RRD-SLAM: Radial-distorted Rolling-shutter Direct SLAM

Jae-Hak Kim, Yasir Latif and Ian Reid

***Abstract*— In this paper, we present a monocular direct semi-dense SLAM (Simultaneous Localization And Mapping) method that can handle both radial distortion and rolling-shutter distortion. Such distortions are common in, but not restricted to, situations when an inexpensive wide-angle lens and a CMOS sensor are used, and leads to significant inaccuracy in the map and trajectory estimates if not modeled correctly. The apparent naive solution of simply undistorting the images using pre-calibrated parameters does not apply to this case since rows in the undistorted image are no longer captured at the same time. To address this we develop an algorithm that incorporates radial distortion into an existing state-of-the-art direct semi-dense SLAM system that takes rolling-shutters into account. We propose a method for finding the generalized epipolar curve for each rolling-shutter radially distorted image. Our experiments demonstrate the efficacy of our approach and compare it favorably with the state-of-the-art in direct semi-dense rolling-shutter SLAM.**

I. Introduction

Visual SLAM systems generally assume that the camera used for SLAM follows the ideal pin-hole model, in which, all the pixels are simultaneously exposed to light for the same amount of time, the so called "global shutter". This is shown in Figure 1(a). However, cheap CMOS based sensors expose and read-out every scan-line (image row) sequentially. This is termed the "rolling-shutter". Under rapid motion, this can lead to images "bending" as each scan-line is acquired at a different location in space as shown in Figure 1(b). Another factor which leads to non-ideal image acquisition is that of lens distortion. The imaging system bends light as it enters the camera leading to distortion that is radially symmetric around the center of projection. This is more prominent for wide field-of-view lenses. The effect of radial distortion together with rolling-shutter can be seen in Figure 1(c). If the camera has a global shutter, radial distortion can be modeled in a calibration step and its effects reversed in a preprocessing step. However, for a rolling-shutter camera, where each scan-line needs to be modeled separately, things are not as straight-forward, and are the focus of this work. A first intuition might be to simply model and remove radial distortion from images (as is done for global-shutter cameras) and then use a rolling-shutter Direct-SLAM method. However, undistorted images thus created, have pixels from rows that have been acquired at different times, thus violating the scan-line based time model of the rolling-shutter camera widely adopted in [9], [11], [2], [12], [7], [8].

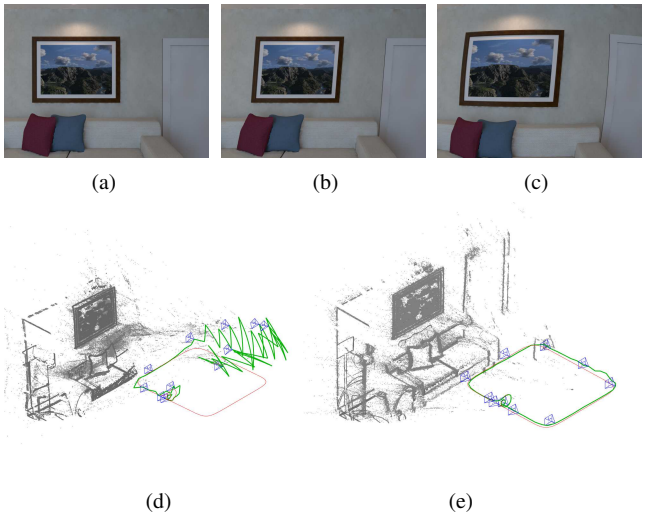

Fig. 1. (a) Global-shutter (GS) image, (b) Rolling-shutter (RS) image, (c) Radial-distorted rolling-shutter (Radial-RS) image, (d) Result of RSD-SLAM [9] on the Radial-RS images and (e) **Our method (RRD-SLAM)** on the Radial-RS images. Green lines indicate the estimated camera trajectory. Blue camera frustums show keyframe poses. Red lines are the ground-truth.

Rolling-shutter and radial distortion impact both tracking and mapping in Direct-SLAM approaches [8], [9] leading to a degradation in performance. The state-of-the-art Direct-SLAM system by Kim *et al*. [9] addresses the rolling-shutter problem, however, their method does not take into account the effect of radial distortion for rolling-shutter cameras. The effect of failing to model the radial distortion is shown in Figure 1(d), which can be contrasted with the method of this present paper in which we address both rolling-shutter and radial distortion in a Direct-SLAM system. This provides more accurate camera localization and structure mapping as can be seen in Figure 1(e). More concretely, this work makes the following contributions:

