A Generic Rolling Shutter Camera Model and its Application to Dynamic Pose Estimation

Ludovic Magerand
(ludovic@magerand.fr)
LASMEA, Clermont Université, France

Adrien Bartoli
(adrien.bartoli@gmail.com)
ERIM, Clermont Université, France

Abstract

Rolling shutter is an acquisition mode for CMOS cameras which sequentially exposes the scan-lines. This creates image distortions when a single object moves with respect to the camera. Previous work showed that the object pose and time-constant kinematics can be estimated from a single rolling shutter image.

We propose a generic model for rolling shutter cameras and its application to the estimation of a dynamic, time-varying, pose. The model parameters are initialised from a 'Piecewise Global Shutter' approximation and refined by a non-linear minimisation of the reprojection error under 'Derivative Based Smooth Rolling Shutter' constraints.

Contrary to previous work, our rolling shutter camera model makes it possible to estimate non-uniform dynamic pose. Experimental results on both synthetic and real data show that for a reasonable level of noise, the proposed framework outperforms previous work.

1. Introduction

CMOS sensors are generally cheaper and offer more acquisition modes than CCD sensors. One of these modes is called *rolling shutter*. It exposes the scan-lines of the sensor sequentially. We call a camera using this acquisition mode a *rolling shutter camera*. Most of the computer vision methods deal with *global shutter* cameras whereby all the sensor pixels are exposed at once. Rolling shutter cameras create distortions in the image when the single filmed object moves during image acquisition. Depending on its motion, the object may appear curved or leaned. An example of such distortions is shown in figure 1. These distortions are not modelled by the classical camera models.

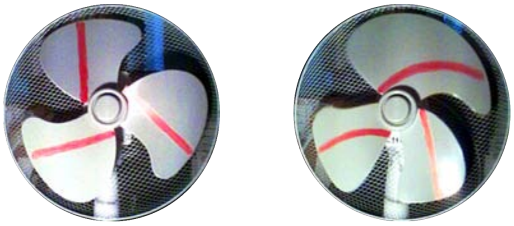

(a) Still fan. (b) Rotating fan.

Figure 1. Example of image distortions created by rolling shutter cameras.

There exist methods that compensate for the rolling shutter effect [3, 7]. Our goal is different, and draws on [1, 2], which showed that modelling the relationship between image distortions and the object motion allows one to estimate the kinematics from a single image of a calibrated rolling shutter camera. We call the camera model used in [1, 2] *uniform rolling shutter*, since it assumes the kinematics to be constant during image acquisition, which means uniform motion. Many natural motions violate this hypothesis, especially in robotics where massive accelerations may happen (tens of times earth gravity).

We propose a generic rolling shutter camera model using *dynamic pose* parametrisation, *i.e.* a pose which depends on the scan-line. Based on 2D-3D correspondences, we introduce a method to estimate a *scan-line-wise pose* implementing dynamic pose. This uses non-linear minimisation of the reprojection error subject to some constraints. We propose a set of such constraints by setting to zero the scan-line-wise pose parameters derivatives across the scan-lines at some order. We dub this method *Derivative-Based Smooth Rolling Shutter*. This optimisation is initialised using a *Piecewise Global Shutter* approximation.

Using simulated data with non-uniform motions, we analyse the behaviour of the proposed model and estimation method, and compare it to previous methods. Three parameters are considered: the density of points across the scan-lines, the number of consecutive points used in each global shutter pose estimation, and the level of noise on the image point positions. Our approach outperforms previous methods.

Paper organisation. Section 2 gives background on both the global shutter and uniform rolling shutter camera models. Section 3 introduces our generic rolling shutter camera model. The estimation method is given in section 4 and experimental results are reported in section 5.

