Sparse to Dense 3D Reconstruction from Rolling Shutter Images

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

Department of Computer Science[1]
ETH Zürich, Switzerland
{saurero, marc.pollefeys}@inf.ethz.ch

Department of Mechanical Engineering[2]
National University of Singapore
mpelgh@nus.edu.sg

Abstract

It is well known that the rolling shutter effect in images captured with a moving rolling shutter camera causes inaccuracies to 3D reconstructions. The problem is further aggravated with weak visual connectivity from wide baseline images captured with a fast moving camera. In this paper, we propose and implement a pipeline for sparse to dense 3D construction with wide baseline images captured from a fast moving rolling shutter camera. Specifically, we propose a cost function for Bundle Adjustment (BA) that models the rolling shutter effect, incorporates GPS/INS readings, and enforces pairwise smoothness between neighboring poses. We optimize over the 3D structures, camera poses and velocities. We also introduce a novel interpolation scheme for the rolling shutter plane sweep stereo algorithm that allows us to achieve a 7× speed up in the depth map computations for dense reconstruction without losing accuracy. We evaluate our proposed pipeline over a 2.6km image sequence captured with a rolling shutter camera mounted on a moving car.

1. Introduction

The majority of image sensors on the market today (found in mobile phones, and compact cameras etc.) are CMOS sensors. In contrast to standard CCD sensors which have global sensor readout, classical CMOS sensors have sequential readout. This leads to sequential exposures of each image row or column - commonly known as rolling shutter. For example, typical readout times in todays mobile phones are around 10-40ms [23, 28]. Given such a delay, significant deformations appear in the image when the camera is in motion. Assuming a camera with a readout time of 35ms and moving at 5km/h, objects which are closer than 25m will show deformations due to the rolling shutter [26] effects.

In the last decade, Structure from Motion (SfM) techniques have been used to build 3D models from both ordered and unordered image sequences [29, 1, 9, 16]. These

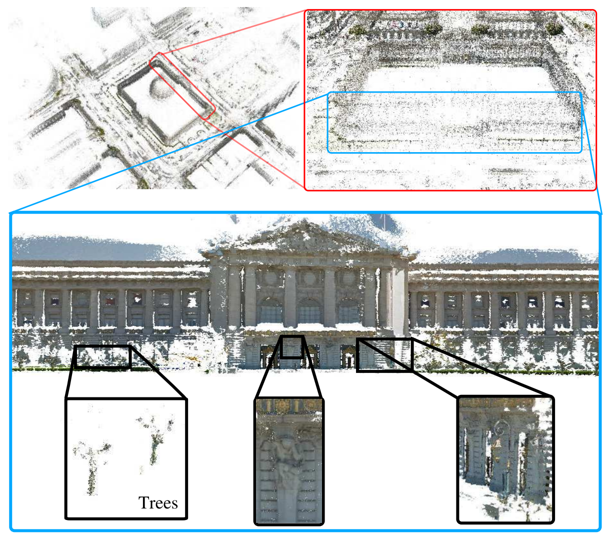


Figure 1. Sparse and dense 3D reconstructions from wide baseline images captured with a rolling shutter camera mounted on a moving car.

SfM techniques rely on a global shutter camera model and become brittle when used on rolling shutter images [18, 19]. Hedbors [12] introduced a BA method which provides accurate structure and motion from a rolling shutter video stream by using a continuous time parametrization of the camera pose. It is however limited to continuous image streams with small baselines and does not work on sparse images with wide baselines. Moreover, none of the existing works showed a full pipeline that does full sparse to dense 3D reconstruction from rolling shutter images with wide baselines.

In this paper, we propose and implement a pipeline for sparse to dense 3D construction with wide baseline images captured from a fast moving rolling shutter camera. Fig. 1 shows an example of the sparse and dense 3D reconstructions obtained from our proposed pipeline. Wide baseline images are captured at a low frame rate of 4Hz by a rolling shutter camera mounted on a moving car, driven at 17km/h.

Specifically, we propose a cost function for BA that models the rolling shutter effect, incorporates GPS/INS readings, and enforces pairwise smoothness between neighboring poses. We optimize over the 3D structures, camera poses and velocities. In contrast to [15] that minimizes the absolute difference of the GPS/INS and camera poses, our smoothness term minimizes the difference between neighboring camera relative poses and their corresponding neighboring GPS/INS relative poses. As a result, our BA is able to compensate for the drifts that are accumulated in large scenes from the weak visual connectivity of wide baseline images without the risk of causing discontinuities in the poses.

The optimized camera poses are used to compute dense motion stereo. We adopt the plane sweep algorithm for rolling shutter camera proposed by [26]. We show that we can achieve a $7\times$ speed up in the depth map computation without losing accuracy with our novel interpolation scheme for the rolling shutter plane sweep stereo algorithm. We evaluate and show the feasibility of our approach on a 2.6km sequence, and compare our sparse and dense 3D reconstructions to those obtained from global shutter reconstructions.