1) To the best of our knowledge, this is the first (feature-less) Direct-SLAM system — via direct image alignment that does not require feature detection and matching — which can handle both rolling-shutter and radial distortion in images, therefore widening the applicability of Direct-SLAM approach regardless of types of the shutter and lens.
2) For tracking, we model a new row-time parameter compensating for the radial distortion. This new parameter is used in the optimization which solves a con-

The authors are with the School of Computer Science, University of Adelaide, Adelaide, Australia. YL and IR are also with the Australian Center for Robotic Vision. {jaehak.kim,yasir.latif,ian.reid}@adelaide.edu.au

The authors are extremely grateful to the Australian Research Council for funding this research through project DP130104413, the ARC Centre of Excellence for Robotic Vision C E140100016, and through a Laureate Fellowship FL130100102 to IR.

tinuous B-spline camera trajectory for rolling-shutter cameras with radial distortion.

3) For mapping, we propose an efficient method of finding a generalized epipolar curve for radial-distorted rolling-shutter images by posing it as an optimization problem, which is significantly faster than a brute-force solution that finds intersections by processing every pixel.

II. Approach

Since the images that we are dealing with contain both radial and rolling-shutter distortion, tracking and mapping have to be carried out differently than a conventional Direct-SLAM system. The basic idea is that each row in the original (Radial-RS) image needs to be modeled as a single row-camera due to the rolling-shutter. However, when radial distortion is removed, this row maps to a curve in the undistorted image and can no longer be treated as a separate row-camera since different rows of the original image contribute to a single row. To cater for this, our system incorporates both the mapping and inverse mapping function for radial distortion in tracking and mapping. This is one of the major differences of our system compared to RSD-SLAM [9] which assumes radial distortion free images as its input.

A. Definitions

In the rest of the text, we use the follow terms:

- **GS (Global-Shutter) image:** An image formed by pinhole camera projection without any radial distortion and rolling-shutter distortion (Figure 1(a)).
- **RS (Rolling-Shutter) image:** An image with only rolling-shutter distortion (Figure 1(b)), that is, an image captured using a rolling-shutter camera with a perfect pinhole model and therefore no radial distortion.
- **Radial-GS image:** An image with only radial distortion captured by a global-shutter camera using a wide field-of-view lens.
- **Radial-RS image:** An image with both radial and rolling distortion (Figure 1(c)), e.g. images captured by a rolling-shutter camera with a wide field-of-view lens. This is the input image we focus on this paper.
- **Radial-RS-Undist image:** An image after radial distortion removal of a Radial-RS image. Note that this is different from RS image as radial distortion removal applied after rolling-shutter effects. We also simply refer as "undistorted" images.
- **Radial-RS stereo:** A stereo configuration consisting of two Radial-RS images captured at different views.
- **D-SLAM:** We use D-SLAM to denote the Direct-SLAM approach of Engel *et al.* [5], [4]. In the Direct-SLAM approach, the tracking part does not extract features but instead uses a direct image alignment over semi-dense points. The direct image alignment assumes a reference image containing 3D depths that are used to warp the reference image to a current tracking image by optimizing a 6 degree of freedom (dof) pose. A mapping part of Direct-SLAM uses photometric cost as in dense multi-view stereo to recover 3D depths.
- **RSD-SLAM:** We use the abbreviation RSD-SLAM for the Rolling-Shutter Direct-SLAM by Kim *et al.* [9] that extends D-SLAM to use rolling-shutter images instead of global-shutter images.
- **RRD-SLAM:** We call our proposed SLAM system as RRD-SLAM (Radial-distorted Rolling-shutter Direct-SLAM), which is capable of using Radial-RS images as input.

B. Tracking in Radial-RS image

We follow the same convention in [9] using a B-spline curve for rolling-shutter images. Important parts of the formulation are repeated here and the reader is referred to [9] for further details. We assume that a pixel $\mathbf{p}$ with depth p_z in an image can be mapped to the corresponding point $\mathbf{q}$ using a warp function $\mathcal{W}(\cdot)$ such that

$$\mathbf{q} = \mathcal{W}(\ \mathbf{p}\ ,\ 1/p_z\ ,\ \xi_p^q\ ) \tag{1}$$

where ξ_p^q encapsulates the 6 degrees of freedom transformation that maps $\mathbf{p}$ to $\mathbf{q}$. Due to the rolling shutter, each row in the input image is modeled as a separate row-camera. The row-pose of a pixel $\mathbf{q}$ is related to the time of acquisition specified by u_q, which in turn is a function of q_y, the row number of $\mathbf{q}$ in the image:

$$u_q = (q_y + S_I - S_s)/N_s\ .$$

Here, S_I is the image starting row, S_s is the B-spline segment starting row and N_s is the number of image rows in a B-spline segment.

