Structure and Motion Estimation from Rolling Shutter Video

Johan Hedborg, Erik Ringaby, Per-Erik Forssén and Michael Felsberg
Computer Vision Laboratory, Department of Electrical Engineering
Linköping University
hedborg@isy.liu.se

Abstract

The majority of consumer quality cameras sold today have CMOS sensors with rolling shutters. In a rolling-shutter camera, images are read out row by row, and thus each row is exposed during a different time interval. A rolling-shutter exposure causes geometric image distortions when either the camera or the scene is moving, and this causes state-of-the-art structure and motion algorithms to fail. We demonstrate a novel method for solving the structure and motion problem for rolling-shutter video. The method relies on exploiting the continuity of the camera motion, both between frames, and across a frame. We demonstrate the effectiveness of our method by controlled experiments on real video sequences. We show, both visually and quantitatively, that our method outperforms standard structure and motion, and is more accurate and efficient than a two-step approach, doing image rectification and structure and motion.

1. Introduction

Estimation of *structure and motion* (SaM) from image sequences, is one of the key problems in computer vision where research has reached a high level of maturity [14]. Applications of SaM include ego-motion estimation [7], automatic acquisition of 3D models from images [15], and augmented reality [17].

Unfortunately, state-of-the-art algorithms for SaM tend to fail on many new cameras because they assume that each image is captured at a single time instance. This assumption holds for cameras with a global-shutter CCD sensor or a fast mechanical shutter (e.g. DSLRs). However, in recent years an increasing number of camera manufacturers have switched to CMOS sensors, and instead of capturing the image with a global-shutter trigger, most CMOS sensors scan the image row by row. This is referred to as having a *rolling shutter* (RS) [11, 12]. A rolling-shutter readout gives a better fill-factor on the chip, but leads to geometric image distortions if the camera or the scene is non-static.

Nowadays rolling-shutter sensors are found in cameras ranging from high end professional cameras (e.g. the cinematic high resolution RED cameras) to nearly all new consumer cameras, including video-capture enabled DLSR cameras. Nearly all mobile phones have cameras with rolling shutters, both for stills (due to the absence of an mechanical shutter) and for video.

In this paper we introduce a method that estimates structure and motion from rolling-shutter video with previously unseen accuracy. The method is based on an initial rectification of interest point trajectories [21], and in contrast to earlier work [1, 17], it requires no special initialisation of the 3D structure.

1.1. Related Work

Geyer et al. [12] estimate rolling-shutter SaM on synthetic data under the assumption of fronto-parallel motion, and using a linearised screw motion. Motion under rolling shutter from known structure is studied in [2]. Structure and motion on a stereo rig where one of the cameras has a rolling shutter is studied in [3].

Ait-Aider et al. [1] solved the *perspective-n-point* [9] problem for rolling-shutter cameras where the camera pose, and linear camera motion is estimated across one frame only. However, we found that without the use of markers and known geometry (as in [1]), the achieved robustness is insufficient in practice.

Klein et al. [17] have ported their augmented reality software PTAM (parallel tracking and mapping) to the CMOS camera of the iPhone 3G. They compensate for rolling-shutter artifacts by assuming a known 3D structure. Using this model, they estimate the velocity of the camera and correct the image points assuming a constant velocity across the frame.

Neither [1] nor [17] address the problem of how to estimate an initial 3D structure using an RS camera. In [17] the system is initialised by imaging a planar scene, for which a homography is iteratively refined over a number of frames. After initialisation, a 3D structure is then allowed to stabilise itself over time. The system relies on the initialisation

being sufficiently accurate, as all subsequent RS corrections assume that the 3D structure is correct. In contrast, we are able to estimate an initial 3D structure by explicitly modelling the RS geometry.

One part of our framework is similar to that of stabilising rolling-shutter video, where a successful approach has been to neglect the translation component of the camera motion, and only correct for camera rotations [10, 21].