Our contributions can be summarized as follow:

- Propose and show a working pipeline for full sparse to dense 3D reconstruction from large scale wide baseline rolling shutter images.
- New cost function for BA that models the rolling shutter effect, incorporates GPS/INS readings, and enforces pairwise smoothness between neighboring poses.
- Achieved $7\times$ speed up in depth map computation without losing accuracy with our novel interpolation scheme on the 3D planes for the rolling shutter plane sweep stereo algorithm.

2. Related Work

In the recent years, an increasing number of 3D vision algorithms originally designed for global shutter cameras are reformulated to include the rolling shutter camera model. These algorithms cover many aspects of 3D vision from camera calibration, pose estimation, bundle adjustment to dense motion stereo etc. However, none of these existing works showed a full sparse to dense 3D reconstruction from wide baseline rolling shutter images.

The authors of [10, 23] suggested algorithms for the calibration of rolling shutter timing. Others [17, 25, 8, 11, 5] have looked into rolling shutter wobble corrections using additional sensors such as gyroscopes or assuming that a rolling shutter wobble is induced by only a pure rotational motion. [27, 4] addressed the rolling shutter pose estimation problem with the minimal solutions from different variants of rolling shutter camera model. Meilland $et\ al.$ [22] proposed a RGB-D SfM algorithm which simultaneously solves for motion blur and rolling shutter deformations. In [21], the authors proposed an iterative algorithm to solve for the camera pose and velocity.

A rolling shutter camera bundle adjustment for a continuous stream of small baseline images was shown in [12]. They reformulate the bundle adjustment problem such that it requires 6 additional parameters compared to the global shutter version. These 6 additional parameters are due to the exploitation of the fact that the pose of the last scanline corresponds to the pose of the first scanline in the next frame. In [14], the authors modelled a rolling shutter camera rig as a generalized camera, where relative poses between cameras are obtained from a GPS/INS system. In the bundle adjustment, they optimized for one global camera rig pose, while using a rolling shutter aware reprojection error.

Ait-Aider $et\ al.$ [3] proposed a stereo algorithm for a rolling shutter stereo rig. Given a set of point correspondences, they recover the object pose and velocity by optimizing a non-linear system of equations. A challenge of finding pixel correspondences between image pairs arises in the monocular setup. While this is done by searching along the epipolar line for global shutter stereo, the search becomes difficult for the rolling shutter stereo where the pixel correspondence lies on an epipolar curve. The curvature of the epipolar curve depends on the motion and lens distortion of both cameras. In [26], the authors addressed this problem with a monocular rolling shutter plane sweep stereo algorithm. The main drawback of the algorithm is that it is computationally expensive to solve a high order polynomial for each pixel at various depths. We built upon their findings and proposed a novel algorithm which is $7\times$ faster in speed and provides similar accuracy in depth estimation.

3. Reconstruction Pipeline

Fig. 2 illustrates our proposed pipeline for sparse to dense 3D reconstruction from rolling shutter images. Our proposed reconstruction pipeline follows the standard SfM approach [24] with modifications made for wide baseline rolling shutter images. In detail:

Data Acquisition: Images are captured with a rolling shutter camera rig mounted on a car. The rig consists of 15 cameras arranged such that they cover over 80% of a sphere. In this work, we consider only 8 cameras which point sideways to the driving direction. Images from consecutive frames have wide baseline because our cameras are recording at a low frame rate of 4Hz. Each image is also synchronized with a GPS/INS position.

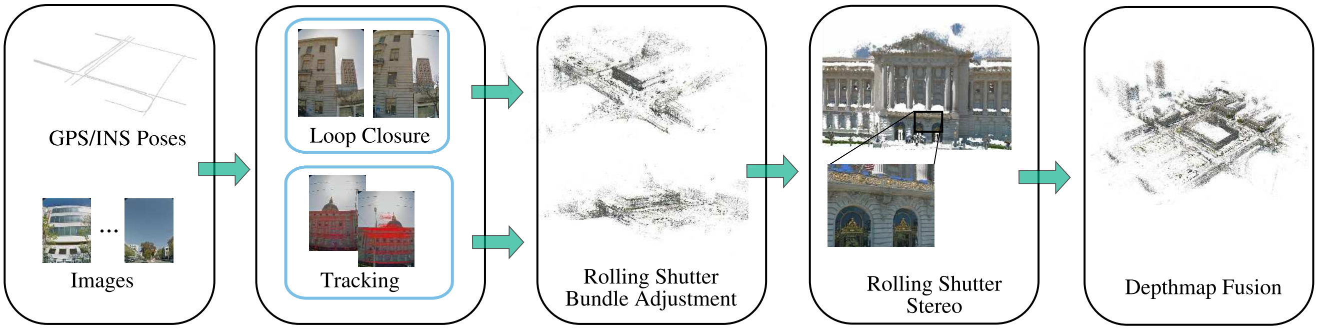


Figure 2. Overview of the reconstruction pipeline.

Tracking: We start by extracting SIFT [20] features using [30] on the radially distorted images. Features are then matched between neighboring images using a GPU brute-force matcher creating tracks between consecutive frames. Frames which show little parallax ($<$ 50 pixels) are removed from the tracks.