This formulation allows us to model row-cameras without radial distortion as a row-camera maps directly to a row in the next image. However, when radial distortion is present, each row transforms into a curve when undistortion is applied. The row coordinate in the original *distorted* image is the correct choice for the pose, but since we use undistortion before the image is used as input to the SLAM method, a distortion function is used to obtain the original *distorted* coordinate. As all distorted coordinates differ by both row and column coordinates in the undistorted image, we replace the row-time parameter u_q in [9] with a new row-time parameter u_q^* which depends on a radial distortion function $f(\cdot)$. The final optimization problem for $\mathbb{SE}(3)$ tracking then becomes:

$$\underset{\Xi,\ u}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{k} \sum_{\mathbf{p} \in D_i} \rho \left({E_p}^2 + \alpha {E_r}^2 \right) \tag{2}$$

$$E_p = \mathcal{I}_{S_i}(\mathbf{p}) - \mathcal{I}_{T_i}(\mathbf{q}) \tag{3}$$

$$E_r = u_q^* - u \tag{4}$$

$$u_q^* = (f(\mathbf{q})_y + S_I - S_S)/N_s\ , \tag{5}$$

where $f(\mathbf{q})_y$ is a y-coordinate of the distorted pixel of a pixel $\mathbf{q}$ in the undistorted image by a radial distortion function f. Also, E_p is an intensity error of point correspondences $\mathbf{p}$ and $\mathbf{q}$ between the source image $\mathcal{I}_{\mathcal{S}_i}$ and target image $\mathcal{I}_{\mathcal{T}_i}$.

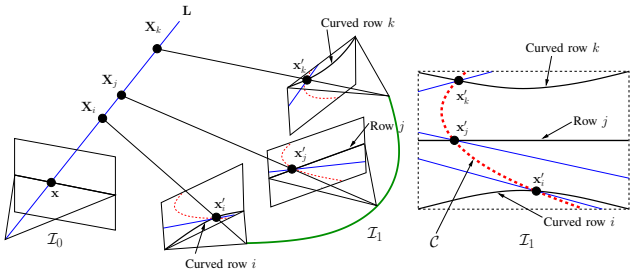

Fig. 2. ***Generalized epipolar curve in Radial-RS stereo.*** Given two Radial-RS images, the possible stereo correspondences (black circles) of an image point in the first image $\mathcal{I}_0$ are shown as $\mathbf{x}'_i$, $\mathbf{x}'_j$ and $\mathbf{x}'_k$ at each curved rows i, j and k, respectively. The curved rows (black) in the second image $\mathcal{I}_1$ were straight lines in the original distorted image. The generalized epipolar curve $\mathcal{C}$ is composed of intersections points of each curved row and the epipolar line (blue).

Another error E_r is an row-time error constraining the row-time u of B-spline curve and the image row-time u_q^* for the point $\mathbf{q}$. The total cost is a sum of Huber function $\rho(\cdot)$ of the sum of squared error of the intensity and row-time error for all pixel $\mathbf{p}$ in the semi-dense points $\mathcal{D}_i$ over k pairs of images. The optimization problem finds the optimal solution of the B-spline time parameter u and poses of k control points Ξ of B-spline camera trajectory.

C. Mapping in Radial-RS image

1) Finding a generalized epipolar curve in Radial-RS image: We first recap a generalized epipolar curve for rolling-shutter stereo without radial distortion, as described in [9]. The generalized epipolar curve is found as the locus of intersection points of an epipolar line and a row scanline of the image. In our problem considering radially distorted images with rolling-shutters, a generalized epipolar curve should be constructed using the intersection points of an epipolar line and the curve in the undistorted image generated by a row-scanline of the (original) distorted image, as shown in Figure 2. Thus in contrast to the case in [9] where each intersection is easily found in a closed-form from a pair of straight lines, the Radial-RS problem must find the intersection of a line and a curve.

Let us assume our camera follows a FOV camera model [10], [3] for radial distortion. Although we use the FOV model as one of examples here, our method is applicable to any camera radial distortion model once mapping and its inverse mapping for radial distortion are derived.