Notation. Scalars and indexes are in italics (*e.g.* c). Vectors are column matrices and noted in bold (*e.g.* $\mathbf{v}$). Other matrices are in sans-serif (*e.g.* M). The 3D rotation matrix group is denoted $\mathcal{SO}(3)$. $\mathbb{R}$ and $\mathbb{Z}$ are the real and integer sets. All other sets are in calligraphic characters (*e.g.* $\mathcal{G}$). $\lceil r \rceil$ and $\lfloor r \rfloor$ are the upper and lower nearest integer to r. The homogeneous coordinates of a point $\mathbf{p}$ are written $\tilde{\mathbf{p}}$. The function $\boldsymbol{\varphi}(\tilde{\mathbf{v}})$ transforms homogeneous coordinates into inhomogeneous ones: $\boldsymbol{\varphi}(\tilde{\mathbf{p}}) = [p_1/p_a, \dots, p_{a-1}/p_a]^\top$ with $\tilde{\mathbf{p}} = [p_1, \dots, p_{a-1}, p_a]^\top$. The symbol $\sim$ represents the homogeneous equality. n is the number of 2D-3D correspondences, indexed by $i \in [1, n]$, and l is the number of scan-lines in an image, indexed by $j \in [1, l]$. $k_i \in [1, l]$ is the scan-line onto which point i lies.

2. Background and Previous Work

We use perspective projection modelled by the classical calibrated pin-hole projection [4, 9]. Correspondences between the 3D object points and their 2D projections in the image are supposed known. For non-linear optimisation, we use a constrained Levenberg-Marquardt method as in bundle adjustment [12].

2.1. Global Shutter

For global shutter cameras exposing all scan-lines at once, a 3D point $\mathbf{p}_i$ is projected on the 2D image point $\mathbf{m}_i^{GS}$ as

$$\tilde{\mathbf{m}}_i^{GS} \sim \mathsf{K}\,[\,\mathsf{R} \,|\, \mathbf{t}\,]\,\tilde{\mathbf{p}}_i, \tag{1}$$

where $\mathsf{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the rotational and translational parts of the object pose. K is the known internal calibration matrix. Assuming rectangular sensor elements then

$$\mathsf{K} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where α_u and α_v are horizontal and vertical scale factors and $[u_0, v_0]^\top$ is the principal point. These parameters are typically estimated using static calibration algorithms [13, 5, 14].

Given K and at least four 3D object points $\{\mathbf{p}_i\}_{i=1}^n$ with their corresponding image projection $\{\mathbf{q}_i\}_{i=1}^n$, the object pose $(\mathsf{R}, \mathbf{t})$ can generally be uniquely estimated [4, 9]. In this paper we use the EPnP method [6]. This method is based on the resolution of up to four systems of linear equations and is of complexity $O(n)$.

2.2. Uniform Rolling Shutter

It was shown in [1] that a uniform rolling shutter camera model can be obtained by extending the global shutter camera model. To do so the unknown kinematics parameters of the moving object are vectorised as $\mathbf{x}$, which is assumed constant during the image acquisition. Let $(\mathsf{R}_0, \mathbf{t}_0)$ be the initial pose of the object, to be estimated too. Equation (1)

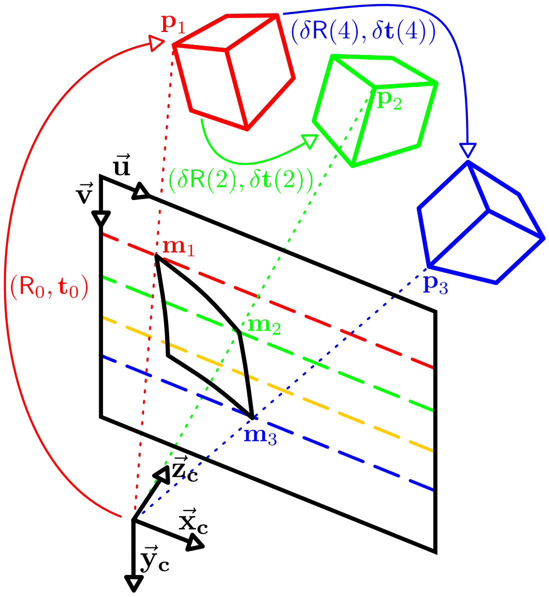

$\mathbf{p}_1$
$(\delta\mathsf{R}(4), \delta\mathbf{t}(4))$
$\mathbf{p}_2$
$\vec{\mathbf{u}}$
$(\delta\mathsf{R}(2), \delta\mathbf{t}(2))$
$\vec{\mathbf{v}}$
$\mathbf{m}_1$
$(\mathsf{R}_0, \mathbf{t}_0)$
$\mathbf{p}_3$
$\mathbf{m}_2$
$\mathbf{m}_3$
$\vec{\mathbf{z}}_\mathbf{c}$
$\vec{\mathbf{x}}_\mathbf{c}$
$\vec{\mathbf{y}}_\mathbf{c}$