Loop closure: Loop closures are detected by comparing the GPS/INS poses. Images that are within a radius of 15m are considered as potential loop closures. Potential loop closures are verified geometrically using a rolling shutter pose estimation algorithm similar to [27]. We consider a loop closure to valid if it is within the vicinity of their original GPS/INS pose (within 15m) and there is enough inliers ($>$ 100) from the geometric verification. Each detected loop closure is added to BA in the form of a pose graph as an additional constraint (see Section 4 for more details).

Rolling Shutter BA: Tracks are first radially undistorted with the standard radial/tangential distortion model proposed by Brown [6]. Next, the undistorted keypoints are triangulated with the camera poses provided by the GPS/INS system. It is important to note that the camera poses provided by the GPS/INS system are not perfect due to systematic errors in calibration and multi-path problems in the urban environment. We use the GPS/INS system for a rough pose estimate of each scanline in the image. We then use our proposed rolling shutter aware BA to optimize over the 3D points, and camera extrinsics i.e. poses and velocities, while enforcing smoothness constraints between neighboring poses in the pose graph. The BA refinement process is further discussed in Section 4.

Rolling Shutter Stereo: The refined poses are then used to compute a dense 3D model using a multi-view and multi-resolution rolling shutter plane sweeping stereo algorithm similar to [26]. Here, we show that our novel interpolation scheme on the 3D planes for the rolling shutter plane sweep stereo algorithm allows us to achieve a 7$\times$ speed up in the depth map computations for dense reconstruction without losing accuracy. The depth maps are regularized using Semi-Global matching [13]. More details are given in Section 5.

Depth map Fusion: Finally, all 3D models are merged into a single coordinate frame. Only depth values which obtained support from at least 3 or more views are kept and used to render a dense point cloud.

4. Rolling Shutter Bundle Adjustment (BA)

Traditional BA [29] that minimizes the total reprojection errors is formulated as:

$$\underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum_m \sum_n ||\mathbf{x}_{m,n} - \mathbf{K}\cdot\mathbf{D}(\pi(\mathbf{P}_m, \mathbf{X}_n))||^2, \quad (1)$$

where $\mathbf{K}$ is the camera intrinsics parameter, $\mathbf{D}$ is the radial distortion function, $\mathbf{P} = [\mathbf{R}, -\mathbf{R}\mathbf{t}]$ is the camera extrinsics, i.e. rotation and translation, $\mathbf{x}_{m,n}$ is the feature point corresponding to the 3D point $\mathbf{X}_n$ observed by the camera m, and $\pi(.) : \mathbb{P}^3 \rightarrow \mathbb{P}^2$ denotes the projection function. This formulation assumes a global shutter camera model. For a moving rolling shutter camera, each scanline gets exposed at a different place in space along the motion trajectory. As a result, Eq. 1 no longer holds.

Similar to [10, 4] we propose to use a constant translational and rotational velocity parametrization for the camera pose:

$$\mathbf{t}(\tau) = \mathbf{t}_0 + \mathbf{v}\tau, \quad (2a)$$
$$\mathbf{R}(\tau) = \mathbf{exp}(\Omega\tau)\mathbf{R}_0, \quad (2b)$$

where $\mathbf{v}$ denotes the translational velocity, Ω denotes the angular velocity and τ the time of exposure (time delay between first an current scanline). $\mathbf{R}_0$ and $\mathbf{t}_0$ are the camera pose at the first 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. The pose of a given scanline exposed at time τ can be linearly interpolated from Eq. 2, which yields the following parametrized transformation matrix:

$$\mathbf{P}(\tau) = \begin{bmatrix}\mathbf{R}(\tau) & -\mathbf{R}(\tau)\mathbf{t}(\tau)\end{bmatrix}. \quad (3)$$

Given the continuous time pose parametrization in Eq. 3, we can rewrite the global shutter reprojection error in Eq. 1 as:

$$\underset{\mathbf{v},\mathbf{\Omega},\mathbf{R}_0,\mathbf{t}_0,\mathbf{X}}{\operatorname{argmin}} \sum_m \sum_n ||\mathbf{x}_{m,n} - \mathbf{K} \cdot \mathbf{D}(\pi(\mathbf{P}_m(\tau) \cdot \mathbf{X}_n))||^2, \tag{4}$$

where we optimize over the camera pose at the first scanline of each image ($\mathbf{R}_0$, $\mathbf{t}_0$), velocities ($\mathbf{v}$, $\mathbf{\Omega}$) and 3D structures $\mathbf{X}$. Unfortunately, the optimization based on Eq. 4 often breaks for our wide baseline images with weak visual connectivity. This problem is made worst at the end of façades in the scene where feature tracks drop tremendously. To overcome this problem, we introduce an additional smoothness term that enforces pairwise smoothness between neighboring poses:

$$\sum_{(i,j)\in\mathcal{G}} ||\mathbf{P}_i^{-1}(0) \cdot \mathbf{P}_j(0) - \mathbf{M}_{i,j}||^2. \tag{5}$$

