A Minimal Solution to the Rolling Shutter Pose Estimation Problem

Olivier Saurer[1], Marc Pollefeys[1], and Gim Hee Lee[2]

[1]Computer Vision and Geometry Lab, Department of Computer Science, ETH Zürich, Switzerland
[2]Department of Mechanical Engineering, National University of Singapore
{saurero, marc.pollefeys}@inf.ethz.ch, mpelgh@nus.edu.sg

***Abstract*— Artefacts that are present in images taken from a moving rolling shutter camera degrade the accuracy of absolute pose estimation. To alleviate this problem, we introduce an addition linear velocity in the camera projection matrix to approximate the motion of the rolling shutter camera. In particular, we derive a minimal solution using the Gröbner Basis that solves for the absolute pose as well as the motion of a rolling shutter camera. We show that the minimal problem requires 5-point correspondences and gives up to 8 real solutions. We also show that our formulation can be extended to use more than 5-point correspondences. We use RANSAC to robustly get all the inliers. In the final step, we relax the linear velocity assumption and do a non-linear refinement on the full motion, i.e. linear and angular velocities, and pose of the rolling shutter camera with all the inliers. We verify the feasibility and accuracy of our algorithm with both simulated and real-world datasets.**

I. Introduction

Absolute pose estimation in Computer Vision refers to the problem of finding the camera pose in the world frame given a set of 3D scene points expressed in the world frame and corresponding set of 2D image points expressed in the camera coordinate frame. A minimum of three 2D-3D correspondences are needed to solve for the absolute pose in the case of a global shutter camera. This is commonly referred to as the Perspective-3-Point or P3P problem [1]. A generalization of the P3P problem to n-point correspondences is known as the PnP problem. The solution to the absolute pose estimation problem has great importance in performing robotics visual Simultaneous Localization and Mapping (visual SLAM), localization with respect to a given map, and Structure-from-Motion (SfM).

Over the years, a huge literature of solutions to the absolute pose estimation problem [1], [2], [3], [4] have been developed by many researchers for the global shutter camera. The solutions to the absolute pose estimation problem for global shutter cameras, however, do not work equally well for a moving rolling shutter camera. This is because the existing solutions are modeled for global shutter cameras that take snapshots of a scene by exposing the entire photo-sensor in a single instance of time. These solutions do not account for the image artifacts caused by a moving rolling shutter camera that simultaneously exposes scanlines of its photo-sensor either horizontally or vertically over a rapid instance of time ($\sim$ 72ms in our real data experiments).

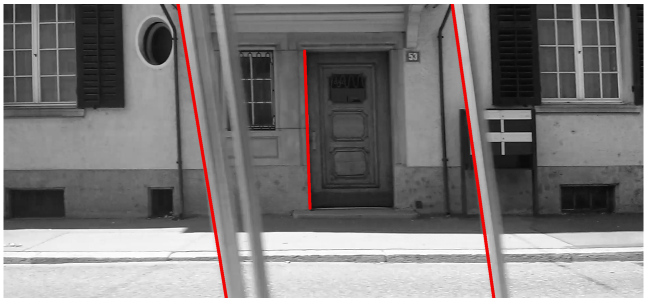


Fig. 1. Artifacts on an image taken with a rolling shutter camera moving towards the left of the image. The poles and and door frame (marked in red), which are supposed to be upright, appear slanted to the left due to the camera motion and the sequential exposure of each scanline from top to bottom of the image. Objects further away from the camera are less affected by the rolling shutter.

Figure 1 shows an example of an image taken by a moving rolling shutter camera. The camera moves towards the left of the image and the scanlines progress from top to bottom of the image. As a result, scene objects such as the fence and building facade edge (marked in red) appear to be slanted to the left of the image.

While there is an inherent difficulty in doing absolute pose estimation with rolling shutter cameras, there is already a widespread usage of the rolling shutter cameras due to the low cost in manufacturing and robustness of the CMOS photo sensors, and the massive incorporation of the cameras into mobile devices such as mobile phones and tablets. It is therefore useful to provide an algorithm that corrects for the moving rolling shutter camera artifacts while doing absolute pose estimation.

In this paper, we propose a minimal solution to the rolling shutter camera pose estimation problem. In particular, we introduce an additional linear velocity in the camera projection matrix to model the motion of the rolling shutter camera. We noted that this assumption holds in practice because the scanline speed is always much faster than the velocity of the rolling shutter camera mounted on a hand-held mobile phone, tablet or a moving car. We show that a minimum of 5-point 2D-3D correspondences are needed to solve for the pose and linear velocity using the Gröbner basis [5] and gives up to eight real solutions. We also show that our formulation can be extended to use more than 5-point correspondences. We use RANSAC [6] for robust estimation

A large part of this paper was done when the last author was at Mitsubishi Electric Research Laboratories (MERL), Cambridge Massachusetts, USA.

to get all the inlier point correspondences. We also identify the correct solution from the eight possible solutions within RANSAC. Finally, we relax the linear velocity assumption and do a non-linear refinement on the full motion, i.e. linear and angular velocities, and pose of the rolling shutter camera with all the inliers. We verify the feasibility and accuracy of our algorithm with both simulated and real-world datasets.