1.2. Contributions

The contributions of this paper are:

- We introduce a method for estimating the structure and motion from rolling-shutter video without any a-priori known 3D structure. Our new method handles any type of motion and an arbitrary number of images.
- We do a thorough comparison of three algorithms on real rolling-shutter video data. The tested methods are (1) Our rolling-shutter aware SaM, (2) global-shutter SaM (i.e. not rolling-shutter aware), and (3) off-the-shelf rolling-shutter image rectification followed by global-shutter SaM.

2. Rolling-shutter image rectification

Since images and videos captured with a moving rolling-shutter camera will suffer from geometric distortions, it is often desirable to rectify and correct these. In recent years, several rolling-shutter rectification methods have been developed. One of the more successful ones is the *global affine distortion model* of Cho and Hong [8]. This has recently been extended by Baker et al. [4] to use different models across several horizontal stripes, yielding a *local affine model*. Another approach is to model the distortion using a *rotational model* [10, 21] of the camera motion.

We have chosen to use the rotational model for two reasons: (1) it models the actual camera motion and (2) it has been shown to perform better than others in a recent evaluation [21]. Even though the translational component of the camera motion is neglected, the model has been shown to work well if a short frame interval is used [21]. The method assumes the distortions arise from camera motion in a rigid scene.

2.1. Camera model

Using the pinhole camera model, the relationship between a 3D point, $\mathbf{X}$, and its projection in the image, $\mathbf{x}$, can be expressed as

$$\mathbf{x} = \mathbf{K}[\mathbf{R}|\mathbf{d}]\mathbf{X}, \tag{1}$$

if the camera has a global shutter (or is stationary). $\mathbf{x}$ is given in homogeneous coordinates, $\mathbf{K}$ is a 5DOF upper triangular 3×3 *intrinsic camera matrix*, $\mathbf{R}$ is the camera rotation and $\mathbf{d}$ is the camera translation [14].

If the moving camera instead makes use of a rolling shutter, the model becomes:

$$\mathbf{x} = \mathbf{K}[\mathbf{R}(t)|\mathbf{d}(t)]\mathbf{X}, \tag{2}$$

where t represents time and is proportional to the image row.

Since most rolling-shutter artifacts arise due to camera rotation, the model can be simplified to a pure rotation about the camera center [21]:

$$\mathbf{x} = \mathbf{K}\mathbf{R}(t)\mathbf{X}. \tag{3}$$

2.2. Camera motion estimation

By assuming that the camera has a smooth motion, the rotations can be parametrised with a linear interpolating spline where a number of knots, called *key-rotations*, are placed one for each frame. Intermediate rotations are interpolated using SLERP (Special Linear intERPolation) [23]. The frame interval used for estimating the camera motion is set short to ensure small translations.

The key-rotations are represented as a three element axis angle vector, and are estimated using iterative optimisation with a rigid scene assumption. The (symmetric) image-plane residuals of a set of corresponding image points are used for the minimisation [21].

In order to calculate the length of the spline between the last row of one frame, and the first row of the consecutive frame (the inter-frame delay), read out time and frame-rate must be known. The frame-rate is given by the video and the read out time is calibrated as in [21].

2.3. Rectification

Once the sequence of rotations has been found, we rectify the images by moving each row to a common coordinate system. If $\mathbf{x}_d$ is a distorted image point it is mapped to its rectified position $\mathbf{x}_r$ by:

$$\mathbf{x}_r = \mathbf{K}\mathbf{R}^T(t)\mathbf{K}^{-1}\mathbf{x}_d \tag{4}$$

where $\mathbf{R}$(t) is the camera rotation from the position it had when it was imaging the first row in the camera, to the position when it was imaging the row where point $\mathbf{x}_d$ is located. Since all pixels within a row were acquired at the same time, they share the same transformation. A new, rectified image is created by applying (4) to all image pixels.

3. Structure and motion

The structure and motion of a rolling-shutter camera can be estimated in several ways, where the most naive approach is to ignore the fact that the camera has a rolling shutter, *i.e.* using global-shutter SaM. A second approach is to apply global-shutter SaM on rectified images. Finally, as we suggest, rectification, structure, and motion are estimated jointly.