$\mathbf{P}_i(0)$ is the i^{th} camera extrinsics parameters at the first scanline we optimize over, and represented as a 6 dimensional vector $[\log(\mathbf{R}_0)^\top, \mathbf{t}_0^\top]$ where $\log(.) : SO(3) \rightarrow so(3)$. $(i,j) \in \mathcal{G}$ denotes all pairs of neighboring poses in the pose graph $\mathcal{G}$. Here, $j = i+1$ denotes consecutive neighboring poses, and $j \neq i+1$ denotes loop closure neighboring poses mentioned in Section 3. For consecutive neighboring poses, i.e. $j = i+1$, $\mathbf{M}_{i,j}$ is the corresponding relative pose obtained from the GPS/INS reading:

$$\mathbf{M}_{i,j} = \tilde{\mathbf{P}}_i^{-1}(0)\tilde{\mathbf{P}}_j(0), \tag{6}$$

where $\tilde{\mathbf{P}}_i(0)$ and $\tilde{\mathbf{P}}_j(0)$ are the absolute poses from the GPS/INS at the first image scanline. It becomes obvious now that Eq. 6 penalizes deviation from the GPS/INS poses and preserves smoothness between neighboring poses. For non-consecutive neighboring poses, i.e. $j \neq i+1$, $\mathbf{M}_{i,j}$ is the relative pose computed from the pose estimation during the geometric verification in the loop closure mentioned in Section 3. Consequently, our BA formulation is also capable of closing large loop closure errors.

Our final objective function then writes as follows:

$$\underset{\mathbf{v},\mathbf{\Omega},\mathbf{R}_0,\mathbf{t}_0,\mathbf{X}}{\operatorname{argmin}} \Bigg\{ \sum_m \sum_n ||\mathbf{x}_{m,n} - \mathbf{K} \cdot \mathbf{D}(\pi(\mathbf{P}_m(\tau) \cdot \mathbf{X}_n))||^2 + \lambda \sum_{(i,j)\in\mathcal{G}} ||\mathbf{P}_i^{-1}(0) \cdot \mathbf{P}_j(0) - \mathbf{M}_{i,j}||^2 \Bigg\}, \tag{7}$$

where λ is a weighting factor we estimated empirically and set to 1e6 in all our experiments. We proposed an alternating optimization approach to minimize the cost function in Eq. 7. First, we optimize over the parameters $\mathbf{R}_0, \mathbf{t}_0, \mathbf{X}$ while keeping the parameters $\mathbf{v}, \mathbf{\Omega}$ fixed. Next, we optimize over $\mathbf{v}, \mathbf{\Omega}$ while keeping $\mathbf{R}_0, \mathbf{t}_0, \mathbf{X}$ fixed. We alternate between the two configurations until we reach convergence. In all our experiments, the solution converged after a maximum of two iterations, and in most cases one iteration is enough for convergence. We use the Ceres [2] optimization framework to solve for the non-linear least squares problem.

5. Time Continuous Rolling Shutter Stereo

We adopt and improve upon the plane sweep stereo algorithm proposed in [26] for rolling shutter cameras. The main difference between the plane sweep algorithm for global and rolling shutter cameras is the way that a pixel gets warped from a target image into a reference image for the evaluation of the photo-consistency cost. This can be done easily with homography for the global shutter cameras. Unfortunately, simple homography is not applicable for the rolling shutter cameras since each scanline has a different pose and therefore the plane parameters vary according to the scanlines. [26] incorporates the rolling shutter camera model into the plane sweep algorithm to achieve the warping of a pixel from a target to reference image. This results in the need to solve for the time of exposure τ that corresponds to the time (scanline) a given 3D point is imaged by the sensor. More precisely, the time of exposure τ is obtained by solving the following fix-point function:

$$s \cdot \mathbf{K} \cdot \mathbf{D}(\pi(\mathbf{P}(\tau) \cdot X)) = \tau, \tag{8}$$

where $\mathbf{K}$ denotes the camera intrinsics matrix, $\mathbf{D}$ is the lens distortion function, $\mathbf{P}(\tau)$ is the camera extrinsic at time τ and $\mathbf{X}$ is the 3D point and $\pi(.) : \mathbb{P}^3 \rightarrow \mathbb{P}^2$ denotes the projection function. s is the scanline selection operator that selects either the top or bottom row of the left hand side of Eq. 8, i.e. we set $s = [1, 0]$ for a shutter that moves horizontally, and $s = [0, 1]$ for a shutter that moves vertically. Note that solving for τ in Eq. 8 requires solving for the roots of a high order polynomial, where the order depends on the lens distortion model and motion parameterization. Using the motion parametrization presented in Section 4, we get a 9^{th} order polynomial in τ, where τ is solved using the Gauss-Newton minimization. The need to solve a 9^{th} order polynomial for every pixel in the image quickly becomes computationally expensive. Authors of [26] reported a processing time of 27ms per image with the Graphics Processing Unit (GPU). In addition, they proposed an alternative approach that first solves the τ values for a subset of pixels on the target image, and next obtain the τ values for all other pixels from interpolations of the subset of pixels with known τ that are warped onto the reference image. However, this technique comes with the cost of interpolation artefacts. Here we propose an alternative interpolation scheme where we process every pixel on the target image, but solve τ for a subset of the plane intersections (each plane corresponds to the depth that is searched for pixel correspondence in the plane sweep stereo algorithm) with each back-projected ray from every pixel on the target image. We solve for all the τ values by interpolating between the τ values solved from the subset of plane intersections projected onto the reference image. Fig. 4 shows how $\tau(d)$ changes with increasing plane depth d over a range of