II. Related Work

Most of the existing works on rolling shutter cameras largely revolve around calibration, correction for rolling shutter distortion on the images, using rolling shutter cameras for in stereo setups, and iterative methods for pose estimation. In contrast, we propose a minimal solution to estimate the rolling shutter camera pose and velocity in this work. Our minimal solution requires only 5-point correspondences and this makes it very suitable to be used within RANSAC for robust estimation to find all the inlier correspondences.

One of the early publications on rolling shutter camera is from Liang *et al.* [7]. They gave detailed discussions on the rolling shutter effect and low level CMOS sensor that usually has an electronic rolling shutter. In this work, the authors proposed to compensate for the rolling shutter effect using optical flow. In [8], Geyer *et al.* proposed a method to calibrate the rolling shutter timings using additional hardware and studied the different rolling shutter effects under special fronto-parallel motion. More recently, Oth *et al.* proposed in [9] to calibrate the shutter timings using a video sequence of a known calibration pattern. A continuous-time trajectory model is combined with a rolling shutter model to estimate the shutter timings.

In [10], [11], the authors proposed 2D approaches for rolling shutter image stabilization and rolling shutter distortion correction using optical flow. Similarly, [12] used optical flow and a mixture of homographies to correct for the rolling shutter effect. In [13], [14], Hanning *et al.* and Karpenk *et al.* proposed rolling shutter distortion correction base on gyroscope measurements. Their assumption is that on hand-held devices the main motion during exposure is due to a rotation and can be compensated with a homography.

While the above approaches are 2D in nature, Forssen and Ringaby [15], [16] proposed a Structure-from-motion approach to compensate for the rolling shutter distortion that is mainly induced by rotational motions. In [17], Hedborg *et al.* proposed a full rolling shutter bundle adjustment on a continuous video stream by enforcing a continuous pose parametrization between consecutive frames. Klein *et .al* proposed in [18] to first estimate a constant velocity between consecutive frames and uses this motion model to undo the RS distortion on the extracted keypoints. The corrected keypoints are then used in a standard bundle adjustment [19] for global shutter camera.

In [20], [21], the authors have proposed stereo algorithms that take into account the rolling shutter model and produces geometrically consistent 3D reconstructions. In [22], Meilland *et al.* proposed a dense 3D model registration which accounts for rolling shutter distortion and motion blur on RGBD data.

Probably closest to our work is the work by Ait-Aider *et al.* [23], where they estimate the pose and velocity of a moving object from a single rolling shutter image. They use a spiral motion parametrization of the camera pose and solve for the pose and velocities as a non-linear least squares problem. Their formulation requires a good initialization which is obtained from a global shutter pose algorithm. The 3D-2D correspondences are provided manually. In [24], the authors extended the initialization process with a homography based formulation, which takes into account the temporal pose parametrization. It is however limited to only planar objects. To overcome the initialization burden, Magerand *et. al* [25] solved the pose and velocities using constrained global optimization by parameterizing the camera motion with degree 2 polynomials. The final objective function they need to solve for consists of a 6 degree polynomial with twelve unknowns.

III. Rolling Shutter Camera Pose Estimation

A. Camera Motion Model

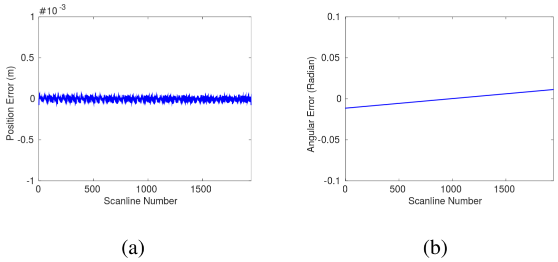
Since a rolling shutter camera typically has a rapid scanning time ($\sim$ 72ms per image), it is reasonable to make the assumption that the camera undergoes constant linear and angular velocities during an image acquisition. We further assume that each scanline takes exactly the same time, and the relative camera translation $\mathbf{t}_n$ and rotation $\mathbf{R}_n$ at the n^{th} scanline with respect to the first scanline can be linearly interpolated as

$$\mathbf{t}_n = \mathbf{v} n \tau \tag{1a}$$

$$\mathbf{R}_n = \mathbf{exp}(\mathbf{\Omega} n \tau), \tag{1b}$$

where $\mathbf{v} = [v_x, v_y, v_z]^\top$ and $\mathbf{\Omega} = [\omega_x, \omega_y, \omega_z]^\top$ denotes the constant linear and angular camera velocities, and τ is the time taken to complete each scanline. The function $\mathbf{exp}(.) : so(3) \rightarrow SO(3)$ denotes the exponential map that transforms the angle-axis rotation representation to a corresponding rotation matrix.

As mentioned in Section I that in practice the scanline speed is always much faster than the velocity of the camera, the camera motion can be approximated with only the linear velocity. As such, we consider only the linear velocity in our derivation of the minimal solution, i.e. $\mathbf{R}_n = \mathbf{exp}(0) = \mathbf{I}_{3\times3}$. We justify the validity of this assumption with the results from a real data experiment. We look at an image sequence where a car with a rolling shutter camera mounted on it takes a 90° turn, while driving at 10km/h. During the scan time of the CMOS sensor (72ms in our case), the car moved 0.2m and the absolute camera orientation changed by 0.02rad. Figure 2 compares the ground truth GPS/INS poses to the interpolated poses assuming zero angular velocity. The maximum absolute angular error obtained is only 0.01rad. We will further discuss the valid range of this assumption in Section IV-A.



