Rolling Shutter Stereo

Olivier Saurer
ETH Zürich
Switzerland
saurero@inf.ethz.ch

Kevin Köser*
GEOMAR Kiel
Germany
kkoeser@geomar.de

Jean-Yves Bouguet
Google, Inc.
Mountain View, CA
jyb@google.com

Marc Pollefeys
ETH Zürich
Switzerland
pomarc@inf.ethz.ch

Abstract

A huge fraction of cameras used nowadays is based on CMOS sensors with a rolling shutter *that exposes the image line by line. For dynamic scenes/cameras this introduces undesired effects like stretch, shear and wobble. It has been shown earlier that rotational shake induced rolling shutter effects in hand-held cell phone capture can be compensated based on an estimate of the camera rotation. In contrast, we analyse the case of significant camera motion, e.g. where a bypassing streetlevel capture vehicle uses a rolling shutter camera in a 3D reconstruction framework. The introduced error is depth dependent and cannot be compensated based on camera motion/rotation alone, invalidating also rectification for stereo camera systems. On top, significant lens distortion as often present in wide angle cameras intertwines with rolling shutter effects as it changes the* time *at which a certain 3D point is seen. We show that naive 3D reconstructions (assuming global shutter) will deliver biased geometry already for very mild assumptions on vehicle speed and resolution. We then develop rolling shutter dense multiview stereo algorithms that solve for time of exposure and depth at the same time, even in the presence of lens distortion and perform an evaluation on ground truth laser scan models as well as on real street-level data.*

1. Introduction

Visual 3D reconstruction of objects, scenes or whole cities nowadays seems to be a well understood problem, building on techniques like structure from motion and dense depth estimation (see e.g. [18, 19]). However, results published so far usually assume classical CCD cameras that capture images in a way that all pixels of the same image are being exposed at the same time. This is however not true for most CMOS sensors, such as those built into nowadays' smart phones or many industrial cameras [9]. Consequently, the analysis of rolling shutter cameras came
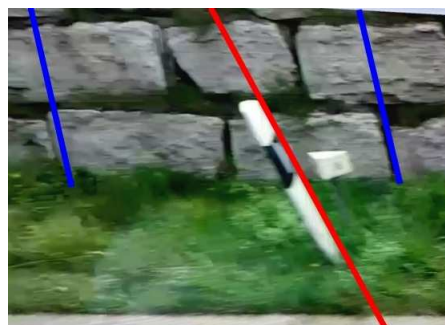
Figure 1. Depth-dependent rolling shutter effect: The pole in the front is vertical but due to fast horizontal camera motion during exposure appears to be slanted (red line). The vertical structures in the back are also slanted (blue lines) but much less, as the motion introduces less disparity to distant objects.

into focus, where exposure of columns (scanlines) happens in sequential order leading to undesired distortion effects when the camera is not fixed during exposure. It has been shown recently that for hand-held smartphone cameras in static scenes, most of the rolling shutter effects can be compensated in the image (without 3D scene information), that is by compensating rotation [6, 20, 12, 1]. However, in case a high resolution camera is mounted on a moderately fast driving capture vehicle, strong rolling shutter effects will be introduced by the motion of the camera, even if the camera orientation is stable (similarly at a smaller scale, for video-based reconstruction of objects using a cell phone). Unfortunately, these effects depend on the distance to the objects, such that closer 3D points will be much more distorted than those very far away (compare Fig. 1), making a simple 2D image warp into a global shutter image impossible. Also "standard stereo" rectification of image pairs (e.g. [7]) is in general not possible: Epipolar curve pairs, where each point on a curve in the left image maps to some point on a "corresponding" curve in the right image and vice versa, exist only in special configurations.

In this paper we analyze the rolling shutter stereo problem and develop fast multi-view stereo algorithms that produce accurate 3D models from rolling shutter cameras.

As real cameras often have lens distortion, and in particular those wide angle cameras often used for capturing streetlevel data, we also consider lens distortion, which we show makes the problem much more complex. To the best of our knowledge, no previous work exists on dense depth estimation with rolling shutter cameras and the common setting of lens distortion in a rolling shutter setting has not been analyzed. We therefore make the following novel contributions:

1. Practical discussion of fast-motion induced rolling shutter effects: Traditional stereo produces biased 3D results for standard streetlevel capture geometries

2. Analysis of interplay between rolling shutter and lens distortion: Correct undistortion requires 3D scene information.

3. Planar rolling shutter warp as a generalization of the plane induced homography

4. Multi-view stereo algorithm for rolling shutter cameras (with or without lens distortion)

In section 2 we will review previous work on rolling shutter cameras. We will then recapitulate the rolling shutter model and analyze fast motion and lens distortion effects in section 3. In section 4 we develop a warp for mapping a point of one rolling shutter image into another rolling shutter image, assuming a planar 3D scene. Based on this we then present both fast and accurate multi-view stereo algorithms in section 4.1. These are then evaluated quantitatively on textured laser scan models and qualitatively on real street-level data in section 5.