We begin by deriving the intersection point analytically. Without loss of generality, we define image pixels in normalized coordinates as $\mathbf{v} = (x,\ y,\ 1)$ and $\mathbf{v}' = (x',\ y',\ 1)$ for (radial) undistorted and (radial) distorted pixels, respectively, as follows:

$$\mathbf{v} = \mathtt{K}^{-1}\mathbf{x} \tag{6}$$

$$\mathbf{v}' = \mathtt{K}'^{-1}\mathbf{x}' \ , \tag{7}$$

where $\mathbf{x} = (x_u,\ y_u, 1)^\top$ and $\mathbf{x}' = (x_d,\ y_d,\ 1)^\top$ are image points in undistorted image and distorted image, respectively, and $\mathtt{K}$ and $\mathtt{K}'$ are camera calibration matrices for undistorted and distorted image, respectively. The radius of the distorted image point $\mathbf{v}'$ to the image optical center and that for the undistorted image point $\mathbf{v}$ are written as

$$r' = \sqrt{x'^2 + y'^2} = \frac{1}{w}\arctan\left(2r\tan(w/2)\right) \tag{8}$$

$$r = \sqrt{x^2 + y^2} = \frac{\tan(r'w)}{2\tan(w/2)} \ , \tag{9}$$

where w is a radial distortion parameter. Then, the undistorted image pixel point $\mathbf{v}'$ can be obtained from the distorted image pixel $\mathbf{v}$ with the ratio r to r' as follows

$$x = \frac{r}{r'}x' \tag{10}$$

$$y = \frac{r}{r'}y' \ . \tag{11}$$

We replace r' in (11) with (8), define $T \equiv \tan(w/2)$, then we obtain

$$y = \frac{wr}{\arctan\left(2rT\right)}y' \ . \tag{12}$$

Approximation of this solution may be found by applying Taylor expansion, however, as accuracy is more important in our system we propose two methods:

2) (Method 1) Finding intersections for each pixel: Given y', we can formulate the system of equations which consists of the epipolar line and a family of circles. This will eventually give us a solution of intersections for each pixel in the row y' from a quadratic equation (refer to Appendix). Therefore, we must process all pixels in several rows to obtain the generalized epipolar curve. The algorithm is shown in Algorithm 1.

Algorithm 1 Method_1 (Slow version)

Input: a, b, c : coefficients of the epipolar line equation
w, calibration distortion parameter;
${}^t y_d$, a row of the distorted target image

Output: $\mathtt{C} : \{{}^t x_u,\ {}^t y_u\}$, intersection points in the target undistorted image

1: **for** ${}^t x_d$ **in** all columns of target distorted image **do**
2: ${}^t r_d := \sqrt{({}^t x_d - c_x)^2 + ({}^t y_d - c_y)^2}$
3: $d := \tan(w * {}^t r_d)/(2 * \tan(w/2))$
4: $A := a^2 + b^2$
5: $B := 2 * a * c + 2 * a * b * c_y - 2 * b^2 * c_x$
6: $C := b^2 * c_x^2 * c^2 + (c + b * c_y)^2 - b^2 * d^2$
7: ${}^t x_{u+} := (-B + \sqrt{B^2 - 4 * A * C})/(2 * A)$
8: ${}^t x_{u-} := (-B - \sqrt{B^2 - 4 * A * C})/(2 * A)$
9: ${}^t y_{u+} := -(a/b) * {}^t x_{u+} - (c/b)$
10: ${}^t y_{u-} := -(a/b) * {}^t x_{u-} - (c/b)$
11: **if** $({}^t x_{u+}, {}^t y_{u+}) \in \mathcal{R}^2$ and inside the image **then**
12: Store it into a set $\mathtt{C}$
13: **if** $({}^t x_{u-}, {}^t y_{u-}) \in \mathcal{R}^2$ and inside the image **then**
14: Store it into a set $\mathtt{C}$

An example of intersection points found for sampled image points by the Algorithm 1 is shown in Figure 3. However, the Algorithm 1 is inefficient because of the exhaustive searching for all columns (line 1 in the algorithm).

Fig. 3. ***Example of intersection points found by proposed algorithms.*** Point-curve correspondences in two views for Radial-RS images. The left image is distorted original first image (Frame 7), and the right image is undistorted second image (Frame 12). Same colors indicate correspondence. For each point in the left image, we show the corresponding generalized epipolar curve in the right image. (Figure best seen in color).

3) (Method 2) Finding intersections by optimization: We propose a more efficient approach which solves for a row instead of each pixel in the row, and is therefore faster than the Algorithm 1. Firstly, we formulate an objective function which is the algebraic cost for an epipolar line equation for a point passing through the curve that is a undistortion of the row scan-line. It is derived from (12) and the epipolar equation $ax + by + c = 0$ as follows:

$$y_u^* = \underset{y_u}{\operatorname{argmin}} \left(\frac{wr}{\arctan(2rT)} y' - y \right)^2 , \qquad (13)$$

where $r = \sqrt{x^2 + y^2}$ with $x = (-(b/a)y_u - (c/a) - \mathtt{K}_{13})/\mathtt{K}_{11}$ and $y = (y_u - \mathtt{K}_{23})/\mathtt{K}_{22}$. $\mathtt{K}_{ij}$ refers to the element at i-th row and j-th column of the camera calibration matrix $\mathtt{K}$. An x_u^* can be found by $x_u^* = -(b/a)y_u^* - (c/a)$. This holds true only if a is nonzero (e.g. non-horizontal epipolar line).