(a) (b)

Fig. 2. The plots show the position and rotation error while the car moves through a 90° turn. During image scan time of 72ms the car moved 0.20m. (a) The maximum position error is 1.8718×10^{-4}m. (b) Maximum rotation error assuming zero angular velocity is 0.0114 rad.

B. Minimal 5-point Algorithm

Making the assumption of a constant linear velocity, we can express a pixel on the n^{th} scanline as

$$\mathbf{x}_n = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t} - \mathbf{t}_n\end{bmatrix}\mathbf{X}, \tag{2}$$

where $\mathbf{K}$ is the camera intrinsic. $\mathbf{R}$ and $\mathbf{t}$ are the camera pose in the world frame, which is also the camera pose for the first scanline. $\mathbf{t}_n$ is the camera pose for the n^{th} scanline as given in Equation 1. $\mathbf{x}_n \leftrightarrow \mathbf{X}$ is the 2D-3D point correspondence. Formally, Equation 2 is the camera projection equation that accounts for the rolling shutter effect. The unknowns are $\mathbf{R}$, $\mathbf{t}$ and the linear velocity $\mathbf{v}$ in $\mathbf{t}_n$, where there are altogether 9 degree-of-freedom (3 degree-of-freedom each for $\mathbf{R}$, $\mathbf{t}$ and $\mathbf{v}$). Since each point correspondence gives two independent equations, a minimum of 5-point correspondences are needed to solve for all the unknowns in Equation 2. Taking the cross product of $\mathbf{x}_n$ with Equation 2, we get

$$\mathbf{x}_n \times (\mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t} - \mathbf{t}_n\end{bmatrix}\mathbf{X}) = 0. \tag{3}$$

With 5-point correspondences, Equation 3 can be rearranged into the form

$$\mathbf{A}\mathbf{y} = \mathbf{0}, \tag{4}$$

where $\mathbf{A}$ is a matrix made up of the known values from the camera intrinsic $\mathbf{K}$, point correspondences $\mathbf{x}_n \leftrightarrow \mathbf{X}$, scanline number n and time τ. Here we choose randomly 9 out of the 10 equations in the minimal 5-point correspondence case (since any 9 out of the 10 equations are always independent) to form the 9×15 matrix $\mathbf{A}$.

$$\mathbf{y} = \begin{bmatrix}\mathbf{r_1} & \mathbf{r_2} & \mathbf{r_3} & t_x & t_y & t_z & v_x & v_y & v_z\end{bmatrix}^\top \tag{5}$$

is a 15×1 vector, where $\mathbf{r}_i$ is the i^{th} row of the rotation matrix $\mathbf{R}$, $[t_x \; t_y \; t_z]$ are the components from the translation vector $\mathbf{t}$ and $[v_x \; v_y \; v_z]$ are from the linear velocity $\mathbf{v}$ of the camera. Solving for the right nullspace of $\mathbf{Ay} = \mathbf{0}$ using the Singular Value Decomposition (SVD) gives 6 basis vectors denoted by $\mathbf{b}_1 ... \mathbf{b}_6$, where the linear combination forms the solution

$$\mathbf{y} = \beta_1\mathbf{b_1} + \beta_2\mathbf{b_2} + \beta_3\mathbf{b_3} + \beta_4\mathbf{b_4} + \beta_5\mathbf{b_5} + \beta_6\mathbf{b_6}. \tag{6}$$

$\beta_1 ... \beta_6$ are any scalar values. Assigning random values to $\beta_1 ... \beta_6$ however do not guarantee the orthogonality of $\mathbf{R}$. We fix $\beta_6 = 1$ and the remaining five scalar values can be found by enforcing the orthogonal constraint on the elements from the rotation matrix $\mathbf{R}$. Following [26], enforcing the orthogonality on $\mathbf{R}$ gives us 10 constraints:

$$||\mathbf{r_1}||^2 - ||\mathbf{r_2}||^2 = 0, \tag{7a}$$

$$||\mathbf{r_1}||^2 - ||\mathbf{r_3}||^2 = 0, \tag{7b}$$

$$||\mathbf{c_1}||^2 - ||\mathbf{c_2}||^2 = 0, \tag{7c}$$

$$||\mathbf{c_1}||^2 - ||\mathbf{c_3}||^2 = 0, \tag{7d}$$

$$\mathbf{r_1}^\top\mathbf{r_2} = 0, \; \mathbf{r_1}^\top\mathbf{r_3} = 0, \; \mathbf{r_2}^\top\mathbf{r_3} = 0, \tag{7e}$$

$$\mathbf{c_1}^\top\mathbf{c_2} = 0, \; \mathbf{c_1}^\top\mathbf{c_3} = 0, \; \mathbf{c_2}^\top\mathbf{c_3} = 0, \tag{7f}$$

which we can use to solve for the scalar values $\beta_1 ... \beta_5$ that formed the solution. $\mathbf{r}_i$ denotes the i^{th} row and $\mathbf{c}_i$ the i^{th} column of $\mathbf{R}$. Putting the elements from $\mathbf{R}$ in Equation 6 into the 10 constraints, we get a system of 10 polynomial equations with $\beta_1 ... \beta_5$ as the unknowns. We use the automatic generator of Gröbner solvers provided by Kukelova $et \; al.$ [27] to generate a solver for the system of polynomial equations that gives up to eight real solutions for $\mathbf{y}$. We divide each of the solution by its respective $||\mathbf{r}_1||$ to make $\mathbf{R}$ an orthonormal matrix. Note that the orthonormal constraint is not enforced earlier in Equation 7 to keep the system of polynomials less complicated. In addition, we ensure that the solution follows a right-hand coordinate system by negating the solution if det($\mathbf{R}$) $= -1$.