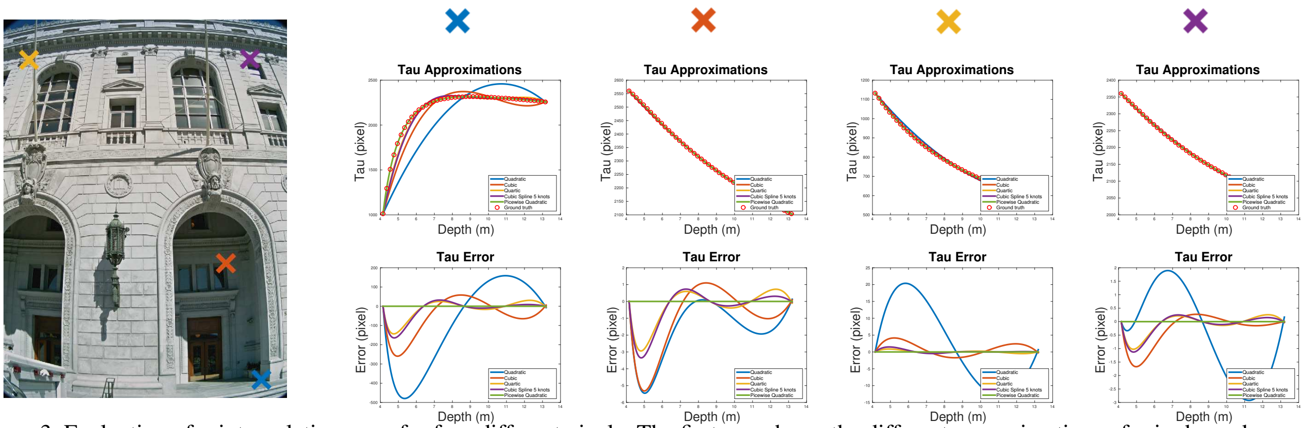


Figure 3. Evaluation of τ interpolation error for four different pixels. The first row shows the different approximations of τ in dependence of the depth d, i.e., $\tau(d)$. The second row shows the error to the ground truth. Only piecewise quadratic interpolation gives an error below 1e-3 pixel, while all other interpolation - quadratic, cubic, quartic and cubic spline give at least 1.0 pixel error.

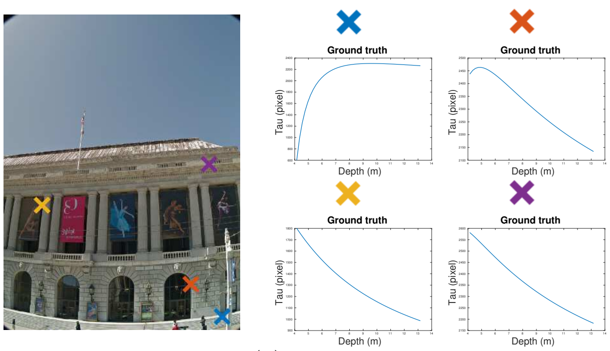


Figure 4. Representation of $\tau(d)$ at different pixel locations over a plane depth range of 9m, $d \in [4m, 13m]$.

4m to 9m. It should be noted that the closer the intersection point of the ray with the plane gets to the camera, the more rapidly (can be increasing or decreasing depending on the pixel location and camera motion) the time of exposure τ (scanline) changes with increasing velocity.