2. Previous work

The chip-level architecture of a CMOS sensor and the reasons for the rolling shutter effect are described by Liang et al. [17] who also propose an optic-flow-like method to compensate rolling shutter effects for in-plane motion. Earlier, in [9] Geyer et al. had analyzed the effect of a rolling shutter camera, in particular for special camera motions and geometries (e.g. fronto-parallel, no forward components) and had suggested a scheme how to calibrate the shutter timings. They showed that in a very special setting a rolling shutter sensor behaves as a x-slits camera [22]. For those, Feldman et al. had discussed epipolar geometry [5]. Rolling shutter cameras are also related to pushbroom camera models [11] often used for satellite images (actually a special case [22] of the x-slits cameras), however for those, under straight motion, backprojected planes are parallel, while for rolling shutter cameras this does not hold.

Recently, several approaches for image stabilization for rolling shutter cameras have been proposed. Here, Bradley et al. [3] use stroboscope lighting and subframe warping

to synchronize multiple rolling shutter cameras and to compensate the sequential exposure effects. Baker et al. [2] pose the rectification as a superresolution problem that can be solved using optical flow. Also Grundmann et al. [10] exploit local flow vectors to compensate rolling shutter for uncalibrated cameras, but using a mixture of homographies. In contrast, Hanning et al. [12] and Karpenko et al. [1] use gyroscopes of cell phones to compensate for rotational shake.

While the above approaches are rather 2D in nature, Forssen, Ringaby, Hedborg et al. applied structure from motion algorithms to tackle the problem for static scenes: First, Forssen and Ringaby [6, 20] had tracked features through cell phone video sequences and compensated cell phone rotation, which they identified as the dominant source of distortion for hand-held videos. In a later work, Hedborg et al. [13] have shown a full bundle adjustment including motion effects as well. Most recently Klingner et al. [16] proposed a structure from motion pipeline, for cameras mounted on a car, which uses relative pose prior along the vehicle path.

Our approach can be seen as the next step of a 3D reconstruction pipeline from rolling shutter cameras. Given camera motion and orientation (from bundle adjustment and/or sensors), our goal is to densely estimate the 3D scene geometry from rolling shutter cameras. In particular, and in contrast to the work on hand-held cell phones, we consider the case where the camera undergoes fast motion (e.g. on a capture vehicle, or in a cell phone close to an object) introducing a depth-dependent rolling shutter effect. On top we consider lens distortion, which cannot be pre-rectified as that would change the image coordinate and thus the time when the particular 3D points was seen by the camera. However, using a plane-sweep stereo approach (see e.g. [21] or [15] for non pinhole cameras), we show how to solve depth estimation, lens undistortion and rolling shutter compensation at the same time. The approach is intended for motion stereo, i.e. with a single camera, which is however valid as well for moving camera rigs.

3. Rolling Shutter Camera Model

Lets look at the case where a 3D point $\mathbf{X}$ is observed by a global shutter pinhole camera P, i.e. it is projected to image position $\mathbf{x}$ given a known camera calibration matrix K (without loss of generality we assume K to be the identity matrix in the remainder of the paper). In case of linear camera motion and constant orientation, the point moves on a straight line in the image, its position depending on the time τ:

$$\mathbf{x}_\tau \simeq \mathsf{P}_\tau \mathbf{X} = (R_0 \mid \boldsymbol{t}_0 + \tau \boldsymbol{t}) \, \mathbf{X} \qquad (1)$$

Now, we will move to the rolling shutter camera model, assuming that image column (scanline) r is exposed at time $\tau = mr + b$. For simplification of notation we assume
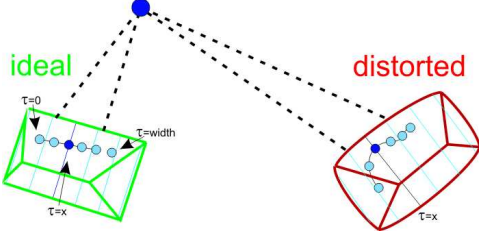
Figure 2. An ideal (green) and a distorted (red) camera observing a 3D point while moving straight. The projection of the point describes a straight line or a more complicated curve (depending on degree of distortion parameters). The distorted camera cannot be undistorted without depth information for the 3D point, as the time of exposure τ depends on the depth.

$m = 1$ and $b = 0$, however, for a real system these coefficients need to be calibrated [9] and considered. In order to find out *when* $\mathbf{X}$ will be seen by the camera, we have to check, at which moment it is projected to an active scanline. For this we have to compute the x-coordinate (the scanline) of the projection into the image and take $\boldsymbol{x}_\tau$ of Eq. 1 from projective space $\mathbb{P}^2$ to $\boldsymbol{x}_\tau$ into euclidean space $\mathbb{R}^2$

$$\boldsymbol{x}_\tau = \left(\begin{array}{c} (c_1\tau + c_2)/(c_5\tau + c_6) \\ (c_3\tau + c_4)/(c_5\tau + c_6) \end{array} \right) \qquad (2)$$

for some coefficients c_i depending on calibration, pose and 3D point. Then we look at the scanline (horizontal coordinate) that must match the time of exposure τ

$$\text{scanline}_{\mathbf{X},\mathsf{P}}(\tau) = (c_1\tau + c_2)/(c_5\tau + c_6) \overset{!}{=} \tau \qquad (3)$$

Essentially, we are looking for the fixpoint of scanline(.) which leads to a quadratic equation in τ. The derivation was based on a straight simple motion model. In Tab. 1 we list a number of alternative parametric motion/camera models and the resulting degree of the τ polynomial (considering extra rotational or translational offsets is possible and will add more freedom to the motion patterns but will not change the degree of the polynomial). For each of those, to project a 3D point a polynomial in τ has to be solved to figure out whether the point is seen on the scanline that is currently exposed.