3.1. Point correspondences and outlier rejection

The first step in most SaM schemes is to find a set of corresponding points between the images. Usually, this is achieved by first extracting descriptors and then matching them. This method is standard if the images are captured with large baseline and under varying illumination. In this paper, the image data is captured at high frame-rate (30 fps) and has therefore small relative viewpoint and illumination changes. In this situation, a point tracking method is both more efficient and more accurate. The applied point tracker minimises the L_2-norm between square patches in the two images. The search is done with a standard gradient descent approach, which is known as the KLT-tracker [19]. Candidate points are selected using FAST [22].

Most of the methods that deal with outlier rejection in global-shutter SaM build on the assumption that the image has been taken at one time instance. When this is not the case, as in rolling-shutter images, they fail to a varying degree. We propose to use a more local approach which is achieved by re-tracking the tracked points back from the current to the previous image, and rejecting correspondences where the deviation in initial and resulting position is larger than a certain threshold [5].

3.2. Global-shutter SaM

With global-shutter SaM we refer to the process of reconstructing scene structure and camera poses from images where the whole image is exposed during the same time interval.

We assume having access to the camera or a similar camera and that the focus mechanism does not change the intrinsic parameters too much. In this case, the lens distortion and the intrinsic camera parameters can be estimated by using the method of Zhang [26].

A widely used approach to do global-shutter SaM is to first find an initial geometry for a small set of views, *e.g.* using the approach by Nistér [20], and use this as an initialisation for a bundle adjustment procedure.

In Nistér's approach, the essential matrix and thus the relative pose between two views are determined for a minimal set of five point correspondences. With the relative pose we estimate the 3D position of the 5 points, and with the *perspective-3-point algorithm* [13] the points are projected into a third camera. The closed-form three view solver is used inside a robust estimator framework, *e.g.* RANSAC [9]. This approach has been shown to be one of the best performing ones when a calibrated camera is used [6]. When the camera poses are estimated we apply an optimal triangulation scheme [16] to generate the 3D points.

The initialisation step is followed by a bundle adjustment scheme where camera views and geometry are successively added to the reconstruction until all views are incorporated and a final reconstruction is achieved. The camera views are added by using the perspective-n-points algorithm [14], where both the 3D points and the corresponding 2D image points are used to estimate the new camera pose. When adding a set of new views, new point correspondences appear and from these we can add more points to the geometry by triangulation. After each set of new camera views is added, the current camera poses and 3D points are adjusted to minimise the reprojection error using a bundle adjustment solver [25]. The number of camera views added between each bundle adjustment optimisation step is chosen so that the convergence of the optimisation is not compromised.

3.3. Rolling-shutter SaM

Rectified rolling-shutter images can be used together with a global-shutter SaM procedure to perform reconstruction. The advantage is that existing methods for reconstruction and for rectification (*e.g.* the rotation method mentioned above or the Deshaker software [24]) can be used off-the-shelf.

A drawback of combining both methods as black-boxes is that both need point correspondences by point tracking or by feature matching, and that these comparably expensive algorithms must be run twice. The rectification step also resamples the input image resulting in potential aliasing or blurring. Since the rectification step transforms the input image, the lens distortion correction in the reconstruction phase cannot be done correctly.

We therefore propose to combine the methods such that the point detection and tracking is only done once, and such that the lens distortion is compensated correctly. The procedure can be divided into a number of steps:

1. Find initial camera poses and structure by:
   (a) Tracking interest points for three or more views
   (b) Estimate camera rotations and rectify points for the views
   (c) Estimate an essential matrix and then the relative poses for 3 of the views (with RANSAC)
   (d) Triangulate points
2. Track interest points for a number of consecutive views
3. Rectify the newly tracked points
4. Find new camera poses with perspective-n-point
5. Triangulate new points
6. Perform bundle adjustment
7. If more cameras, then go to 2

The methods used in steps 1 (b)-(c) and 3-5 are the ones described in section 3.2. Steps 2-5 are performed for all new views in the sequence.

Figure 1. Tripod with pencil marker. This is the actual setup used for data collection.