Figure 2. The Uniform Rolling Shutter camera model of [1] based on Rodrigues formula. This model handles constant kinematics only.

is modified as

$$\tilde{\mathbf{m}}_i^{URS} \sim \mathsf{K}\left[\,\mathsf{R}_0\delta\mathsf{R}(k_i, \mathbf{x}) \mid \mathbf{t}_0 + \delta\mathbf{t}(k_i, \mathbf{x})\,\right]\tilde{\mathbf{p}}_i, \tag{3}$$

where $\delta\mathbf{t}(k_i, \mathbf{x})$ and $\delta\mathsf{R}(k_i, \mathbf{x})$ are given by Rodrigues formula [8]. This projection model is illustrated in figure 2.

In [1], the kinematics and the initial pose are estimated by minimising the reprojection error

$$\min_{\mathbf{x},(\mathsf{R}_0,\mathbf{t}_0)} \sum_{i=1}^{n} \|\mathbf{q}_i - \varphi\left(\tilde{\mathbf{m}}_i^{URS}\right)\|^2, \qquad s.t. \quad \mathsf{R}_0 \in \mathcal{SO}(3). \tag{4}$$

This non-linear optimisation problem is initialized by the global shutter model for $(\mathsf{R}_0, \mathbf{t}_0)$ and some constant arbitrary value for the kinematics.

3. A Unified Camera Model

3.1. Generic Rolling Shutter Camera Model

For $\mathcal{F} \subset \mathbb{Z}$, we dub *dynamic pose* a pose written as a function of the scan-line $j \in \mathcal{F}$ being considered:

$$\{(\mathsf{R}(j, \mathbf{v}), \mathbf{t}(j, \mathbf{v}))\}_{j\in\mathcal{F}}, \tag{5}$$

where $\mathbf{v}$ contains the parameters of the motion model. See section 3.2 for some practical examples.

Providing that $\mathcal{F}$ includes $\{k_i\}_{i=1}^n$, we define our generic rolling shutter camera model with dynamic pose as

$$\tilde{\mathbf{m}}_i^{GRS} \sim \mathsf{K}\left[\,\mathsf{R}(k_i, \mathbf{v}) \mid \mathbf{t}(k_i, \mathbf{v})\,\right]\tilde{\mathbf{p}}_i. \tag{6}$$

The estimation of a dynamic pose can be done accurately with this camera model using a non-linear optimisation of the reprojection error

$$\min_{\mathbf{v}} \sum_{i=1}^{n} \|\mathbf{q}_i - \varphi\left(\tilde{\mathbf{m}}_i^{GRS}\right)\|^2, \qquad s.t. \begin{cases} \mathsf{R}(j, \mathbf{v}) \in \mathcal{SO}(3), & \forall j \in \mathcal{F} \\ \mathbf{C}(\mathbf{v}) = 0, \end{cases} \tag{7}$$

where $\mathbf{C}(\mathbf{v})$ represents constraints of the motion that must be imposed to make the problem well-posed when the length of $\mathbf{v}$ is greater than $2n$, the number of constraints provided by the reprojection error. A set of such constraints is examined in section 4.2.

3.2. Instances of the Generic Rolling Shutter Camera Model

Global Shutter. Using our generic rolling shutter camera model of equation (6), global shutter can be written with

$$\begin{cases} \mathsf{R}(j, \mathbf{v}) = \mathsf{R} \\ \mathbf{t}(j, \mathbf{v}) = \mathbf{t}, \end{cases} \tag{8}$$

where $\mathbf{v}$ contains a vectorised parametrisation of the static pose $(\mathsf{R}, \mathbf{t})$. Given at least four 2D-3D correspondences, no additional constraint has to be imposed.