Many lenses, in particular wide angle lenses, show a significant amount of distortion and in the following we will briefly re-derive a standard radial/tangential distortion model that dates back to Brown [4]:

$$\boldsymbol{x}'_\tau = (1 + r_2 r^2 + r_4 r^4 + r_6 r^6)\boldsymbol{x}_\tau + \boldsymbol{dx} \qquad (4)$$

with $\boldsymbol{x}_\tau = (x, y)^\mathsf{T}$ being the (undistorted) offset vector from the distortion center[1], $r = \|\boldsymbol{x}_\tau\|$, $\boldsymbol{x}'_\tau$ being the offset in the distorted image and

$$\boldsymbol{dx} = \left(\begin{array}{c} 2t_a xy + t_b(r^2 + 2x^2) \\ t_a(r^2 + 2y^2) + 2t_b xy \end{array} \right). \qquad (5)$$

[1]For simplicity of notation, we assume the distortion center at $(0,0)^\mathsf{T}$

Motion	Orient.	Dist.	Pose P_τ	deg.
linear	const	no	$(\mathsf{I} \mid \tau\boldsymbol{t})$	2
orbital	linear	no	$\left(\mathsf{I} + \tau\,[\boldsymbol{r}]_\times \mid \boldsymbol{t}\right)$	2
spiral	linear	no	$\left(\mathsf{I} + \tau\,[\boldsymbol{r}]_\times \mid \tau\boldsymbol{t}\right)$	2
linear	linear	no	$\left(\mathsf{I} + \tau\,[\boldsymbol{r}]_\times \mid (\mathsf{I} + \tau\,[\boldsymbol{r}]_\times)\tau\boldsymbol{t}\right)$	3
linear	const	r_2	$(\mathsf{I} \mid \tau\boldsymbol{t})$	4
orbital	linear	r_2	$\left(\mathsf{I} + \tau\,[\boldsymbol{r}]_\times \mid \boldsymbol{t}\right)$	4
spiral	linear	r_2	$\left(\mathsf{I} + \tau\,[\boldsymbol{r}]_\times \mid \tau\boldsymbol{t}\right)$	4
linear	linear	r_2	$\left(\mathsf{I} + \tau\,[\boldsymbol{r}]_\times \mid (\mathsf{I} + \tau\,[\boldsymbol{r}]_\times)\tau\boldsymbol{t}\right)$	5
not const.	any	r_2, r_4 r_6 t_a, t_b		≥ 8

Table 1. Some common short term motions and the resulting polynomial degree when (not) considering distortion for obtaining the τ in a rolling shutter camera (see also [9]). Note that for very short term (intraframe) motion on fast driving cars we can assume constant orientation but that the more general models would not lead to a significantly more difficult problem.

Herein, r_2, r_4, r_6, t_a, t_b are the distortion coefficients. In case radial or tangential distortion is present in the image, the curve $\boldsymbol{x}'_\tau$ described by a point in the image even under straight camera motion becomes more complicated and the degree of Eq. 3 will increase (see Tab. 1 and Fig. 2). Note that when the lens distortion of such a rolling shutter image is compensated (classical global shutter like inversion of Eq. 4), it means that straight lines in space will also become straight in the image, but that the shape of an original CMOS sensor scanline (those pixels that were exposed jointly at the same time) will become a more complicated curve rather than an image column. Consequently, the complexity is just shifted from the left hand side of Eq. 3 to the right hand side.

For short term motions (during exposure time of one image, which is usually a fraction of a second) of rolling shutter cameras the linear motion with no rotation (car driving straight) and the orbital motion (cell phone filming a handheld object) are the most important cases. The linear/linear case is somewhat more special and applies to panning cameras on a linear stage as those track-level "slow-motion" cameras used for the 100m sprints at the Olympic games. Also note that for the important cases the first (most significant) radial distortion coefficient can be considered for a closed form solution (polynomial degree up to 4).