Therefore, we derive another cost for the horizontal epipolar line (when a = 0) as follows:

$$x_u^* = \underset{x_u}{\operatorname{argmin}} \left(\frac{wr}{\arctan(2rT)} y' - y \right)^2 , \qquad (14)$$

where $x = (x_u - \mathtt{K}_{13})/\mathtt{K}_{11}$ and $y = -c/b$. Then y_u^* can be found from the epipolar line equation. As there can be two least two minima, we can introduce a bound condition on the parameter from a search region or prior depths. In experiments, we validated that Method 2 is significantly faster (about four orders of magnitude) than Method 1.

4) Degenerate case handling: The motion of the camera parallel to the row scanline of rolling-shutter camera, is a degenerate case as the problem reduces to finding the intersection of two identical lines. In Radial-RS images, this happens less frequently compared to RS image case. It occurs only at the center row which passes through the optical center. This row does not get bent under radial distorted (i.e. it is still mapped to a straight line in distorted images). Therefore, we can safely ignore this degenerate case in practice.

5) Building uniform and dense sample points: When the generalized epipolar curve is nearly horizontal, the intersection points on the generalized epipolar curve will be far apart from each other. This is because we sample points in a row-major order of the image. To sample uniformly and preferably densely (not more than 2 pixels away from each other), we linearly interpolate between existing point on the generalized epipolar curve to build uniform and dense sample points. These uniformly sampled dense points are then used as candidate points in Radial-RS stereo.

Algorithm 2 Method_2 (Fast version)

Input: a, b, c : coefficients of the epipolar line equation
w, calibration distortion parameter;
${}^t y_d$, a row of the distorted target image

Output: $\mathtt{C} : \{{}^t x_u,\ {}^t y_u\}$, intersection points in the target undistorted image

if a is not zero **then**
  Solve y_u^* minimizing (13) by nonlinear least squares given the bound of y_u.
  $x_u^* := (-b/a) * y_u^* - (c/a)$
else
  Solve x_u^* minimizing (14) by nonlinear least squares given the bound of x_u
  $y_u^* := (-c/b) * x_u^*$
if solutions exist **then**
  $\mathtt{C} := (x_u^*,\ y_u^*)$

6) Depth map propagation for Radial-RS images: When a keyframe gets changed, a current depth map in the old keyframe needs to be propagated as a new depth map to the new keyframe. As two keyframe images have both radial distortion and rolling-shutter distortion, we obtained a row-pose of the rolling-shutter from a row of the pixel in the old keyframe after distorting the pixel back to the original input image, then the depth map is transformed to the new keyframe after considering its projection followed by distortion coincides at a row we are searching in the new keyframe.

III. Experiments

This section presents experimental results for the proposed method. All optimizations were carried out using Ceres [1] as the solver on a desktop computer. The source code for RRD-SLAM can be found at: `https://github.com/jaehak/rrd_slam`.

A. Synthetic dataset

A synthetic dataset for Radial-RS images was generated by POV-ray software similar to [9], [6] with a fisheye camera model which meets a FOV camera model.

Radial-RS images can simply be built by stacking rows of Radial-GS images at each distinct row pose. We built a Radial-RS image from a center cropped region that is 480 rows for 640 columns from a fisheye image rendered by POV-Ray software. Poses of the camera are pre-determined and consist of a small looped trajectory. Samples of the Radial-RS images obtained by rendering and followed by row-stacking are shown in Figure 4.

Fig. 4. **Samples of GS, RS and Radial-RS images.** (Top-row) Images rendered by POV-ray via perspective projection without any distortion for a loop trajectory. (Middle-row) Images rendered and rolling-shutter effect is simulated by stacking scan-lines of each row. (Bottom-row) Images rendered for rolling-shutter effect accompanying radial distortion via fisheye projection.

TABLE I
COMPARISON OF RRD-SLAM TO ORB-SLAM

	RRD-SLAM	ORB-SLAM
Translational RMSE(cm)	**5.34**	94.85
Rotation RMSE (degrees)	**0.4698**	8.9811

We compare our system with RSD-SLAM using two different images as inputs: **(a)** Radial-RS and **(b)** Radial-RS-Undist images. Figure 5 and Figure 6 respectively show the outputs of RSD-SLAM on Radial-RS images and Radial-RS-Undist images. It can be seen that with Radial-RS images, the system fails catastrophically and with the Radial-RS-Undist images, there is a systematic error introduced by the violation of the rolling-shutter assumption. In contrast, the proposed system shown in Figure 7, shows improvement tracking and mapping, that comes from treating the rolling shutter and radial distortion correctly.