Uniform Rolling Shutter. The same goes for Uniform Rolling Shutter which can be written using

$$\begin{cases} \mathsf{R}(j, \mathbf{v}) = \mathsf{R}_0\delta\mathsf{R}(j, \mathbf{x}) \\ \mathbf{t}(j, \mathbf{v}) = \mathbf{t}_0 + \delta\mathbf{t}(j, \mathbf{x}), \end{cases} \tag{9}$$

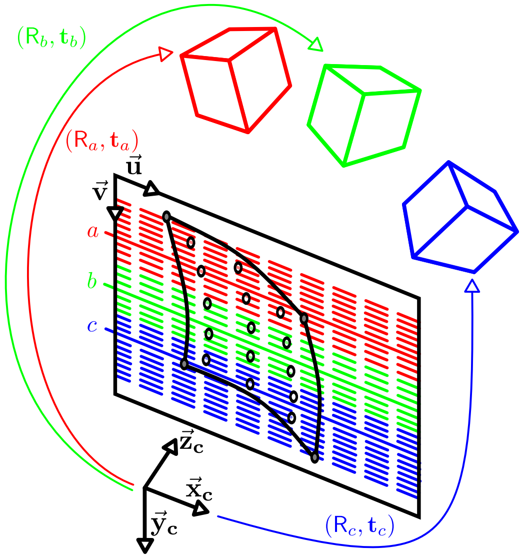

$(\mathsf{R}_b, \mathbf{t}_b)$
$(\mathsf{R}_a, \mathbf{t}_a)$
$\vec{\mathbf{u}}$
$\vec{\mathbf{v}}$
a
b
c
$\vec{\mathbf{z}}_\mathbf{c}$
$\vec{\mathbf{x}}_\mathbf{c}$
$\vec{\mathbf{y}}_\mathbf{c}$
$(\mathsf{R}_c, \mathbf{t}_c)$

Figure 3. Our piecewise global shutter initialisation. The $n = 19$ tracked points are the black ellipsoids. Here $s = 6$ and thus three sets of points appear, the first one having 7 points. The three centre scan-lines to which the estimated pose is assigned are denoted as a, b and c.

where $\mathbf{v}$ contains a vectorised parametrisation of the initial pose $(\mathsf{R}_0, \mathbf{t}_0)$ and $\mathbf{x}$, the kinematics parameter vector defined in section 2.2. Given at least seven 2D-3D correspondences, no additional constraint has to be imposed.

3.3. Scan-line-wise Pose

One of the most generic dynamic pose is obtained by parametrising the pose of each scan-line in the image: $\{(\mathsf{R}_j, \mathbf{t}_j)\}_{j=1}^{l}$. In this case, $\mathbf{v}$ contains some vectorised parametrisation of all these poses. We call such a dynamic pose *scan-line-wise pose*. It allows one to represent any motion, including non-continuous ones. We also define a sparse form of the scan-line-wise pose given by $\{(\mathsf{R}_j, \mathbf{t}_j)\}_{j \in \mathcal{S}}$ with $\mathcal{S} \subset [1, l]$. For this sparse form, $\mathbf{v}$ contains only the parameters of the pose for the scan-lines in $\mathcal{S}$.

4. Scan-line-wise Pose Estimation

In order to estimate our scan-line-wise object pose from a single rolling shutter image, we solve the non-linear optimisation problem (7) under a set of constraints based on the derivatives of the scan-line-wise pose parameters We propose an initialisation method and dub it Piecewise Global Shutter.

4.1. Initialisation: Piecewise Global Shutter

Our method to initialise the scan-line-wise pose is summarized in algorithm 1. This method is made of two main steps, explained below, but still has an overall complexity of $O(n)$.

Sparse estimation. We consider that for a limited number s of consecutive points across the scan-lines, the rolling shutter distortions can be neglected. The $\lfloor n/s \rfloor$ sets of scan-lines (line 1 *alg.* 1) does not necessarily have the same count of scan-lines. Some sets actually contain $s+1$ points as we dispatch the $n - s\lfloor n/s \rfloor$ remaining points along them. The loop (line 3 *alg.* 1) gives a sparse scan-line-wise pose $\{(\mathsf{R}_j, \mathbf{t}_j)\}_{j \in \mathcal{M}}$. This is illustrated in figure 3. Defining $\eta(j) \in \mathcal{M}$ to be the function mapping j to the centre scan-line of its set, Piecewise Global Shutter is written using (6) with

$$\begin{cases} \mathsf{R}(j, \mathbf{v}) = \mathsf{R}_{\eta(j)} \\ \mathbf{t}(j, \mathbf{v}) = \mathbf{t}_{\eta(j)}, \end{cases} \tag{10}$$

where $\mathbf{v}$ contains the parameters of the $\lfloor n/s \rfloor$ poses.