It should be noted that our formulation also works for m-point correspondences where $m \geq 5$, i.e. ≥ 10 independent equations are used to form Equation 4. In this case, we get an over-determinate system. The 6 basis vector in Equation 6 can be obtained from the 6 singular vectors that correspond to the 6 smallest singular values of $\mathbf{A}$.

C. Robust Estimation

We use the minimal 5-point algorithm within RANSAC [6] to robustly select all the inlier 2D-3D correspondences. We also determine the correct solution from the 8 solutions within each RANSAC loop as the one that gives the most inlier count.

D. Non-linear Refinement

The 5-point minimal solver with RANSAC provides an initial solution and finds all the inlier 2D-3D correspondences, which is then used for further refinement using a non-linear solver [28]. Here, we relaxed the linear velocity assumption and do a refinement on the full camera motion, i.e. linear $\mathbf{v}$ and angular $\mathbf{\Omega}$ velocities, and pose ($\mathbf{R}$, $\mathbf{t}$). Formally, we seek to minimize the total reprojection errors over $\mathbf{v}$, $\mathbf{\Omega}$, $\mathbf{R}$ and $\mathbf{t}$. The objective function is given by

$$\underset{\mathbf{v},\mathbf{\Omega},\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum_i ||\mathbf{x}_{n,i} - \pi(\mathbf{P}_{n,i}, \mathbf{X}_i)||^2, \tag{8}$$

where $\mathbf{x}_{n,i}$ is the i^{th} 2D image point on the n^{th} scanline, $\mathbf{X}_i$ is the corresponding 3D point, $\pi(.) = \mathbf{P}_{n,i}\mathbf{X}_i$ is the reprojection function. $\mathbf{P}_{n,i}$ is the rolling shutter camera projection matrix for the n^{th} scanline given by

$$\mathbf{P}_{n,i} = \mathbf{K}\mathbf{R}_n \begin{bmatrix} \mathbf{R} & \mathbf{t} - \mathbf{t}_n \end{bmatrix}, \tag{9}$$

where ($\mathbf{t}_n$, $\mathbf{R}_n$) is the pose of the rolling shutter camera when the n^{th} scanline was taken as defined in Equation 1.

IV. Evaluation

A. Synthetic Data

We first evaluate our proposed algorithm on several synthetic configurations. Specifically, we do comparisons for the following three methods:

1) **GS + refinement**: Global shutter P3P [4] with nonlinear rolling shutter aware refinement of pose and velocities as described in Section III-D. Note that in a perspective configuration the generalized P3P algorithm [4] simplifies to [29], its perspective counter part.
2) **RS**: Our proposed minimal 5-point rolling shutter pose and translational velocity solver as described in Section III-B.
3) **RS + refinement**: Our proposed minimal 5-point rolling shutter pose and translational velocity solver with refined pose and velocities as described in Section III-D.

The evaluations are done under varying image noise, increasing translational and angular velocities, and different shutter directions relative to the camera motions. There are a total of four different combinations for the shutter directions relative to the camera motions:

1) Horizontal shutter and sideway camera motion.
2) Horizontal shutter and forward camera motion.
3) Vertical shutter and sideway camera motion.
4) Vertical shutter and forward camera motion.

For each setting and method, we report the *median* error over 1000 random trials. Each trial consists of a random camera pose generated within the range of [0,1]m and [-0.01,0.01]rad for the respective axes, and the scene consists of 1000 randomly generated points with an average depth of 20m. We used an image resolution of 1000 pixels with a fixed focal length of 1000 pixels, which results in a field-of-view of about 53°. We assume a fixed rolling shutter scan time of 72ms for all experiments. The following error measure which averages the rotation and translation errors over all scanlines are used to evaluate the synthetic experiments:

- Angle difference in $\mathbf{R}$, averaged over all scanlines:

$$\delta_\theta = \frac{1}{N}\sum_{n}^{N} \cos^{-1}\left(\frac{Tr(\mathbf{R}_n\tilde{\mathbf{R}}_n^\top) - 1}{2}\right), \tag{10}$$

- Translation difference, averaged over all scanlines:

$$\delta_t = \frac{1}{N}\sum_{n}^{N} ||\mathbf{t}_n - \tilde{\mathbf{t}}_n||, \tag{11}$$

where $\mathbf{R}_n$, $\mathbf{t}_n$ denote the ground-truth transformation for a given scanline n and $\tilde{\mathbf{R}}_n$, $\tilde{\mathbf{t}}_n$ are the corresponding estimated measurements and N represents the total number of scanlines.