3.1. Rolling Shutter Observability

Rolling shutter needs to be considered only when its effects are significant, i.e. for stereo in the range of one pixel or more. In the following we concretize the assessment of [9] with practical numbers and considerations to allow for a decision of whether a rolling shutter model makes sense for a particular capture configuration. We assume that a capture

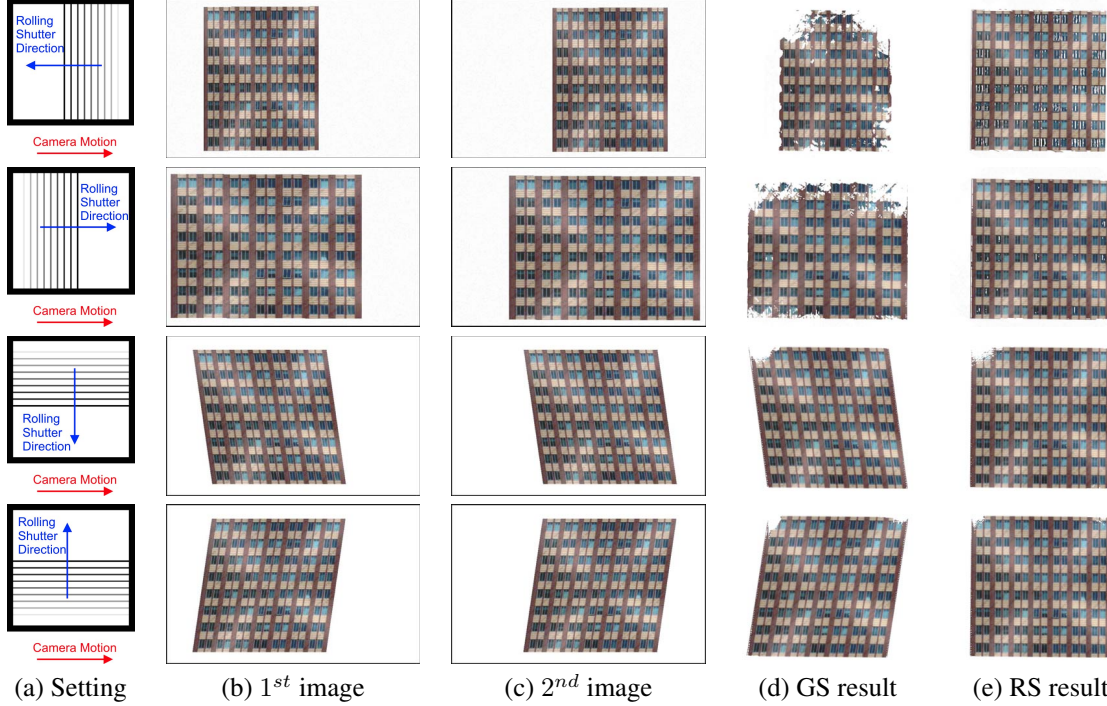|				
(a) Setting	(b) 1st image	(c) 2nd image	(d) GS result	(e) RS result

Figure 3. Reconstruction of a square building facade observed by a rolling shutter camera moving parallel to the facade from left to right, with different simulated rolling shutter directions. (a) shows rolling shutter direction relative to motion direction, resulting in some image (b) and later in another image (c). Column (d) shows the reconstruction when assuming the same pose for all pixels (classical global shutter model), where one can see horizontal stretch, horizontal compression or slant in the 3D model. Finally, column (e) shows the reconstruction using our formulation, taking into account the rolling shutter effect, which matches the grounds truth 3D model.

vehicles drives at a certain speed v (e.g. 25 km/h) and uses a camera with a certain field of view ϕ (e.g. 90 degrees) and image width w (e.g. 2000 pixels). We assume that all lines of an image have been exposed after t (e.g. 72 ms), i.e. there is a time difference of $t/2$ between the center scanline and a boundary scanline of an image. The camera position error compared to the center pose is $\Delta x = t/2 \cdot v$. The focal length can be computed as $f = \frac{w}{2}/tan(\frac{\phi}{2})$. Looking at some point on the optical axis of the camera (i.e. $(0 \ 0 \ z)^{\mathsf{T}}$ in camera coordinates), it will be projected to the principal point (in a global shutter camera model). If that camera now moves by some amount Δx depending on the speed defined above, then if we want at most 1 pixel displacement, the 3D point must be at least $z > f\Delta x$ away, i.e.

$$z > \frac{\frac{w}{2}}{tan(\frac{\phi}{2})} \cdot \frac{t}{2} \cdot v \qquad (6)$$

which, inserting the above values and approximating $tan(x) \approx x \cdot tan(45°)$, leads to the rule of thumb

$$z_{\min} \approx 6.25m \frac{w_{[\text{pixel}]}}{\phi_{[degree]}} \cdot t_{[\text{sec}]} \cdot v_{[km/h]} = 250m \qquad (7)$$

That is, for ten times less resolution or ten times less speed there is still at least one pixel error up to $25m$ distance but

neither such speeds nor such resolution is any useful. Because *locally* these errors are not as visible (between neighboring scanlines the error is 1000 times smaller, as there are thousand scanlines between the center and the border) they might seem to be of minor importance, however as can be seen above *for accurate reconstruction from a driving car they are significant.*

Vertical or Horizontal Rolling Shutter? For vehicle mounted cameras there are several considerations for how to mount the camera, such as different field of view in x and y direction, mounting space with respect to other cameras, full dome coverage and so on. Besides those, the direction of the rolling shutter plays an important role. For camera planes parallel with the facades and rolling shutter orthogonal to the motion direction, a shearing effect will be visible in each image (maybe less visually pleasant when displayed as raw image) and such images do not align well with Manhattan structures in the scene. On the other hand, when the rolling shutter is parallel with the motion, the image will be shrunk or stretched in that direction and for certain driving speeds undersampling issues may appear. The resulting images and qualitative effects when observing a plane with different shutter directions can be seen in Fig. 3.