Algorithm 1: Estimation of a scan-line-wise pose using a piecewise global shutter approach.

```
1 Split the tracked points into $\lfloor n/s \rfloor$ non
  overlapping sets of consecutive points ;
2 $\mathcal{M} \leftarrow \varnothing$ ;    // List of the centre
  scan-lines
3 forall sets of consecutive points do
4     Compute $j$, the centre scan-line ;
5     $\mathcal{M} \leftarrow \mathcal{M} \cup \{j\}$ ;
6     Estimate $(\mathsf{R}_j, \mathbf{t}_j)$ using EPnP [6] ;
7 Despike using Savitzky-Golay filter [10] ;
8 Interpolate $\{(\mathsf{R}_j, \mathbf{t}_j)\}_{j \in \mathcal{M}}$ to $\{(\mathsf{R}_j, \mathbf{t}_j)\}_{j=1}^{l}$ .
```

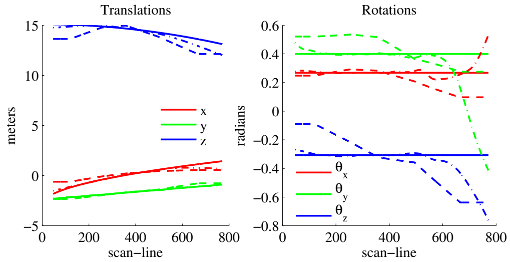


Figure 4. Example of interpolated scan-line-wise pose from simulated data with a sparse density of points across the scan-lines. The plain line is ground truth and the dashed line is the Piecewise Global Shutter initialisation. The discontinuities corresponds to the limits of the points sets. The dash dotted line is the Derivative-Based Smooth Rolling Shutter refinement.

Interpolation. The sparse scan-line-wise pose may contain some spikes due to ill-conditioned sets of points. The method we use to filter it before interpolation (line 7 *alg.* 1) performs a local polynomial regression to determine the smoothed pose parameters for each set. The interpolation (line 8 *alg.* 1) uses linear interpolation for the translational part and spherical linear interpolation [11] for the rotational part. The later interpolates rotations using unit quaternions such that the result is consistent with a rotation around a fixed axis at uniform angular velocity. Figure 4 shows an example of such an interpolation.

4.2. Refinement: Derivative-Based Smooth Rolling Shutter

Given a scan-line-wise pose $\{(\mathsf{R}_j, \mathbf{t}_j)\}_{j=1}^{l}$, it is possible to refine it. Noting $\tilde{\mathbf{m}}_i$ the 2D projection of $\tilde{\mathbf{p}}_i$ as defined in equation (6), this is done with the minimisation (7) which becomes

$$\min_{\{(\mathsf{R}_j,\mathbf{t}_j)\}_{j=1}^{l}} \sum_{i=1}^{n} \|\mathbf{q}_i - \boldsymbol{\varphi}\left(\tilde{\mathbf{m}}_i^{GRS}\right)\|^2, \qquad s.t. \begin{cases} \mathsf{R}_j \in \mathcal{SO}(3), & \forall j \in [1,l] \\ \mathbf{C}(\mathbf{v}) = 0. \end{cases} \tag{11}$$

We must define a set of constraints $\mathbf{C}(\mathbf{v})$ as there is $6l$ independent parameters in this optimisation problem and the reprojection error gives only $2n \ll 6l$ constraints.