Since we are correcting for lens distortions, special care has to be taken in the rectification step. In the lens correction step, points within the same row in the original image may be transformed to different rows. As the transformation depends on the unrectified image rows, we thus need to define the times t_n and t'_n to be proportional to these. The cost function for estimating the rotations in [10] is now changed so it depends on both these versions of the points:

$$J = \sum_{n=1}^{N} d(\hat{\mathbf{x}}_n, \mathbf{H}\hat{\mathbf{x}}'_n)^2 + d(\hat{\mathbf{x}}'_n, \mathbf{H}^{-1}\hat{\mathbf{x}}_n)^2 \quad (5)$$

$$\text{where} \quad \mathbf{H} = \mathbf{K}\mathbf{R}(t_n)\mathbf{R}^T(t'_n)\mathbf{K}^{-1} \quad (6)$$

$$\text{and} \quad d(\mathbf{x}, \mathbf{x}')^2 = (x_1/x_3 - x'_1/x'_3)^2 + (x_2/x_3 - x'_2/x'_3)^2 \,. \quad (7)$$

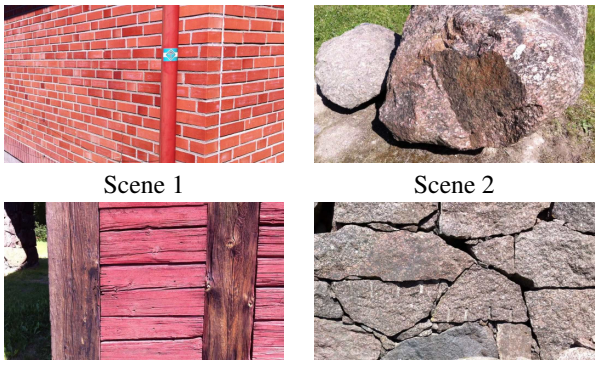
N is the number of used points in one image, and $\hat{\mathbf{x}}_n \leftrightarrow \hat{\mathbf{x}}'_n$ are corresponding points which have been lens-corrected. This cost function is used in the iterative optimisation to estimate the camera rotation. The points are rectified with these rotations, which are then used to get the global pose and structure.

4. Experimental evaluation

For evaluation, we collected data by moving the camera by hand along a near linear path and then returning it to the starting position. In order to find the starting position again, we used a tripod, equipped with a pen marker to define the common starting and stopping point. The setup is shown in figure 1 and frames from each of the 4 different scenes are shown in figure 2. These scenes are used to evaluate three different methods. The first method is global-shutter SaM (described in section 3.2) used on the original data, the second method is global-shutter SaM used on rectified images from the Deshaker software, and the last method is our own, described in section 3.3.

4.1. Experimental setup

All image data is collected using an iPhone 4, which has a camera capable of capturing HD video (1280 × 720) at 30 frames per second. The iPhone was calibrated using the



Scene 1 Scene 2 Scene 3 Scene 4

Figure 2. Sample frames from the four different scenes used in the evaluation.

OpenCV implementation of Zhang's method [26], and our implementation is configured to use 5 intrinsic parameters and 4 parameters for the lens distortion.

As mentioned in section 2.2 rolling-shutter cameras differ in readout time, and in order to get an accurate motion estimation one has to account for the readout time of the current camera. The readout time is approximated by a calibration step using a flashing LED, as described in [12, 21]. The readout time estimated for the iPhone 4 was 31.98 ms.

The KLT tracker [19] used is the implementation found in OpenCV. The track-re-track threshold is set to 0.05 pixel, in order to eliminate as many outliers as possible. For the relative pose estimation, a large enough baseline is created by picking 3 images that are 8 images apart and this is done for 500 iterations within a RANSAC framework.

The bundle adjustment solver used in this paper is the implementation by Lourakis [18]. It is semi sparse in the sense that it implements the Schur complement but not the sparse Cholesky decomposition, which is quite suitable for the amount of views and the view-overlap that is present in the video data. We run the bundle adjustment optimisation on the whole system after 10 new camera views were added. Adding fewer camera views in-between every bundle step did not improve the result for any of the evaluated methods on our data.