Figure 3 shows the average error plots from the three algorithms under increasing translational velocity, zero angular velocity and an image noise of 0.5 pixel standard deviation. It can be seen that our proposed method shows a constant error with increasing translational velocity, while the translational and rotational errors for **GS + refinement** increased linearly. In Figure 4, we evaluate the robustness of the algorithm under varying pixel noise with a constant translational velocity of 6.9m/s and zero angular velocity. The results show that our proposed method is less sensitive to the image noise than the **GS + refinement** approach for all the four combinations of shutter directions and camera motions.

Figure 5 shows the error plots when the zero angular velocity assumption is violated. Here, we vary the angular velocity while maintaining a constant translational velocity of 6.9m/s and an image noise of standard deviation 0.5 pixel. Our proposed method with refinement **RS + refinement** is observed to be more robust in the angular velocity interval of $[0-2.2]$rad/s. It is important to note that in practice the angular velocity of a camera mounted on hand-held devices or a moving car is normally $\leq$ 2.2rad/s. This can be seen from our real-world data taken from a camera mounted on a moving car in Section IV-B. The car reached a maximum angular velocity of only 0.31rad/s when making a 90° turn. Motion blur might also occur for any angular velocity that is greater than 2.2rad/s, thus making it useless for pose estimation.

In general it might be counter intuitive that the algebraic minimization of the *RS* solution outperforms the **GS + Refinement** approach. The reason is the initial solution of the **GS** approach only finds a reduced set of inliers that satisfies the **GS** perspective model. This poorly distributed set of inliers does not constraint the camera motion well enough for the geometric refinement to converge to the correct solution. In our synthetic experiments the number of inliers obtained in the **GS** case drops by over 70% with a motion of 12m/s.

B. Streetview Data

We evaluate the proposed algorithm on 5 different datasets - Dataset (a)-(e). These datasets were captured by a Google Streetview car. The images have a native resolution of 1944×2592 pixels and are recorded at 4Hz. The shutter time for the rolling shutter camera is 72ms. This corresponds to a motion of 0.5m during image formation for a car driving at 25km/h. The camera poses are obtained from a GPS/INS system and interpolated to provide a position and orientation for each scanline. We will refer to these poses as the ground truth poses. It should be noted that the baseline between consecutive images taken from the rolling shutter camera is approximately 1m, and we are not using any temporal constraints to estimate the velocity and pose of the camera. An overview of the 5 sequences is given in Table I. In the

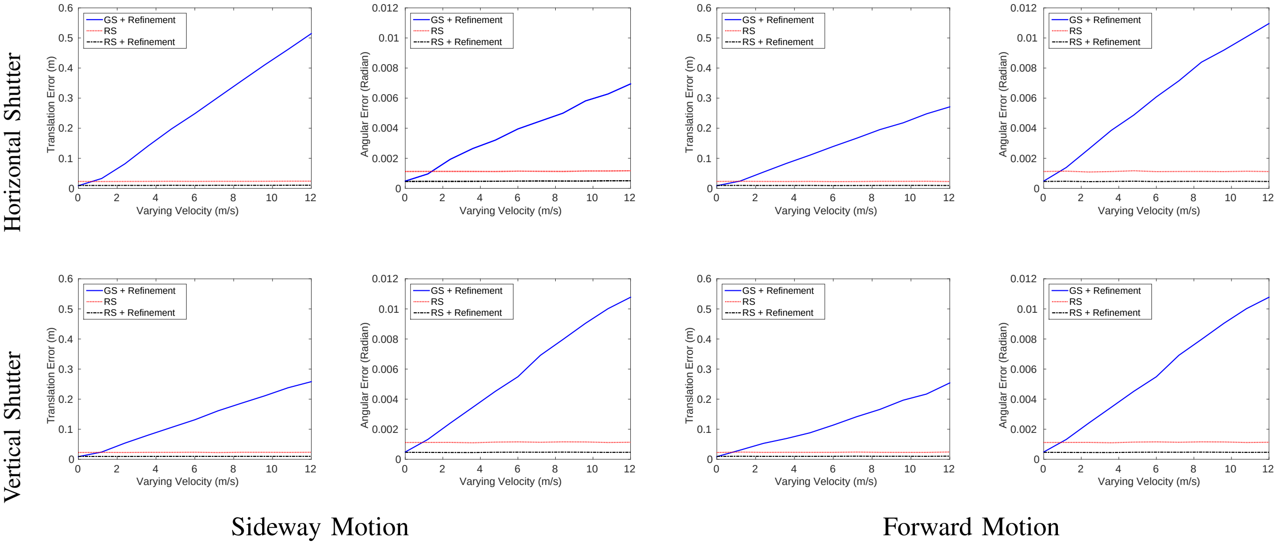


Fig. 3. Evaluation on increasing translational velocity with image noise of 0.5 pixel standard deviation and zero angular velocity. The first two columns show error plots for a sideways motions of the camera and the two last columns show errors for a forward (into the scene) moving camera.