Independently of the direction of the rolling shutter, we

will develop a depth dependent rolling shutter image warp in the next section, that can warp one rolling shutter image into another one taken from somewhere else, assuming some scene plane Π, similar in spirit to a plane-induced homography (with the goal of enabling plane sweep stereo).

4. Rolling Shutter Warp Across a Plane

To warp a point from one rolling shutter camera to another, we first backproject it to a plane Π and then project it into the other image.

RS Backprojection of pixel onto space plane Π: Given some pixel position $\mathbf{p} \in \mathbb{P}^2$ in a rolling shutter image, from its scanline we know immediately the time of exposure τ_p and consequently the corresponding projection matrix P_{τ_p}. Consequently, we can choose a 3D point $\mathbf{X}_i \in \mathbb{P}^3$ on the ray through the camera center $\mathbf{C}_{\tau_p} \in \mathbb{P}^3$ that projects to $\mathbf{p}$. All points $\mathbf{L}_i$ on that ray can be represented as

$$\mathbf{L}_i = \mathbf{C}_{\tau_p} + \lambda \mathbf{X}_i, \qquad \lambda \in \mathbb{R} \tag{8}$$

All points $\mathbf{X}$ that lie on the plane $\Pi \in \mathbb{P}^3$ fulfill

$$\Pi^\mathsf{T}\mathbf{X} = 0, \quad \text{where} \quad \Pi = (\boldsymbol{n}_\Pi \;\; -d)^\mathsf{T}, \tag{9}$$

and substituting Eq. 8 into Eq. 9 we arrive at a linear equation in λ

$$\Pi^\mathsf{T}(\mathbf{C}_i + \lambda \mathbf{X}_i) = 0, \tag{10}$$

that allows to find the 3D intersection $\mathbf{X}_\Pi$ of the plane Π and the backprojected ray.

RS Projection of plane point into other view: The time of exposure of a certain 3D point can be computed according to Eq. 3, that is quadratic in τ or, in case of distortion, using $\boldsymbol{x}'_\tau$ from Eq. 4 substituted for $\boldsymbol{x}_\tau$ in Eq. 3:

$$\frac{\alpha_7 \tau_q^7 + \alpha_6 \tau_q^6 + \cdots + \alpha_0}{\beta_7 \tau_q^7 + \beta_6 \tau_q^6 + \cdots + \beta_0} = \tau \tag{11}$$

that can be rewritten as

$$\gamma_8 \tau_q^8 + \gamma_7 \tau_q^7 + \cdots + \gamma_0 = 0, \tag{12}$$

for some $\alpha_j, \beta_k, \gamma_l \in \mathbb{R}$. The degree of the polynomial depends on the lens distortion and the motion model as can be seen in Tab. 1. Up to fourth order, i.e. using only the first radial distortion coefficient and one of the important motion models, this can be solved in closed form for the time of exposure τ_q in the other image. Only τ_qs are valid that lie in the exposure time interval, in our case $[0; \text{width} - 1]$. In the rare case that more than one solution fulfills this, the same 3D point is seen multiple times in the same image (remember the rolling shutter creates a multi-perspective image when the camera moves). If we just want to find the

color of the point (as is the case for our warp), all solutions are valid and we simply choose the earliest time of exposure τ_q. Given τ_q, we know the camera pose and we can project the 3D point on the plane to finally obtain the image coordinates $\mathbf{q}$ in the other image, which completes the warp.

For the full radial/tangential distortion model according to Eq. 4, we obtain a polynomial that cannot be solved in closed form. Consequently, we perform gradient descent on Eq. 12, initialized with $\tau_0 = \text{width}/2$.

Although the previously described warp can be fully parallelized and runs on the GPU it is computationally expensive since it requires to solve for the time τ for each pixel individually. We suggest - and later evaluate - two approximate strategies, that promise a speedup at the cost of some accuracy:

Fast approximation 1 (FA1): global shutter lens undistortion An efficient approximation is to perform the expensive lens undistortion globally, and then solve for the quadratic Eq. 3 in closed form. This is in particular useful when the lens distortion is minor, because then the time of exposure does not change much with or without distortion. In this case the undistortion can be precomputed offline using a lookup table (as standard for global shutter undistortion) and has to be done only once per image (if warps across multiple planes are run as in plane sweeping there is no need to run it per plane).

Fast approximation 2 (FA2): coarse grid computation of warp's texture coordinates Alternatively, rather than computing τ and then the resulting texture coordinate for each pixel, we propose to evaluate the texture coordinates in dependence of τ on a coarser grid (e.g. only every 10 pixels) which is then used to compute the actual texture lookup coordinates using texture interpolation. This approach (FA 2) can exploit highly optimized GPU texture handling.