The time between the exposure of two consecutive scan-lines is assumed constant. Therefore, setting to zero a central finite differentiation scheme of the pose parameters across the scan-lines imposes a smooth variation of these parameters over time. At a defined order d and for scan-line $j \in [1 + \lceil d/2 \rceil, l - \lfloor d/2 \rfloor]$, the derivatives of the pose parameters *w.r.t.* the scan-lines are

$$\delta^d \mathbf{v}_j = \sum_{h=0}^{d} (-1)^h \binom{d}{h} \mathbf{v}_{j+\lfloor \frac{d}{2} \rfloor - h}, \tag{12}$$

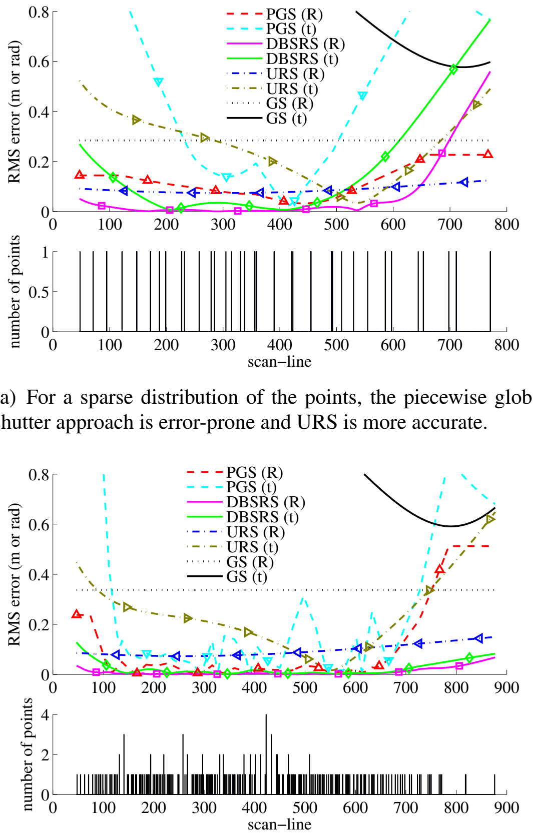
where $\mathbf{v}_j$ is the vectorised pose parameters of scan-line j. The derivative-based priors are then

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} \delta^d \mathbf{v}_{1+\lceil d/2 \rceil} \\ \vdots \\ \delta^d \mathbf{v}_{l-\lfloor d/2 \rfloor} \end{bmatrix} = \mathbf{0}. \tag{13}$$

Practical details. We parametrise the rotations using unit quaternions. The Jacobian of the reprojection residuals, the unit quaternion constraints and derivative priors with respect to the pose parameters are highly sparse. The $12n+4l+7d(l-d)$ only non-zero entries of this $2n + l + 7(l-d) \times 7l$ Jacobian are analytically computed, resulting in a fast optimization. One needs to take care of the scale of the optimized parameters. The quaternions lie in the $[0,1]^4$ range whereas the translations can take any value in $\mathbb{R}^3$. We used a rescaling based on the Jacobian in order to get rid of this issue, as in Matlab Levenberg-Marquardt implementation.

5. Results

In this section ‘PGS’ stands for the *Piecewise Global Shutter* initialisation. ‘DBSRS’ is used for *Derivative Based Smooth Rolling Shutter*. ‘URS’ means *Uniform Rolling Shutter*. In legends, these terms are followed by either ‘(R)’ or ‘(t)’, standing for rotations and translations. For DBSRS we used an arbitrary derivative order $d = 2$, as $d = 3$ tends to overfit and $d = 1$ results in discontinuities.



(a) For a sparse distribution of the points, the piecewise global shutter approach is error-prone and URS is more accurate.

(b) The same motion with a dense distribution of the points, PGS is at least as accurate as URS and DBSRS outperforms both of them.

Figure 5. The histogram shows the distribution of the tracked points across the scan-lines, and the graph shows the residual mean squared error to ground truth. PGS is sensitive to the density across the scan-lines. This sensitivity is also noticeable inside a single frame if the density is not uniform across the scan-lines With a dense distribution, DBSRS outperforms URS. In this experiment, the noise level was around half a pixel.

5.1. Simulated Data

We simulated three faces of a cube being projected with a rolling shutter camera. The cube is simulated to be either static, undergoing pure translation, pure rotation or general motion. The three motions are non-uniform as seen in figure 4.

