.

each scene. Since we observe that both our RS models with constant velocity or acceleration give similar results, we only report the rectified images using the accelerated motion model. It can be seen that our method works consistently better than the two commercial software products in removing the RS artifacts such as skew and wobble in the images (highlighted by the red boxes). For example, the slanted window on the original RS image shown on the top row of Fig. 8 becomes most close to vertical in our result.

7. Conclusion

In this paper, we proposed two tractable algorithms to correct the inaccuracies in differential SfM caused by the RS effect in images collected from a RS camera moving under constant velocity and acceleration respectively. In addition, we proposed the use of a RS-aware warping for image rectification that removes the RS distortion on images. Quantitative and qualitative experimental results on both synthetic and real RS images demonstrated the effectiveness of our algorithm.

Acknowledgements. This work was partially supported by the Singapore PSF grant 1521200082 and Singapore MOE Tier 1 grant R-252-000-636-133.

References

[1] C. Albl, Z. Kukelova, and T. Pajdla. R6p-rolling shutter absolute camera pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2292–2300, 2015.

[2] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2392–2399. IEEE, 2010.

[3] A. R. Bruss and B. K. Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21(1):3–20, 1983.

[4] D. Cox, J. Little, and D. OShea. Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra, 2007.

[5] Y. Dai, H. Li, and L. Kneip. Rolling shutter camera relative pose: Generalized epipolar geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *European Conference on Computer Vision (ECCV)*, 2010.

[7] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Calibration-free rolling shutter removal. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012.

[8] G. Hanning, N. Forslöw, P.-E. Forssén, E. Ringaby, D. Törnqvist, and J. Callmer. Stabilizing cell phone video using inertial measurement sensors. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1–8. IEEE, 2011.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[10] T. Hassner. Viewing real-world faces in 3d. In *International Conference on Computer Vision (ICCV)*, pages 3607–3614, 2013.

[11] J. Hedborg, P.-E. Forssen, M. Felsberg, and E. Ringaby. Rolling shutter bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1434–1441, 2012.

[12] C. Hu and L. F. Cheong. Linear quasi-parallax sfm using laterally-placed eyes. *International journal of computer vision*, 84(1):21–39, 2009.

[13] S. Im, H. Ha, G. Choe, H.-G. Jeon, K. Joo, and I. S. Kweon. High quality structure from small motion for rolling shutter cameras. In *International Conference on Computer Vision (ICCV)*, pages 837–845, 2015.

[14] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR*, 1:2, 2011.

[15] B. Klingner, D. Martin, and J. Roseborough. Street view motion-from-structure-from-motion. In *International Conference on Computer Vision (ICCV)*, 2013.

[16] Y. Ma, J. Košecká, and S. Sastry. Linear differential algorithm for motion recovery: A geometric approach. *International Journal of Computer Vision*, 36(1):71–89, 2000.

[17] L. Magerand, A. Bartoli, O. Ait-Aider, and D. Pizarro. Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching. In *European Conference on Computer Vision*, pages 456–469. Springer, 2012.

[18] M. Meingast, C. Geyer, and S. Sastry. Geometric models of rolling-shutter cameras. In *Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, 2005.

[19] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[20] E. Ringaby and P.-E. Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision*, 96(3):335–352, 2012.

[21] O. Saurer, K. Koser, J.-Y. Bouguet, and M. Pollefeys. Rolling shutter stereo. In *International Conference on Computer Vision (ICCV)*, pages 465–472, 2013.

[22] O. Saurer, M. Pollefeys, and G. H. Lee. A minimal solution to the rolling shutter pose estimation problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

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

[24] H. Stewénius, C. Engels, and D. Nistér. An efficient minimal solution for infinitesimal camera motion. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[25] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *International Conference on Computer Vision (ICCV)*, pages 1385–1392, 2013.

[26] C. Wu. Visualsfm: A visual structure from motion system. `http://ccwu.me/vsfm/index.html`, 2014.

[27] M. Zucchelli. *Optical flow based structure from motion*. Citeseer, 2002.

[28] M. Zucchelli, J. Santos-Victor, and H. I. Christensen. Maximum likelihood structure and motion estimation integrated over time. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 260–263. IEEE, 2002.