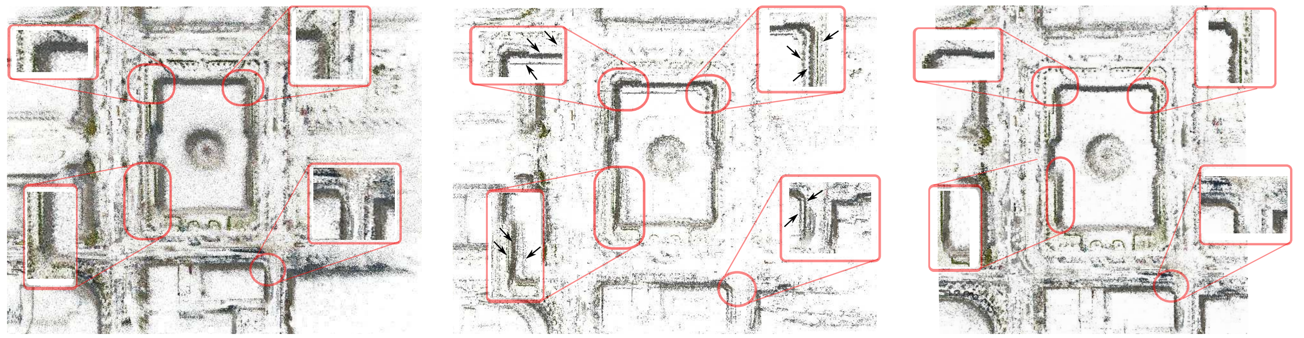
Finding a suitable parametrization for the interpolation of τ can be challenging since the curve $\tau(d)$ varies a lot with different pixel location, camera motion and radial distortion coefficients. We noted that the depth in which we search for pixel correspondences in the plane sweep stereo is bounded by the two planes, i.e. closest and furthest away from the camera. Exploiting this fact, we can evaluate the τ values for a subset of depths for each pixel and interpolate the missing intermediate τ values. We experimented with different interpolation schemes - quadratic, cubic, quartic, cubic spline and piecewise quadratic. Fig. 3 shows the errors for the different interpolation schemes. We can see that the error obtained from piecewise quadratic interpolation is the lowest at $\leq 1e-3$ pixel. All the other interpolation schemes give errors that are > 1 pixel, which means that the estimated scanline is off by at least one scanline. We compare the results from the interpolation schemes against a ground truth obtained by solving Eq. 8 with a Gauss-Newton scheme. Since the rolling shutter effect is dependent on the scene depth, the curvature of τ becomes much higher when the plane is closer to the camera, and becomes almost linear when the plane is located further away from the camera. This motivates us to use an adaptive interpolation range in which a quadratic function is fitted in dependence of the relative scene depth. Let c ($\sqrt{3}$ in our experiments) be the exponential factor and b be the initial interpolation range. The adaptive depth range d_i used for the piecewise depth interpolation is obtained using $d_i = c^i b$. Given the total number of planes to sweep w, the number of times τ needs to be fully evaluated for each pixel then is: $i = log(w/b)$. In addition, we can sparsely evaluate τ in the image space and bi-linearly interpolate the in between values. Combining those two interpolation schemes gives us an overall speedup of $6.56\times$, i.e. $\sim 7\times$.

6. Results

We evaluate our proposed pipeline on both synthetic and real datasets. We use the "Castle" and "Old Town" datasets provided by [26] for the synthetic experiments. The real dataset was captured with a car mounted with a rig of 15 rolling shutter cameras, covering 80% of a sphere. The trajectory has a total length of 2.6km with 17k images. Images have a resolution of 1944×2592 pixels and are sparsely recorded at 4Hz. The shutter scan time for each camera is 72ms. On average the car drives at 17km/h, which results in an average camera motion of 0.34m during image formation, meaning the distance between the first and last scanline is 0.34m apart.

6.1. Bundle Adjustment

Fig. 5 shows a comparison of the sparse 3D reconstructions (a) without optimization, (b) with global shutter model and (c) with our proposed BA cost. It can be seen that the 3D structures reconstructed from GPS/INS readings with-



(a) (b) (c)

Figure 5. Sparse 3D reconstructions: (a) Initial model obtained using GPS/INS poses, where the 3D points are very noisy with typical misalignment of 0.2-0.3m. (b) After global shutter BA with misaligned façades (arrows). (c) After rolling shutter BA (proposed method) with sharp and well aligned façades. Note that only points with $\leq$ 5 pixels reprojection error are shown in (b) and (c). The respective reprojection error distribution for the reconstructions are shown in Figure 6.

out BA is noisy. Although the 3D structures appear sharper after global shutter refinement, it is obvious that the method produces misaligned façades. In contrast, our proposed rolling shutter refinement produces a sharp and consistent 3D sparse model. Quantitatively, the histograms in Fig. 6 shows the count of reprojection error for all $1,371,294$ 3D points. Fig. 6 (center) shows that the global shutter BA has a very high count of reprojection errors $\geq$ 5 pixels, while the reprojection errors from our method is significantly lower. Fig. 7 shows a comparison of the "zoomed-in view" of the camera poses from GPS/INS readings (blue), our rolling shutter BA with pairwise smoothness (green), and global shutter BA without pairwise smoothness (red). It can be clearly seen that the wide baselines between the camera poses cause breakages in the global shutter BA without pairwise smoothness (red). On the other hand, our method (green) produces a smooth trajectory of camera poses.

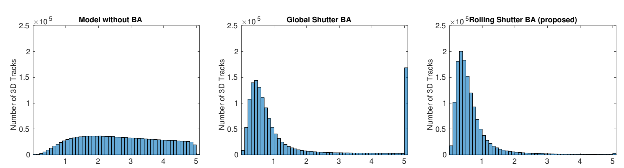


Figure 6. Reprojection error distributions for all $1,371,294$ 3D points from the reconstruction in Figure 5. Note that the last bin is extended to infinity.