Deshaker was used with a rolling shutter amount of 97.11% while motion smoothness and maximal correction limits were set to their lowest values. This disables the stabilisation but still corrects for rolling shutter. Other settings were set to be most precise and most robust, together with edge filling to remove black borders.

4.2. Results

The reconstruction results for the four different scenes are presented in three different ways. In figures 3-6 we visually show the camera trajectories. The second evaluation

Scene	#1	#2	#3	#4
GS SaM	0.3636	0.0070	0.9855	0.9677
Deshaker	**0.0540**	0.0063	0.1000	0.9218
Our	0.0650	**0.0052**	**0.0583**	**0.0286**

Table 1. Distance between the first and last camera view.

Scene	#1	#2	#3	#4
GS SaM	4.33	4.02	55.64	312.68
Deshaker	**3.50**	3.28	**2.90**	3.49
Our	3.53	**3.14**	2.93	**3.11**

Table 2. Mean reprojection error in pixels.

measure is shown in table 1, where the distance between the first and the last camera view is presented. The distance is normalised with the distance to the camera view furthest away from the starting position for each scene. This camera view is determined by visually inspecting the sequences. In the ideal case start and end view should coincide, thus the best score is the lowest. The mean reprojection error after bundle adjustment (in pixels) for all views is the last evaluation measure and is shown in table 2.

The results for the global-shutter SaM on scene #1 show large drift, especially at the turning point where the rolling-shutter artifacts are larger. The camera motion in the scene is quite smooth, and the Deshaker method and our method handles the scene very well, as can be seen in table 1 and 2.

Scene #2 has good texture and a well defined 3D structure and is the most well conditioned scene of the four. Parts of the scene are tracked through the whole sequence, resulting in that even the global-shutter SaM method can recover and find the starting position. Our method is able to produce slightly better result on both the distance measurement in table 1 and the mean reprojection error in table 2.

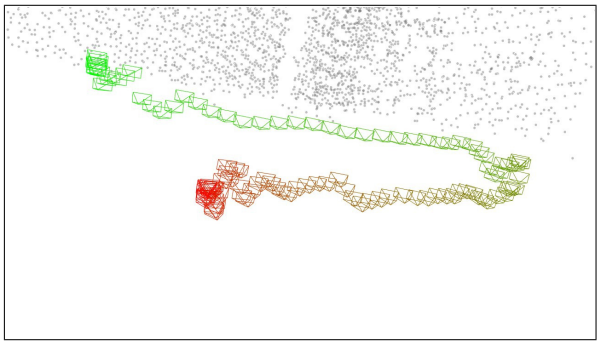
Scene #3 is more difficult, and the global-shutter SaM breaks down near the turning point. The Deshaker method handles things better, but is not able to produce as accurate results as our method, which is significantly closer to the starting point with the last camera view (see Table 1).

In Scene #4, both the global-shutter SaM and the Deshaker method break down and the only one which succeeds is our method. This is the most difficult scene of the four due to a more complex camera motion resulting in larger rolling-shutter artifacts.

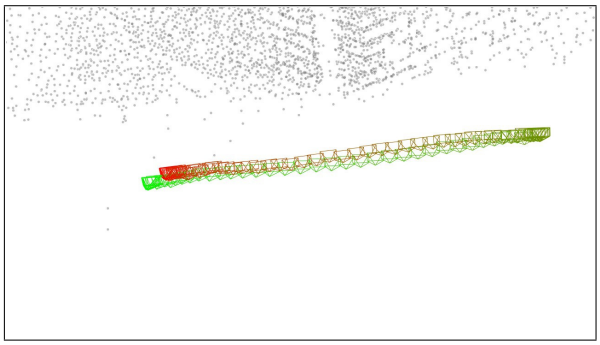
5. Concluding remarks

In this paper, we have demonstrated a structure and motion scheme for rolling-shutter video that works for general camera motions and any number of camera views. It is clearly superior to the naive approach of applying a SaM method directly on the images, which for many cases breaks down. We have further showed that our method in general gives more robust and accurate results than doing SaM on images rectified with an off-the-shelf product such as Deshaker. This is due to both a superior rectification model, and the ability to more accurately incorporate information such as lens distortion. Our method also eliminates the need for multiple point detection and tracking steps, and the need to resample the input images, making it more efficient compared to the Deshaker approach.