We compare the performance of RRD-SLAM against ORB-SLAM [13] with the same data (Radial-RS-Undist) used in Figure 7. The results are given in Table I, the output of monocular ORB-SLAM was scaled to match the scale of the ground truth. This shows the importance of correctly modeling radial and rolling shutter distortions.

B. Real sequence

A sample result of our RRD-SLAM on a real sequence captured by GoPro camera with CMOS sensor is shown in Figure 8. The images are captured at a resolution of 1920×1080 at 30Hz with a wide field-of-view lens and contain a large amount of radial distortion. The camera was mounted on a moving bike and therefore rolling-shutter effects are observable. Our initial result shows refining an initial random depth correctly as the system processes the sequence, which indicates our system is handling the input Radial-RS images of GoPro camera. Also, Radial-RS stereo matching correctly finds matched points between the pair of stereo. Lastly, tracking results show the initial pose of the camera looking at the left-side has a correct turning to the right-side. However, the estimated trajectory shows many oscillation mostly happening when the estimation is not reaching an optimal solution. All these indicate our full system correctly estimates tracking and mapping. However, this is yet a preliminary result as it contains only 100 frames which probably is in the process of initial phase refining random depths for the entire sequence.

IV. CONCLUSION AND DISCUSSION

In this paper, we presented a Direct semi-dense SLAM system that can handle images containing both radial distortion and rolling-shutter. Our system allows the use of many inexpensive cameras since it models radial distortion and rolling shutter. It overcomes limitation of LSD-SLAM [8] and RSD-SLAM [9] on radially-distorted rolling-shutter images. Our system, as presented here, is not for real-time applications as the processing time is about 450 sec per frame. We are currently working on improving this by reducing the amount of parameters to be estimated in tracking and a more optimized mapping implementation.

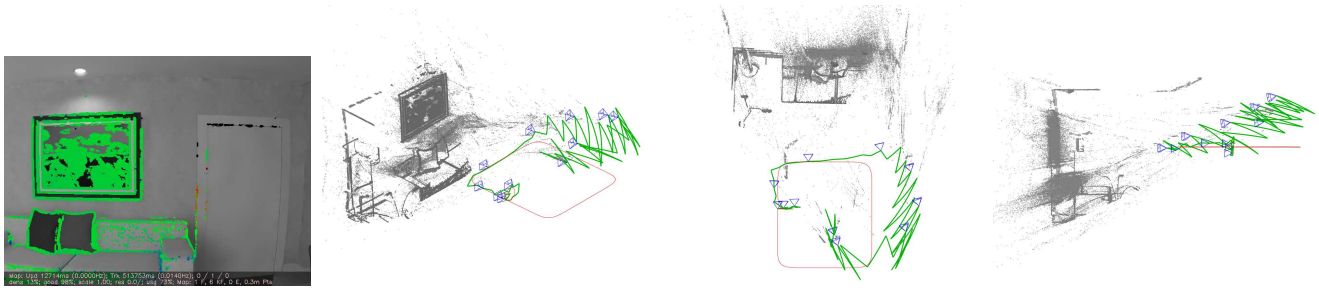

Fig. 5. ***A result of RSD-SLAM [9] on synthetic Radial-RS images.*** From Left to Right. Depth images at a keyframe, top-side, top and side-view of the 3D map and camera trajectory estimated poorly and some map points are missing (e.g. door). No loop-closing is used and the initial ground-truth depth is given.