The speedup of the approaches above is given in Tab. 2. The timings are evaluated on a GeForce GTX 680 graphics processing unit.

4.1. Integration to Plane Sweep Stereo

For global shutter cameras, it has been shown that the ability of the graphics processing unit to handle smooth warps [21] can be exploited for real-time stereo approaches. Having understood under what speeds, resolutions and distances a rolling shutter camera model must be used we can exploit the warps of the previous section in a plane sweep approach where we hypothesize a scene plane, warp our image across that scene plane into another reference view and determine the agreement of those images for each pixel. This is repeated for a number of planes to obtain a whole cost volume, i.e. we obtain costs (dissimilarity) for each

plane hypothesis at each pixel position of the reference view. In order to robustify the approach with respect to partial occlusions, we generate the cost volume in the reference view from n neighboring views and for each plane at each pixel consider only the mean of the k best correlation costs out of the n neighbouring views.

On that cost volume, smoothness terms can be used and any suitable optimization technique to solve them. We follow smoothness terms and optimization strategy as proposed by Hirschmüller [14]. The result is a depth map for the reference view encoding a depth value for each pixel. Using also the image and the camera poses this can be used to generate a 3D model.

5. Evaluation

We evaluated the proposed rolling shutter stereo algorithm on both synthetic and real datasets that mimic a single rolling shutter camera mounted on a moving car (allowing motion stereo to be computed from consecutive images). All synthetic data are available on the project website [2].

Implementation Details: For the evaluations in this paper we stick to a simple plane sweep model with a single plane normal (e.g. obtained from dominant scene planes [8] in a prior sparse reconstruction step like [13] or [16]) and a single sweeping direction. The sweep is performed by creating additional planes within the distance range $[D_{min}, D_{max}]$. The planes are sampled approximately linearly in image space, such that a warp over two neighbouring planes results in a pixel displacement of maximum one pixel distance. A warp over a plane is computed by first undistorting a pixel and intersecting the ray passing through the pixel with the plane being considered. The intersection point is then projected into the reference view according to the formulation presented in section 4. The dissimilarity measure used is 1-NCC (normalized cross correlation) on a 5×5 window. However, the similarity is summed up over multiple pyramid levels [21], giving always 1/4 weight to the smaller pyramid level. For choosing the k best views we choose $k = 3$ out of $n = 7$, however for the synthetic experiments just two views have been used. Finally, for each pixel a geometric verification step is performed once the depth map for the next reference view has been computed: Each pixel of depth map one is backprojected into space, projected into the other image and compared to the depth estimated for that position. Discrepancies of more than 0.1m result in the depth value being declared invalid.

5.1. Ground Truth Evaluation

Rolling Shutter Direction: First, we qualitatively analyze the effects, when ignoring rolling shutter in stereo al-

---

gorithms for different shutter directions: In Fig. 3 we texture a square plane in 3D space and synthetically generate two rolling shutter images. Already the shape of the images looks very different (squeezed, stretched, slanted). Consequently, when ignoring this effect and performing standard stereo, the reconstructions are also biased. For this setting we chose quite strong motion to visualize the effects, but it should be clear that the same type of systematic errors will appear also at smaller speeds. Note that the obtained (biased!) depths maps for global shutter are dense; This means that obtaining visually plausible results when applying a global shutter model to rolling shutter data does not mean the data is actually correct.

Quantitative Evaluation using Ground Truth: The datasets *castle* and *old town* were originally captured using a 3D laser scanner. The resulting point clouds have been smoothed, meshed and textured with high resolution photos. We then define a plausible streetlevel camera path and render 976 images (image resolution 976×732) along the path. From each of those we pick one column and compose a novel image out of these scanlines to simulate the rolling shutter effect (afterwards, the GPU's z-buffer is handled in the same way to obtain a rolling shutter depth map). This allows for evaluation of absolute 3D errors of the 3D reconstruction algorithms in meters using extremely realistic geometry and texture. For this setup the camera is assumed to have a linear motion with a constant orientation (first motion model). We evaluated rolling shutter stereo against global shutter stereo while the camera undergoes a motion of 0m, 0.318m, 0.636m and 1.27m respectively (castle) and 0m, 0.122m, 0.243m, 0.487m (old town) during exposure time, simulating different speeds. The baseline lengths between the two images were 3.9m and 0.75m respectively. This corresponds to maximum driving speeds of 65km/h (castle) and 24km/h (old town) for exposure time of approximately 1/14s (as in our real system).

In Fig. 5 and Fig. 4 it can be seen that the global shutter algorithm performs worse with increasing rolling shutter effect while the rolling shutter algorithm is approximately constant. We visualize the 3D error, that is the distance between the estimated 3D point and the GT 3D point. Note that the GS algorithm shows errors of more than a meter which were not detected by the final 0.1m depth consistency check, confirming again that the errors when using the global shutter model are significant but hard to detect.