Figure 5 shows the behaviour of all the methods with respect to the density of the points across the scan-lines. The sparser the distribution, the stranger the rolling shutter effect. A dense distribution makes the PGS hypothesis to hold better, and thus makes a better initialisation of DBSRS. From now on, every reported result is obtained using a sufficiently dense distribution, *i.e.* at least one point each 4 or 5 lines.

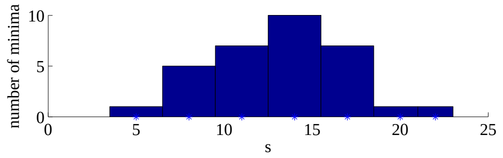


Figure 6. Distribution of the minimum errors to ground truth with respect to the value of s. The best value of s usually lies in the range $[7, 18]$.

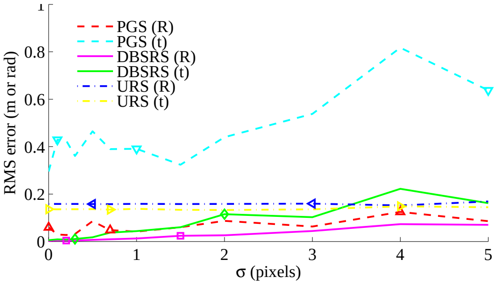


Figure 7. Behaviour of the error relatively to the noise level on tracked points coordinates. For less than a few pixels of noise, DBSRS outperforms the other approaches.

A way to limit the effect of the distribution is to carefully choose the number of consecutive points used for the global shutter pose estimation. This value acts as a trade-off between two effects: the smaller s, the better the PGS hypothesis, and the larger s, the more robust to noise the EPnP [6]. Figure 6 shows how the method reacts to variations in s. Since the PGS method is of complexity $O(n)$, we can take a greedy approach: we test all the values $s \in [7, 18]$ and keep the solution with the minimum reprojection error.

Keeping the greedy approach to automatically choose the correct s value, we can test the performance of the model when dealing with noisy data.

	GS	URS	DBSRS
Reproj. error	353px	70.24px	0.019px
Std dev. (t)	X	X	2.8mm
Std dev. (R)	X	0.0417rad	0.0493rad

Table 1. Reprojection errors, standard deviations of the translations from the mean straight line and standard deviation of the rotations parameters. X indicates entries where a value would be meaningless.

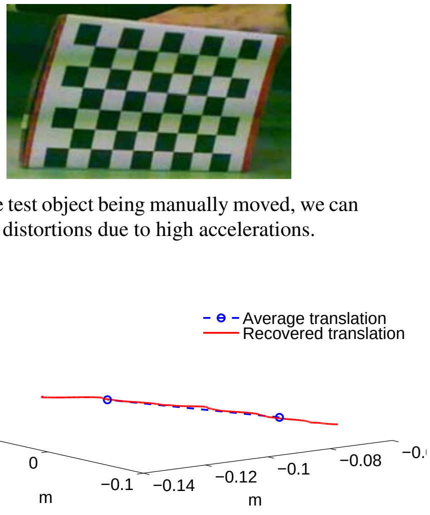
The generated noise is Gaussian, centred and of standard deviation $\sigma \in [0, 5]$ pixels. Figure 7 shows a typical response of the proposed methods to the noise, compared with URS. For high level of noise, more than a few pixels, URS may become more accurate as it is more constrained. With a reasonable level of noise, DBSRS outperforms the other approaches.

5.2. Real Data

We evaluate the method and compare it to previous ones using two sequence, one with an object undergoing a nearly pure translation and the other undergoing a nearly pure rotation. We use these two simplest motion in order to evaluate the deviation with respect to the motion without having the ground truth.

First Sequence. We use the camera of a Nokia 5800 XpressMusic phone. This sequence shows a box undergoing a translation guided by a rail. The box is manually moved quickly and randomly so that the motion is non-uniform. An example of result for this sequence is given on figure 8. Table 1 shows that the recovered dynamic pose is coherent with a nearly pure translation: quasi-static rotations and translations following a straight line. In this sequence, the noise on image points coordinates is about a few pixels, resulting from the poor quality of the camera. There is 48 correspondences.