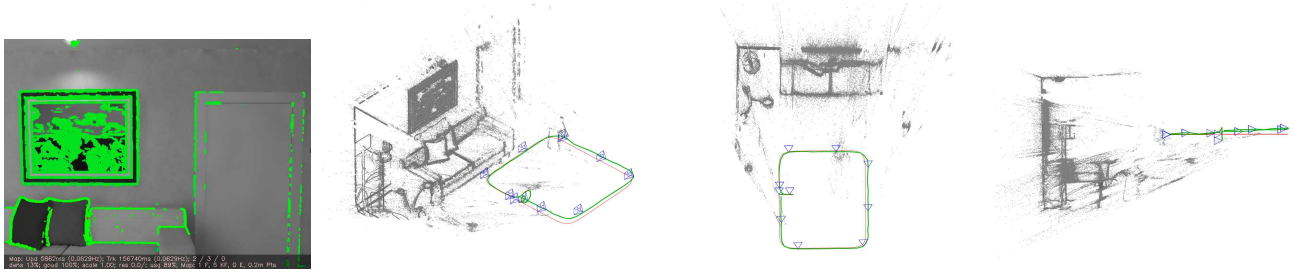

Fig. 6. ***A result of RSD-SLAM [9] on synthetic Radial-RS-Undist images*** From Left to Right. Depth images at a keyframe, top-side, top and side-view of the result. Input Radial-RS-Undist images were given by removing radial distortion of Radial-RS images. No loop-closing is used and the initial ground truth depth is given. Notice that the estimated trajectory drifts out of the plane of motion of the ground-truth trajectory.

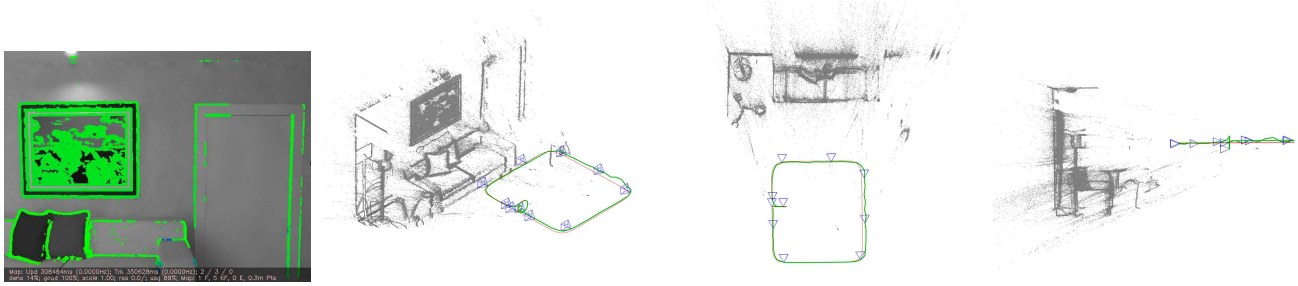

Fig. 7. ***A result of our RRD-SLAM on synthetic Radial-RS images.*** From Left to Right. Depth images at a keyframe, top-side, top and side-view of the result. Depths and camera trajectory are estimated close to the ground-truth.

REFERENCES

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[2] Y. Dai, H. Li, and L. Kneip. Rolling shutter camera relative pose: generalized epipolar geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4132–4140, 2016.

[3] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.

[4] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Dcale Direct monocular SLAM. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[5] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013.

[6] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *Robotics and automation (ICRA), 2014 IEEE international conference on*, pages 1524–1531. IEEE, 2014.

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

[8] C. Kerl, J. Stuckler, and D. Cremers. Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2264–2272, 2015.

[9] J.-H. Kim, C. Cadena, and I. Reid. Direct semi-dense slam for rolling shutter cameras. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1308–1315. IEEE, 2016.

[10] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[11] S. Lovegrove, A. Patron-Perez, and G. Sibley. Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In *BMVC*, 2013.

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

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

APPENDIX

Since we consider a fixed row y_d in the distorted image, we may say r_d is a function of a column x_d given the row y_d.