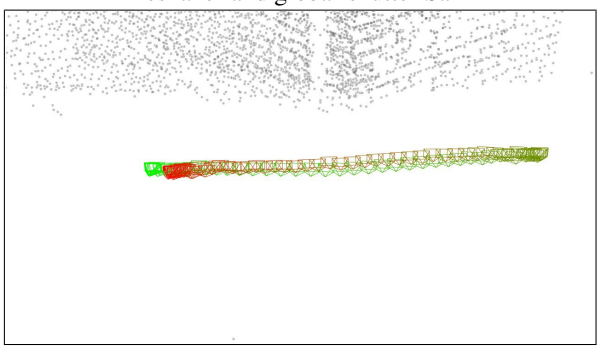
Estimating structure and motion on rolling-shutter images is often a much more ill-conditioned problem than for global-shutter images. This is clearly visible in both stability and mean reprojection error when compared to similar



Global-shutter SaM

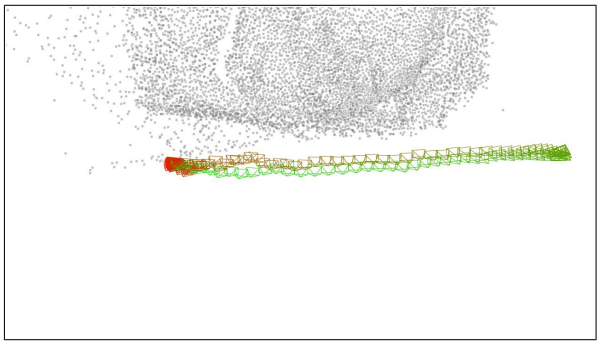


Deshaker and global-shutter SaM

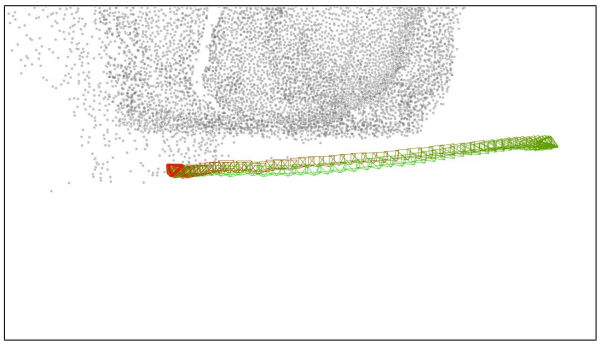


Our method

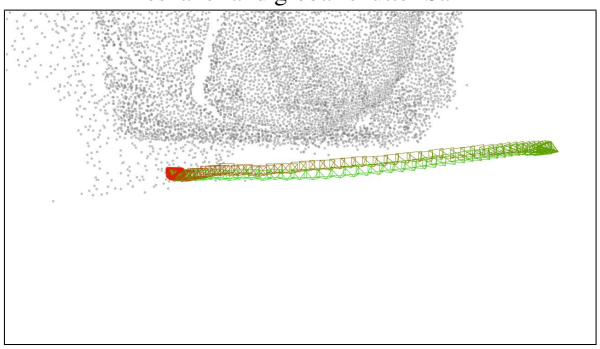
Figure 3. Results for scene #1. Images show estimated structure as grey dots, and all estimated cameras, coloured from green to red.