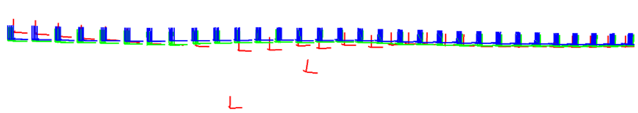


Figure 7. A close-up comparison of camera poses from initial GPS/INS readings (blue), our Rolling Shutter BA with pairwise smoothness (green), and Global Shutter BA (red).

6.2. Stereo

For our experiments, we use our proposed plane sweep algorithm with a single reference plane normal. The reference plane is obtained by finding the dominant plane using RANSAC [7] on the sparse point cloud obtained from SfM. A sweep through 3D space is performed within the distance of $[d_{front}, d_{back}]$ around the reference plane. We set $d_{front} = 5m$ and $d_{back} = 3m$. Planes in between are sampled linearly in image space. A pixel transfer between two images is computed by first undistorting a pixel and intersecting the resulting ray with the corresponding plane. The intersection point is then back-projected onto the other view for texture lookup. The correlation between two images is computed using Normalized Cross-Correlation (NCC), with a window size of 5×5 pixels. We make use of a multi-resolution approach, where we aggregate the correlation cost over multiple pyramid levels (3 levels in our case) to be more robust towards textureless surfaces. Furthermore we use a multi-view approach to handle occlusions. At each depth (plane), we consider the $k = 3$ best views out of $n = 6$ that provide the highest correlation. Once the cost volume is obtained, we regularize it with Semi-Global-Matching [13] using 16 different path directions. We fit a quadratic through the depth that provides the lowest cost to obtain the final depth. Lastly, all estimated 3D points undergo a geometric verification, where points that give a consistent depth within 3 or more views are considered to be valid. We use a threshold of 0.1m for our consistency check.

Synthetic Data: Tab. 1 shows the evaluation of our algorithm on synthetic data. It can be seen that the performance of our proposed piecewise quadratic interpolation has accuracies (median error and fill rate) that are comparable to the RS method from [26] for both datasets, yet achieving $3.7\times$ speedup. We can further see that by combining bilinear and piecewise quadratic interpolations, we achieved a

Method	speed / warp [ms]	Castle median [m]	Castle MAD [m]	Castle fill rate	Old Town median [m]	Old Town MAD [m]	Old Town fill rate
FA2 [26]	2.2	1.02	1.02	52.8%	0.26	0.22	58.6%
Bilinear + PQI	4.2	0.05	0.049	75.6%	0.099	0.096	57.1%
Piecewise Quadratic	7.4	0.049	0.041	75.6%	0.098	0.096	57.2%
RS [26]	27.7	0.041	0.032	76.3%	0.085	0.077	62.0%

Table 1. Comparison of our methods - Piecewise Quadratic Interpolation (PQI) and Bilinear Interpolation with PQI with RS and FA2 on the "Castle" and "Old Town" synthetic datasets.

speedup of $\sim 7\times$ over RS without losing much accuracy. Fig. 8 shows a visualization of the voxelwise errors from the stereo reconstruction compared to ground truth for RS, FA2 [26] and ours. It is clear that FA2 has the highest errors (more red), and there is an insignificant reduction in accuracy of our method compared to RS.

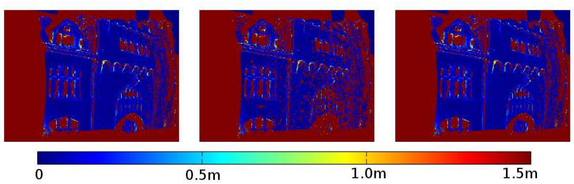


Figure 8. Visualization of the voxelwise errors from the stereo reconstruction compared to ground truth for RS [26], FA2 [26] and ours on the synthetic "Castle" dataset.

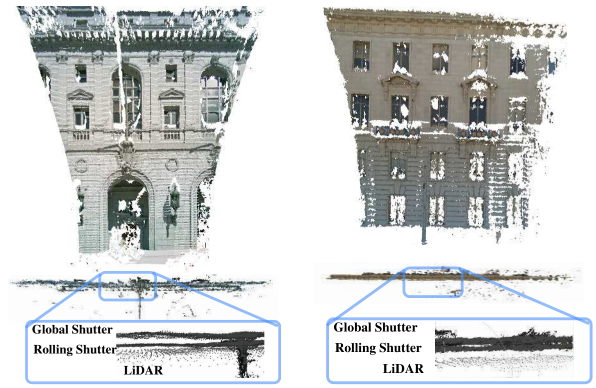
LiDAR: We also compare our reconstruction to sparse LiDAR data, which was captured alongside with the image data, in Fig. 9 and Tab. 2. We re-project the LiDAR data into the estimated depth map and evaluate the respective error of the estimated depth maps computed with our rolling shutter pipeline (rolling shutter BA and stereo) and the global shutter pipeline (global shutter BA and stereo). It should be noted that there might be some inaccuracies in this experiment because the offset between our LiDAR and camera is physically measured. Nonetheless, we observe that in general the global shutter reconstructions have median errors that are ~ 0.15m more than the rolling shutter reconstructions.