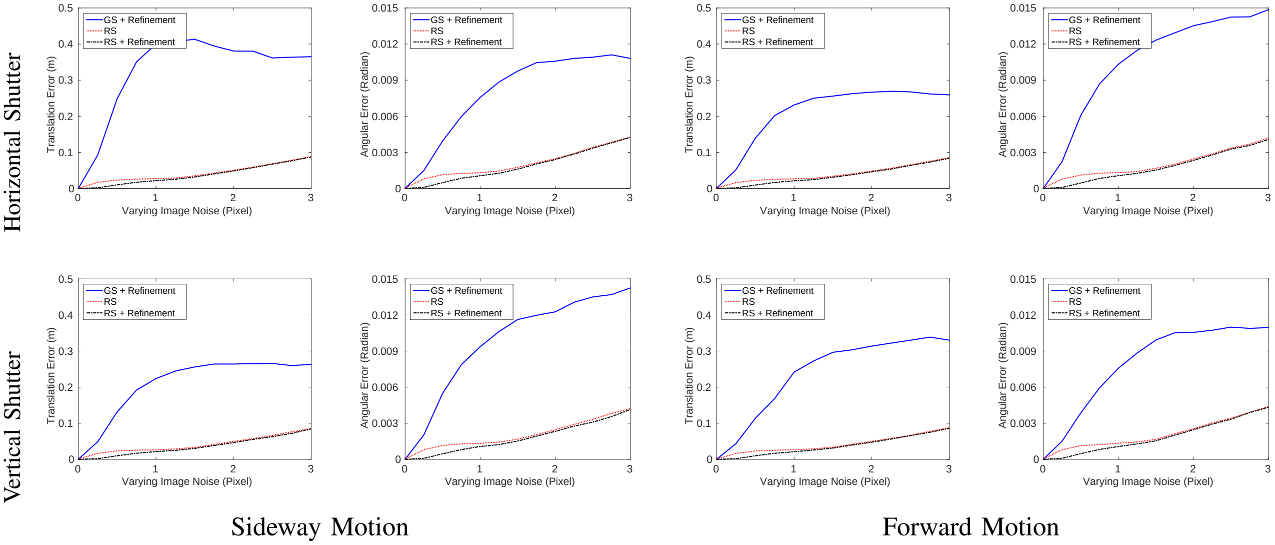


Fig. 4. Evaluation on increasing pixel noise with a fixed translational velocity of 6.9m/s and zero angular velocity. The first two columns show error plots for a sideways motions of the camera and the two last columns show errors for a forward (into the scene) moving camera.

first step, we create a 3D map by using all the even numbered images from each of the sequences. We extract and match SIFT [30] features using [31]. The point correspondences are then radially undistorted and triangulated using the provided GPS/INS pose. 3D points with large reprojection error (> 1 pixel) are discarded from the model. Figure 6 second row, shows the completed 3D models.

TABLE I
DATASET OVERVIEW

Dataset	# of Cameras	# of 3D Points	Length (m)	Median δ_θ (rad)	Median δ_t (m)
(a)	118	215936	276.82	0.0014	0.0539
(b)	178	338806	377.44	0.0109	0.1698
(c)	190	326254	451.40	0.0050	0.2975
(d)	156	336712	353.61	0.0016	0.0423
(e)	162	310893	376.39	0.0030	0.0951

For each odd numbered image in the sequences, we extract SIFT features, radially undistort the keypoints and match them to the 3D model. This gives us the potential 2D-3D correspondences. The matches are used in RANSAC together with our 5-point minimal solver to find a consensus set. The pose and velocity hypothesis is refined by minimizing the objective function in Equation 8 with the Google Ceres [28] solver using all inliers. We compare the final pose to the ground truth using the same error measure as in the synthetic evaluation. The third and fourth rows of Figure 6 show