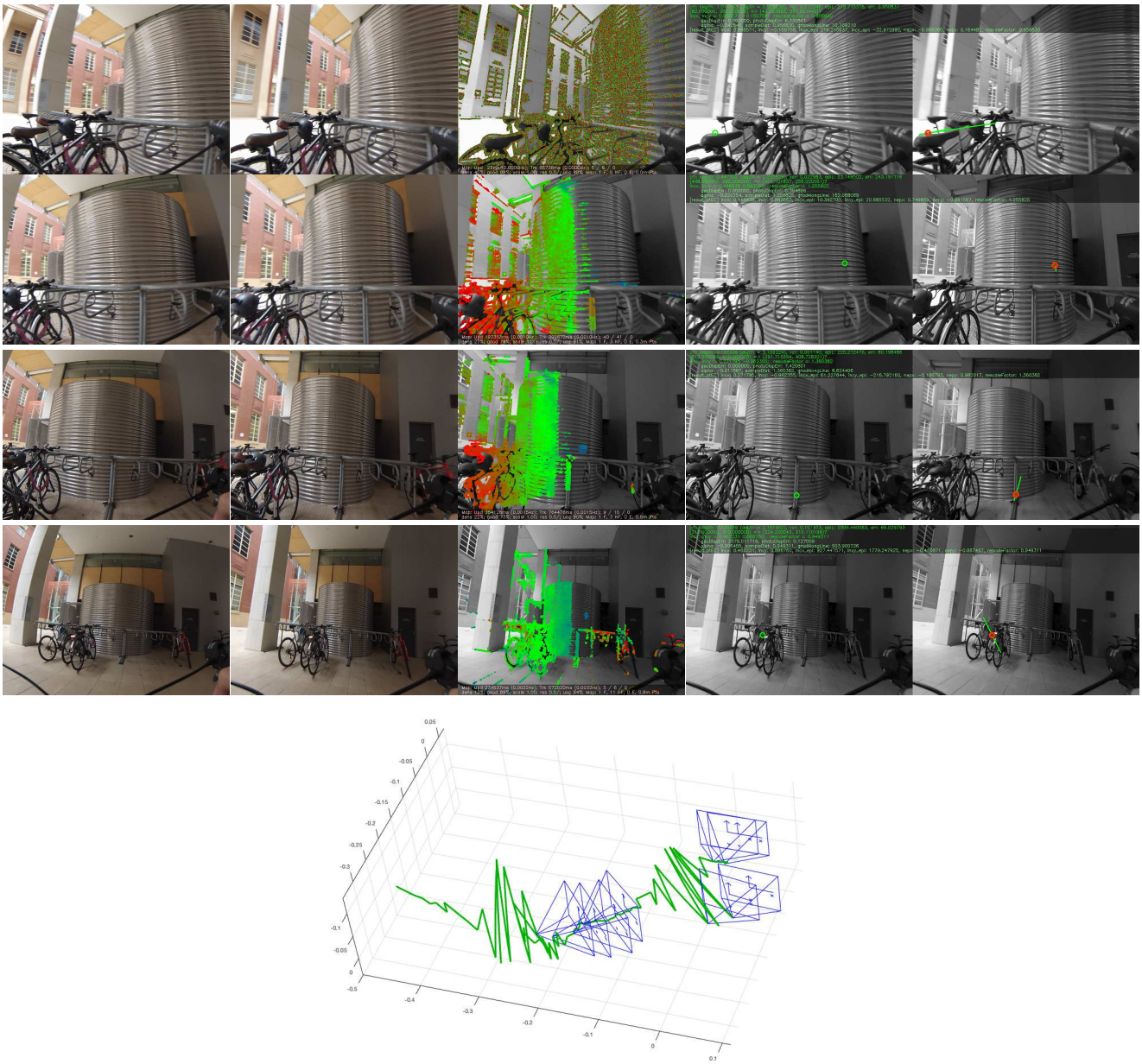

Fig. 8. ***A result of our RRD-SLAM on a real sequence.*** (Top to Bottom) Frame 6, 23, 93 and 125. (Left to Right) Input image, undistorted image, depth at keyframe image, and a pair of stereo with an epipolar curve and match. Initial random depths (Top-row and 3rd column) were correctly updated while our system is running (Bottom-row and 3rd column) Red color indicates near and green color means far distance.

Also, $r_u^2 = ((x_u - c_x^u)/f_x^u)^2 + ((y_u - c_y^u)/f_y^u)^2$, then

$$((y_u - c_y^u)/f_y^u)^2 = r_u^2 - ((x_u - c_x^u)/f_x^u)^2 \tag{15}$$

$$= \left(\frac{\tan\left(r_d(x_d; y_d)w\right)}{2\tan(w/2)}\right)^2 - ((x_u - c_x^u)/f_x^u)^2 \ . \tag{16}$$

This is a family of circles with each radius (the first term in (16)) determined by a function of x_d. Then, with an epipolar line equation $ax_u + by_u + c = 0$ and known bounds of x_u and y_u, we may solve a system of equations, which gives two intersection points.

By rewriting the first term of (16) as d^2, we formulate the system of equations as follows:

$$(x_u - c_x)^2 + (y_u - c_y)^2 - d^2 = 0 \tag{17}$$

$$ax_u + by_u + c = 0 \tag{18}$$

Then, its solution is found by replacing y_u in (17) with (18) as

$$(x_u - c_x)^2 + \left(-\frac{a}{b}x_u - \frac{c}{b} - c_y\right)^2 - d^2 = 0 \tag{19}$$

$$Ax_u^2 + Bx_u + C = 0 \ , \tag{20}$$

where

$$A = a^2 + b^2 \tag{21}$$

$$B = 2ac + 2abc_y - 2b^2c_x \tag{22}$$

$$C = b^2c_x^2 + (c + bc_y)^2 - b^2d^2 \tag{23}$$

Using the quadratic formula, we can find $x_u \in \mathcal{R}$. Then, y_u can be found by (18). We use Ceres [1] to solve the system of equations.