Second Sequence. We use a PixeLINK PL-B771F filming an object undergoing a nearly pure rotation with acceleration. This sequence is made of 48 images. The point density is quite sparse as only 17 correspondences are known. One of these images and the corresponding recovered translations are shown in figure 9. We fitted a plane through the translations and a circle through the points projected on this plane. Table 2 shows that the translations are coherent with a nearly pure rotation.



(a) The test object being manually moved, we can see the distortions due to high accelerations.

(b) The recovered translations and average straight line.

Figure 8. The test object for a pure translation motion obtained by moving with high acceleration a box along a rail.

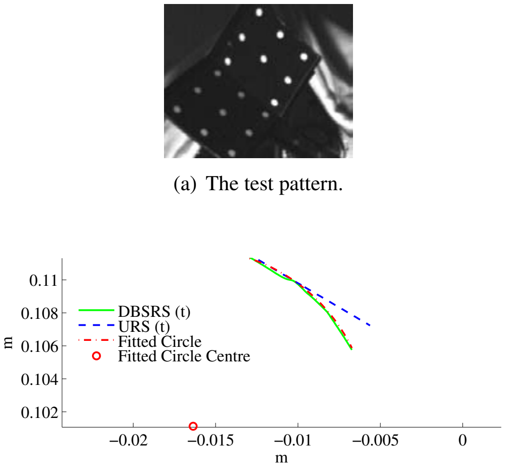
	GS	URS	DBSRS
Reproj. error	26.3px	0.781px	7e-4px
Std dev. (plane)	X	X	5.6e-5mm
Std dev. (circle)	X	X	0.1468mm

Table 2. Reprojection errors, standard deviations of the translations from the mean plane and fitted circle.

6. Conclusion

In spite of the low cost of rolling shutter cameras, they can be turned into dynamic pose sensors as shown by previous work [1], assuming the object motion to be uniform during the image exposure.

We proposed a generic rolling shutter camera model capable of handling both the global and uniform rolling shutter, and also any other type of rolling shutter image. For this, we used the con-



(a) The test pattern.

(b) The recovered translations, and fitted circle.

Figure 9. The test object for a pure rotation motion obtained by mounting a test pattern on a drill. We fitted a plane through the translations and a circle through the points projected on this plane.

cept of dynamic pose, a pose which is written as a function of the scan-line being considered. Using this camera model we showed how to estimate a scan-line-wise pose, one of the most trivial dynamic pose. We use a constrained non-linear optimisation of the reprojection error, initialised by a piecewise global shutter approach. The constraints we use are built upon the derivative of the pose parameters.

Using simulated data, we showed that this generic rolling shutter camera model is capable of handling various non-uniform motions with a better accuracy than previous methods. We also showed results on real data that exhibits estimated dynamic poses coherent with the motion given to the object: for a pure translation motion, the estimated rotations are nearly constant and the translations are nearly a straight line.

There is still some work to do to determine the limits of the optimisation used, and to expand this method to use lines correspondences as in [2].

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. *ECCV*, 2006. 1, 2, 3, 7

[2] O. Ait-Aider, A. Bartoli, and N. Andreff. Kinematics from lines in a single rolling shutter image. *CVPR*, 2007. 1, 8

[3] D. Bradley, B. Atcheson, I. Ihrke, and W. Heidrich. Synchronization and rolling shutter compensation for consumer video camera arrays. *PROCAMS*, 2009. 1

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2004. 2

[5] J. Heikkila and O. Silvén. A four-step camera calibration procedure with implicit image correction. *CVPR*, 1997. 2

[6] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate O(n) solution to the PnP problem. *IJCV*, 2009. 2, 4, 6

[7] C. Liang, L. Chang, and H. Chen. Analysis and compensation of rolling shutter effect. *IEEE Trans. on Image Processing*, 2008. 1

[8] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press Inc., 1994. 3

[9] N. Paragios, Y. Chen, and O. Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer, 2006. 2

[10] A. Savitzky and M. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 1964. 4

[11] K. Shoemake. Animating rotation with quaternion curves. *SIGGRAPH*, 1985. 5

[12] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. *Vision Algorithms: Theory and Practice*, 2000. 2

[13] R. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *CVPR*, 1986. 2

[14] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV*, 1999. 2