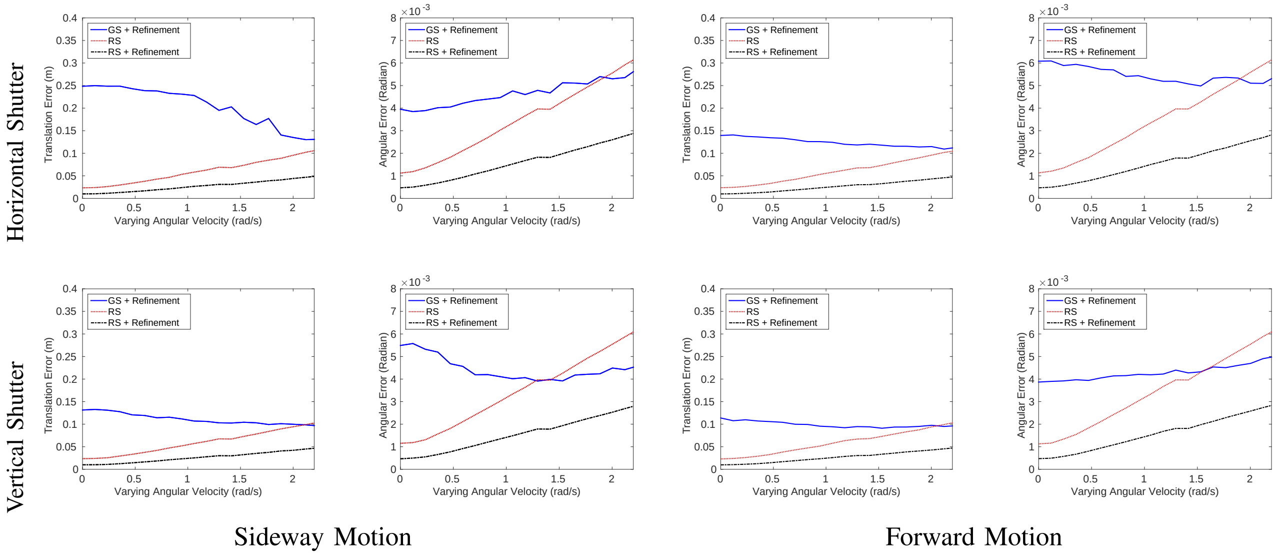
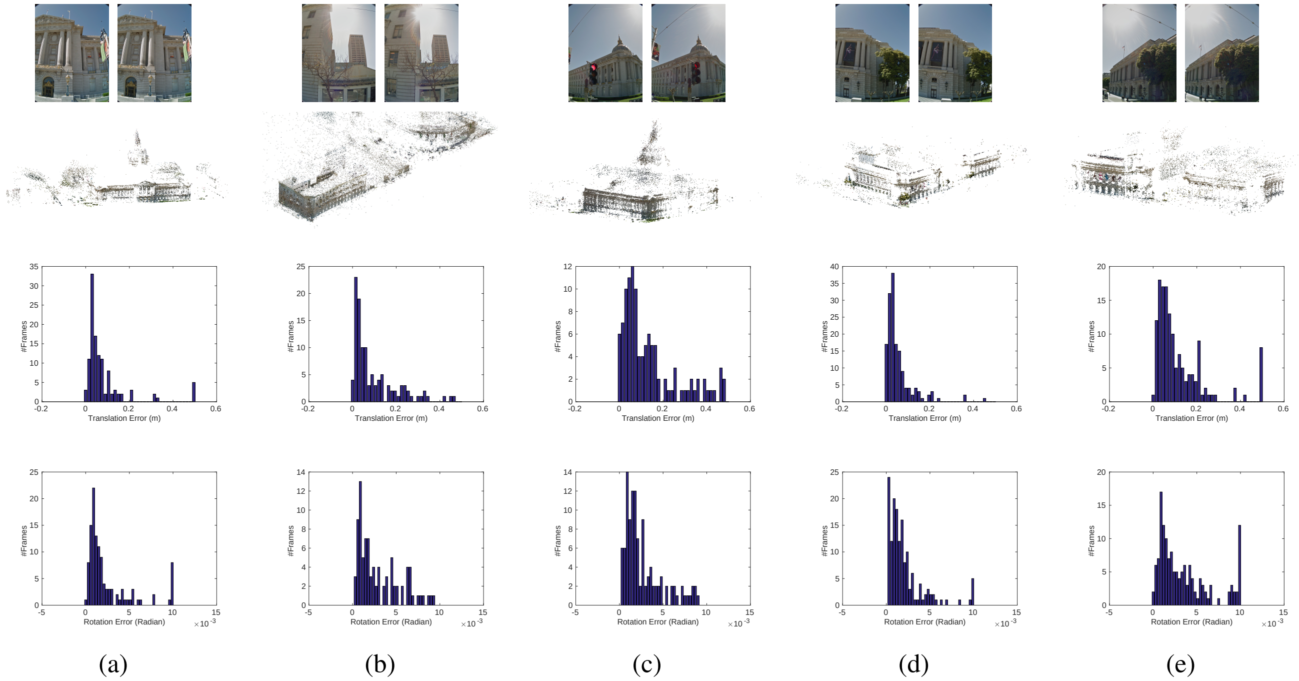


Fig. 5. Evaluation on increasing angular velocity with fixed image noise of 0.5 pixel standard deviation and translational velocity of $6.9m/s$. The first two columns show error plots for a sideways motions of the camera and the two last columns show errors for a forward (into the scene) moving camera.



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

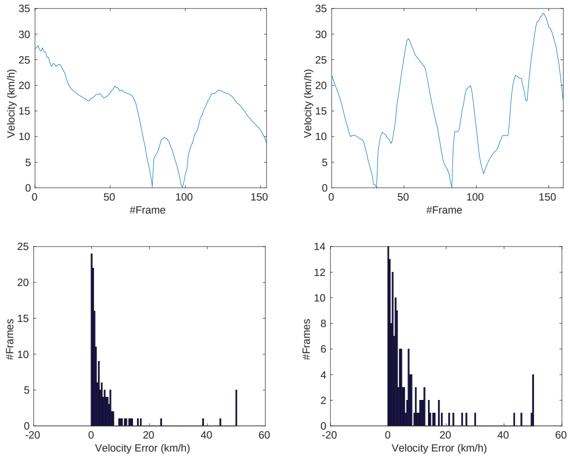
Fig. 6. First row shows the sample input images for Dataset (a)-(e). The second row shows the sparse 3d reconstructions of the City Hall and surrounding buildings in San Francisco. The third and fourth rows show the distribution of rotation and translation error against ground truth. Results with fewer than 20 matches were removed. Note that the last bin of the histogram is expands to infinity.

the translational and angular error distributions for each of the datasets. We can see that both the translational and angular errors are very small (translational error ~ 0.05m and rotational error ~ 0.125rad) as compared to the ground truth. This shows the accuracy of our algorithm on real-world datasets.