Global-shutter SaM

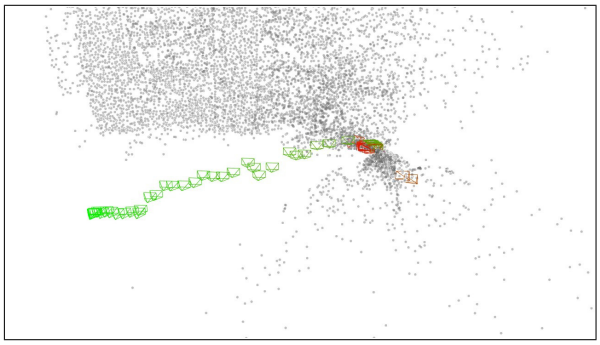


Deshaker and global-shutter SaM

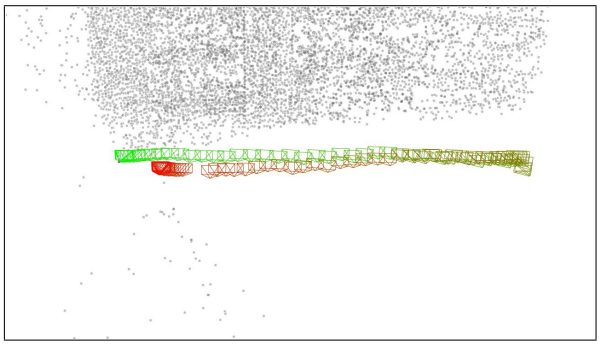


Our method

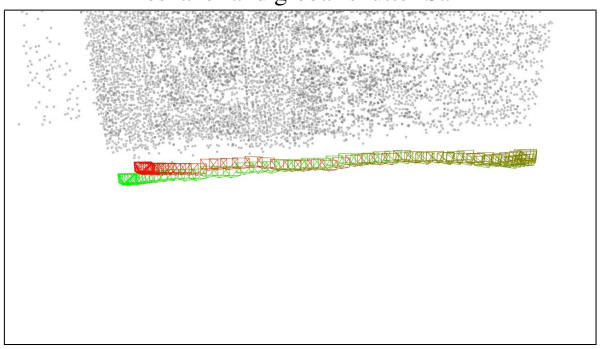
Figure 4. Results for scene #2. Images show estimated structure as grey dots, and all estimated cameras, coloured from green to red.



Global-shutter SaM



Deshaker and global-shutter SaM



Our method

Figure 5. Results for scene #3. Images show estimated structure as grey dots, and all estimated cameras, coloured from green to red.

global-shutter data.

A natural continuation of this work would be to develop a bundle adjustment solver for rolling-shutter cameras, where the presented result could serve as a good initialisation.

Acknowledgements

This work has been supported by ELLIIT, the Strategic Area for ICT research, funded by the Swedish Government, the CENIIT organisation at the Linköping Institute of technology, the Swedish Research Council through a grant for the project *Embodied Visual Object Recognition*, and by Linköping University.