Approximation 1 and 2 perform in between in terms of quality, however they have a completely different error pattern. While FA1 shows a global, systematic error because of the incorrect lens undistortion, FA2 performs correct at the grid, however shows a high frequency error that increases inside the grid cells.

$\Delta \boldsymbol{x} = 127\,cm$

$\Delta \boldsymbol{x} = 49cm$ GT depth   RS = GS   GS error

0   0.5m   1.0m   1.5m
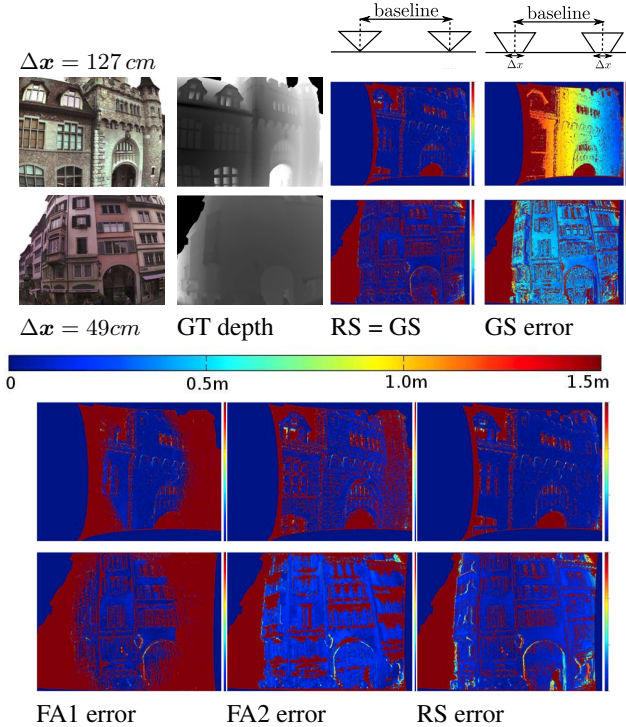
FA1 error   FA2 error   RS error

Figure 4. 3D error visualization for rolling shutter image pairs of two ground truth scenes (top and bottom): For no motion during exposure ($\Delta \boldsymbol{x} = 0$, that is global shutter) all algorithms produce the same results. For the maximum motion according to Fig. 5. For the global shutter, the error is generally higher and produces a systematic offset depending on depth and also distance to the distortion center.
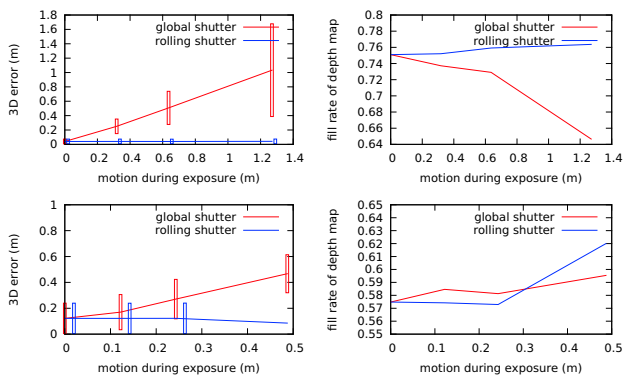

Figure 5. Top: castle sequence, bottom historic town center. Left: Median 3D error (boxes indicate median absolute deviation) when using a global or a rolling shutter algorithm. Right: corresponding fill rates of depth maps.

General motion: In another experiment (see Fig. 6) we construct the rolling shutter images in a way that the motions during exposure of image 1 is in a different local direction than the one of image 2. This happens when using different rolling shutter cameras on a car which are looking

	speed / warp [ms]	median [m]	MAD [m]	fill rate
FA 1	10.0	40.88 / 49.51	3.242 / 3.17	40.8% / 32.9%
FA 2	2.2	1.02 / 0.26	1.02 / 0.22	52.8% / 58.6%
RS	27.7	0.041 / 0.085	0.032 / 0.077	76.3% / 62%

Table 2. Evaluation of the different warps, speed vs. accuracy on the castle and old town dataset. We use a grid resolution of 1/10 of the image resolution (976×732) for FA2.

Camera 1   Camera 2   GS   RS

Figure 7. Left, sample input image of two cameras mounted on a car. We apply GS and RS stereo on both camera streams independently and fuse the resulting models into a single coordinate frame. Right, bird's eye view of GS and RS reconstruction. Note how the pole aligns in the RS reconstruction, while in the GS reconstruction it appears as two different poles.

into different directions. In this case there is no longer just a systematic bias in the data, but global shutter stereo just cannot find the correct correspondence any more.

5.2. Evaluation on Real Data

Real data has been recorded using a capture vehicle driving at different speeds. The exposure time of the camera was 72ms (approximately 1/14 s). For the *Oak street* there is approximately 0.5m displacement during exposure and a baseline of 2m between frames, driving speed was 25km/h. For the *Fillmore street* displacement was 0.74m, baseline 2.6m and driving speed 37km/h, with an image resolution of 1944×2599, results are given in Fig. 8. We compare GS and RS stereo on the Oake street sequence and observe that the facade is reconstructed further away from the camera in the GS case, compared to the RS case. Objects reconstructed from different cameras independently and fused into a single coordinate system don't overlap in the GS case while they do in the RS case, see Fig. 7.