It is also interesting to show that we could potentially use a rolling shutter camera mounted on a car as cheap velocity sensor to estimate the speed of the car from a single rolling shutter image. In Figure 7, we show two examples of the application on Dataset (d) and (e). The top row shows the ground truth velocities. The bottom row shows we have achieved a median error of only 2.01km/h and 3.4km/h for the speed estimate on the datasets.

V. Conclusion

We derived a minimal solution to the pose and translation velocity estimation problem for the rolling shutter camera using 5-point 2D-3D correspondences. The solution is based on Gröbner Basis and can give up to 8 real solutions. Our 5-point algorithm can be used efficiently within RANSAC for



Dataset (d)   Dataset (e)

Fig. 7. Top row ground truth velocity of the car. Bottom row error distribution of estimated car speed. In case of the Dataset (d) we achieve an error of 2.01km/h and 3.4km/h for Dataset (e). Note that the last bin of the histogram is expanded to infinity.

robust estimation to get all the inlier 2D-3D correspondences. Finally, we relaxed the linear velocity assumption and do a non-linear refinement on the full velocity, i.e. translation and angular, and pose of the rolling shutter camera. We verified the accuracy of our algorithm on both synthetic and real-world data obtained from a Streetview car.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their constructive comments. This work was partially supported by a Google award.

REFERENCES

[1] R. Haralick, D. Lee, K. Ottenburg, and M. Nolle, “Analysis and solutions of the three point perspective pose estimation problem,” in *Computer Vision and Pattern Recognition (CVPR)*, 1991.

[2] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnp: An accurate o(n) solution to the pnp problem,” in *International Journal of Computer Vision (IJCV)*, vol. 81, no. 2, 2009.

[3] L. Quan and Z. D. Lan, “Linear n-point camera pose determination,” in *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 8, 1999, pp. 774–780.

[4] G. H. Lee, B. Li, M. Pollefeys, and F. Fraundorfer, “Minimal solutions for pose estimation of a multi-camera system,” in *International Symposium on Robotics Research (ISRR)*, 2013.

[5] D. A. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra*. Springer, 1997.

[6] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[7] C.-K. Liang, L.-W. Chang, and H. Chen, “Analysis and compensation of rolling shutter effect,” *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1323–1330, 2008.

[8] C. Geyer, M. Meingast, , and S. Sastry, “Geometric models of rolling-shutter cameras,” in *Proceedings of OMNIVIS*, 2005.

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

[10] D. Bradley, B. Atcheson, I. Ihrke, and W. Heidrich, “Synchronization and rolling shutter compensation for consumer video camera arrays,” in *International Workshop on Projector-Camera Systems (PROCAMS)*, 2009.

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

[12] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, “Effective calibration free rolling shutter removal,” *IEEE International Conference on Computational Photography (ICCP)*, 2012.

[13] G. Hanning, N. Forslöw, P.-E. Forssén, E. Ringaby, D. Törnqvist, and J. Callmer, “Stabilizing cell phone video using inertial measurement sensors,” in *The Second IEEE International Workshop on Mobile Vision*, 2011.

[14] J. B. Alexandre Karpenko, David E. Jacobs and M. Levoy, “Digital video stabilization and rolling shutter correction using gyroscopes,” Stanford University, Tech. Rep. CTSR 2011-03, 2011.

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

[16] E. Ringaby and P.-E. Forssén, “Efficient video rectification and stabilisation for cell-phones,” *International Journal of Computer Vision (IJCV)*, vol. 96, no. 3, pp. 335–352, 2012.

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

[18] G. Klein and D. Murray, “Parallel tracking and mapping on a camera phone,” in *Proc. Eigth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009.

[19] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment a modern synthesis,” in *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ser. ICCV. Springer, 2000, pp. 298–372.

[20] O. Ait-Aider and F. Berry, “Structure and kinematics triangulation with a rolling shutter stereo rig,” in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 1835–1840.

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

[22] M. Meilland, T. Drummond, and A. I. Comport, “A unified rolling shutter and motion blur model for 3d visual registration,” in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2016–2023.

[23] O. Ait-Aider, N. Andreff, J.-M. Lavest, and P. Martinet, “Exploiting rolling shutter distortions for simultaneous object pose and velocity computation using a single view,” in *IEEE International Conference on Computer Vision Systems (ICVS)*, 2006, pp. 35–35.

[24] O. Ait-Aider, N. Andreff, J. M. Lavest, and P. Martinet, “Simultaneous object pose and velocity computation using a single view from a rolling shutter camera,” in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 56–68.

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

[26] J. Ventura, C. Arth, G. Reitmayr, and D. Schmalstieg, “A minimal solution to the generalized pose-and-scale problem,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[27] Z. Kukelova, M. Bujnak, and T. Pajdla, “Automatic generator of minimal problem solvers,” in *European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 302–315.

[28] S. Agarwal, K. Mierle, and Others, “Ceres solver,” http://ceres-solver.org.

[29] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, “Review and analysis of solutions of the three point perspective pose estimation problem,” *Int. J. Comput. Vision*, vol. 13, no. 3, pp. 331–356, Dec. 1994.

[30] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” in *International Journal on Computer Vision (IJCV)*, vol. 60, no. 2, 2004, pp. 91–110.

[31] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms,” http://www.vlfeat.org/, 2008.