	Global Shutter (m)	Rolling Shutter (m)
Fig.9(a)	0.42	0.23
Fig.9(b)	0.37	0.25

Table 2. Median errors from global and rolling shutter depth maps, compared to sparse LiDAR data.

6.3. Full Pipeline

We show results of sparse to dense reconstruction in Fig. 10 with our proposed methods on a large scale dataset over 2.6km in length taken at San Francisco City Hall and its surroundings. First we use a global window BA with smoothness prior to refine the camera poses and 3D structure, as described in Section 4. A robust Huber loss function is used to handle outliers. Only tracks of size ≥ 3 are consider in the BA process. After BA, we remove points that have a reprojection error > 1 pixel. In the second stage, we run our rolling shutter aware motion stereo algorithm, as mentioned in Section 5. We adaptively evaluate τ in sweep space at a rate of $d_i = c^i b$ in all our experiments, where d_i denotes the depth interval, $c = 1.5$ and $b = 6$. In addition, τ is evaluated sparsely in the image at every 5^{th} pixel and bilinearly interpolated.



(a) (b)

Figure 9. Comparison of rolling shutter and global shutter stereo to LiDAR data.

7. Conclusion

We proposed a rolling shutter bundle adjustment, which optimizes over the 3D structure, the camera poses and velocities using an additional smoothness term to compensate for drift. In addition, we proposed a simple yet efficient interpolation scheme for rolling shutter stereo which speeds up the algorithm by $7\times$, while providing almost same accuracy as state-of-the art algorithms. We evaluated our pipeline on a camera trajectory of 2.6km length and show quantitative results of the sparse and dense reconstruction. In the future, we would like to fuse the terrestrial reconstruction with aerial reconstructions.

Acknowledgment

We thank the reviewers for their constructive comments. This work is partially supported by a Google award. The last author is funded by a start-up grant #R-265-000-548-133 from the Faculty of Engineering at the National University of Singapore.

Figure 10. Sample reconstructions of the San Francisco city hall and its surroundings.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, Oct. 2011. 1

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 4

[3] O. Ait-Aider and F. Berry. Structure and kinematics triangulation with a rolling shutter stereo rig. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1835–1840, 2009. 2

[4] C. Albl, Z. Kukelova, and T. Pajdla. R6p - rolling shutter absolute camera pose. June 2015. 2, 3

[5] J. B. Alexandre Karpenko, David E. Jacobs and M. Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. Technical Report CTSR 2011-03, Stanford University, 2011. 2

[6] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971. 3

[7] 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*, 24(6):381–395, June 1981. 6

[8] 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. 2

[9] 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 *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 368–381, Berlin, Heidelberg, 2010. Springer-Verlag. 1

[10] C. Geyer, M. Meingast, , and S. Sastry. Geometric models of rolling-shutter cameras. In *Proceedings of OMNIVIS*, 2005. 2, 3

[11] 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. 2

[12] 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. 1, 2

[13] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):328–341, Feb. 2008. 3, 6

[14] B. Klingner, D. Martin, and J. Roseborough. Street view motion-from-structure-from-motion. In *Proceedings of the International Conference on Computer Vision*, 2013. 2

[15] M. Lhuillier. Fusion of gps and structure-from-motion using constrained bundle adjustments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3025–3032. IEEE, 2011. 2

[16] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 427–440, Berlin, Heidelberg, 2008. Springer-Verlag. 1

[17] C.-K. Liang, L.-W. Chang, and H. Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330, Aug. 2

[18] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. *ACM Trans. Graph.*, 30(1):4:1–4:10, Feb. 2011. 1

[19] M. I. A. Lourakis and A. A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Softw.*, 36(1):2:1–2:30, Mar. 2009. 1

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal on Computer Vision (IJCV)*, volume 60, pages 91–110, 2004. 3

[21] 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 *Proceedings of the 12th European conference on Computer Vision - Volume Part I*, ECCV'12, pages 456–469, Berlin, Heidelberg, 2012. Springer. 2

[22] M. Meilland, T. Drummond, and A. I. Comport. A unified rolling shutter and motion blur model for 3d visual registration. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 2

[23] L. Oth, P. Furgale, L. Kneip, and R. Siegwart. Rolling shutter camera calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2

[24] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008. 2

[25] E. Ringaby and P.-E. Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision (IJCV)*, 96(3):335–352, 2012. 2

[26] O. Saurer, K. Koser, J.-Y. Bouguet, and M. Pollefeys. Rolling shutter stereo. In *IEEE International Conference on Computer Vision (ICCV)*, pages 465–472, 2013. 1, 2, 3, 4, 5, 6, 7

[27] O. Saurer, M. Pollefeys, and G. H. Lee. A minimal solution to the rolling shutter pose estimation problem. In *Intelligent Robots and Systems (IROS 2015), 2015 IEEE/RSJ International Conference on*, 2015. 2, 3

[28] G. Thalin. Camera rolling shutter amounts. http://www.guthspot.se/video/deshaker.htm. 1

[29] 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*, ICCV, pages 298–372. Springer, 2000. 1, 3

[30] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 3