6. Conclusion

We have analyzed the setting of camera *motion* induced rolling shutter effects and have shown that already for very moderate speeds and resolutions, effects are significant. In particular, although global shutter algorithms *seem* to work out well (resulting in a dense, smooth depth map) the results are actually not correct. We then generalized the homography transfer across a plane known for global shutter cameras to the setting of rolling shutter, considering also lens distortion that intertwines with the rolling shutter. Based on this building block, a plane sweep approach has been im-

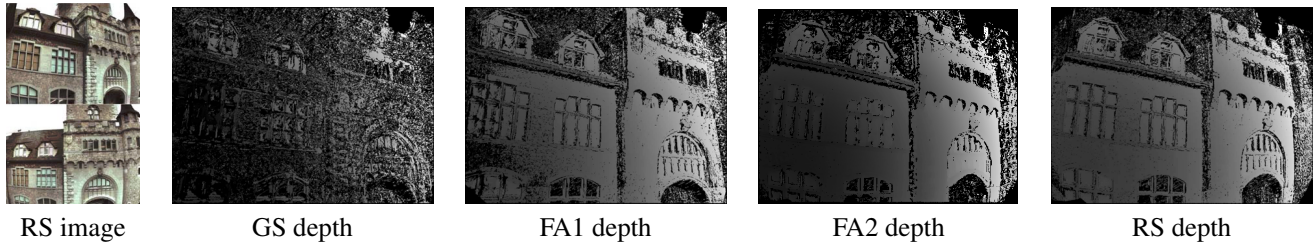| RS image | GS depth | FA1 depth | FA2 depth | RS depth |

Figure 6. Left, input images captured with independent linear motions (sideways- and forward-motion). Global shutter (GS) stereo fails as the correct correspondence is not in the search range. Fast approximation 1 (FA1) gives consistent depth which degrades towards the image boundaries. Fast approximation 2 (FA2) provides consistent depth at the grid vertices which then degrades inside the grid cell. Rolling shutter (RS) stereo provides throughout consistent depth.

Figure 8. Real world reconstruction of the Oak street (top) and Fillmore street (bottom) data sets show systematic differences in bird's eye view (right column).

plemented that was shown to produce correct results on real and synthetic data. We have furthermore analyzed two approximations that provide a significant speedup at the cost of reduced accuracy and analyzed the structure of the residual error. This allows to decide for speed or precision in case the rule of thumb presented indicates a rolling shutter model should be used for the setting at hand.

References

[1] J. B. Alexandre Karpenko, David E. Jacobs and M. Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. In *CSTR*, 2011. 1, 2

[2] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *CVPR*, 2010. 2

[3] D. Bradley, B. Atcheson, I. Ihrke, and W. Heidrich. Synchronization and rolling shutter compensation for consumer video camera arrays. In *Int. Workshop on Projector-Camera Systems*, 2009. 2

[4] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 1971. 3

[5] D. Feldman, T. Pajdla, and D. Weinshall. On the epipolar geometry of the crossed-slits projection. In *ICCV*, 2003. 2

[6] P.-E. Forssén and E. Ringaby. Rectifying rolling shutter video from hand-held devices. In *CVPR*, 2010. 1, 2

[7] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Mach. Vis. Appl.*, 12(1), 2000. 1

[8] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *CVPR*, 2007. 6

[9] C. Geyer, M. Meingast, , and S. Sastry. Geometric models of rolling-shutter cameras. In *OMNIVIS*, 2005. 1, 2, 3

[10] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Effective calibration free rolling shutter removal. *ICCP*, 2012. 2

[11] R. Gupta and R. I. Hartley. Linear pushbroom cameras. *PAMI*, 1997. 2

[12] G. Hanning, N. Forslöw, P.-E. Forssén, E. Ringaby, D. Törnqvist, and J. Callmer. Stabilizing cell phone video using inertial measurement sensors. In *Sec. Int. Workshop on mobile Vision*, 2011. 1, 2

[13] J. Hedborg, P.-E. Forssén, M. Felsberg, and E. Ringaby. Rolling shutter bundle adjustment. In *CVPR*, 2012. 2, 6

[14] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 2008. 6

[15] A. Jordt-Sedlazeck, D. Jung, and R. Koch. Refractive plane sweep for underwater images. In *GCPR 2013*, 2013. 2

[16] B. Klingner, D. Martin, and J. Roseborough. Street view structure-from-motion. In *ICCV*, 2013. 2, 6

[17] C.-K. Liang, L.-W. Chang, and H. Chen. Analysis and compensation of rolling shutter effect. *Trans. Img. Proc.*, 2008. 2

[18] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *CVPR*, 2010. 1

[19] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *IJCV*, 2008. 1

[20] E. Ringaby and P.-E. Forssén. Efficient video rectification and stabilisation for cell-phones. *IJCV*, 2012. 1, 2

[21] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *CVPR*, 2003. 2, 5, 6

[22] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *PAMI*, 2003. 2