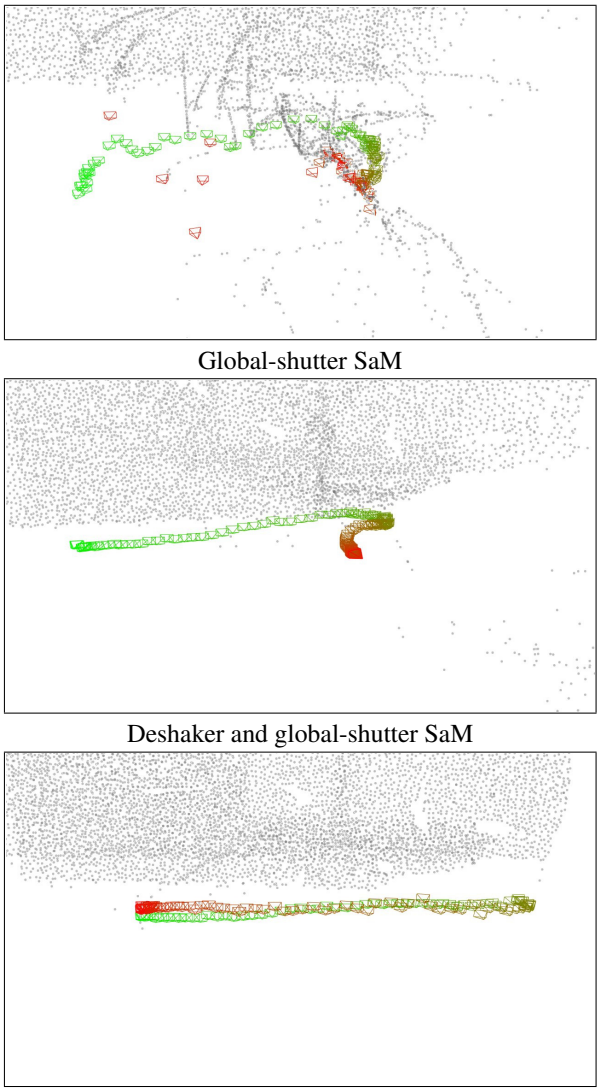
References

[1] O. Ait-Aider, N. Andreff, J. M. Lavest, and P. Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *Proceedings of ECCV'06*, pages 56–68, Graz, Austria, May 2006. 1

[2] O. Ait-Aider, A. Bartoli, and N. Andreff. Kinematics from lines in a single rolling shutter image. In *CVPR'07*, Minneapolis, USA, June 2007. 1

[3] O. Ait-Aider and F. Berry. Structure and kinematics triangulation with a rolling shutter stereo rig. In *ICCV*, 2009. 1

[4] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, June



Global-shutter SaM

Deshaker and global-shutter SaM

Our method

Figure 6. Results for scene #4. Images show estimated structure as grey dots, and all estimated cameras, coloured from green to red.

2010. IEEE Computer Society. 2

[5] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision (ICCV07)*, Rio de Janeiro, Brazil, 2007. 3

[6] M. Brückner, F. Bajramovic, and J. Denzler. Experimental evaluation of relative pose estimation algorithms. In *VISAPP (2)*, pages 431–438, 2008. 3

[7] J. Chandaria, G. Thomas, B. Bartczak, K. Koeser, R. Koch, M. Becker, G. Bleser, D. Stricker, C. Wohlleber, M. Felsberg, F. Gustafsson, J. Hol, T. B. Schön, J. Skoglund, P. J. Slycke, and S. Smeitz. Realtime camera tracking in the MATRIS project. *SMPTE Motion Imaging Journal*, 116:266–271, 2007. 1

[8] W.-H. Cho and K.-S. Kong. Affine motion based CMOS distortion analysis and CMOS digital image stabilization. *IEEE Transactions on Consumer Electronics*, 53(3):833–841, August 2007. 2

[9] 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*, 24:381–395, June 1981. 1, 3

[10] P.-E. Forssén and E. Ringaby. Rectifying rolling shutter video from hand-held devices. In *CVPR*, San Francisco, USA, June 2010. IEEE Computer Society, IEEE. 2, 4

[11] A. E. Gamal and H. Eltoukhy. CMOS image sensors. *IEEE Circuits and Devices Magazine*, May/June 2005. 1

[12] C. Geyer, M. Meingast, and S. Sastry. Geometric models of rolling-shutter cameras. In *6th OmniVis WS*, 2005. 1, 4

[13] R. Haralick, D. Lee, K. Ottenburg, and M. Nolle. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 592 –598, June 1991. 3

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004. 1, 2, 3

[15] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 2009. 1

[16] K. Kanatani, Y. Sugaya, and H. Niitsuma. Triangulation from two views revisited: Hartley-sturm vs. optimal correction. In *BMVC*, pages 173–182, 2008. 3

[17] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *(ISMAR'09)*, Orlando, October 2009. 1

[18] M. A. Lourakis and A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009. 4

[19] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981. 3, 4

[20] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE TPAMI*, 6(26):756–770, June 2004. 3

[21] E. Ringaby and P.-E. Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision*, June 2011. http://dx.doi.org/10.1007/s11263-011-0465-8. 1, 2, 4

[22] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006. 3

[23] K. Shoemake. Animating rotation with quaternion curves. In *Int. Conf. on CGIT*, pages 245–254, 1985. 2

[24] G. Thalin. Deshaker. `http://www.guthspot.se/video/deshaker.htm`. 3

[25] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000. 3

[26] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. 3, 4