Rolling Shutter Camera Calibration

Luc Oth, Paul Furgale, Laurent Kneip, Roland Siegwart
othlu@student.ethz.ch, {paul.furgale, laurent.kneip}@mavt.ethz.ch, rsiegwart@ethz.ch
Autonomous Systems Lab, ETH Zurich

Abstract

Rolling Shutter (RS) cameras are used across a wide range of consumer electronic devices—from smart-phones to high-end cameras. It is well known, that if a RS camera is used with a moving camera or scene, significant image distortions are introduced. The quality or even success of structure from motion on rolling shutter images requires the usual intrinsic parameters such as focal length and distortion coefficients as well as accurate modelling of the shutter timing.

The current state-of-the-art technique for calibrating the shutter timings requires specialised hardware. We present a new method that only requires video of a known calibration pattern. Experimental results on over 60 real datasets show that our method is more accurate than the current state of the art.

1. Introduction

Most consumer electronic devices—such as smartphones—leave no room for a mechanical global shutter (GS). Those cameras make use of what is called an electronic shutter. The electronic shutter of CCD sensors exposes every row of the image during the same timespan, similar to a mechanical shutter. CMOS sensors often only have a rolling shutter (RS). The image rows are exposed and read sequentially, which introduces significant image distortions if either the camera or the scene are in motion. To account for those RS effects, the frame time and the time of integration of every line need to be known. On most RS cameras, each image is captured with a fixed *line delay*—the time between the start of integration of two consecutive lines. Using this assumption, the timestamp of each line is uniquely defined by the line delay and the frame time. A GS image only encodes the pose of an object relative to the camera frame. The RS camera may be interpreted as a high frequency sensor returning sparse spatial information with a dense temporal coverage encoded by distortions [2].

Conventional structure from motion (SfM) methods estimate a single camera pose per frame. While a single pose is sufficient in the GS case, RS models require one camera pose estimate per sensor line. This results in a large number of parameters to estimate. A common solution is to only estimate one pose per frame and perform a linear interpolation between consecutive camera poses to approximate the intermediate states. An alternative solution that allows dense temporal modelling without a growing number of parameters is given by Furgale *et al*. [7], who propose a continuous-time batch optimisation approach. In contrast to all prior work, which uses linear interpolation between poses, we propose the use of a continuous-time batch optimisation technique for higher order modelling of the rolling-shutter effect.

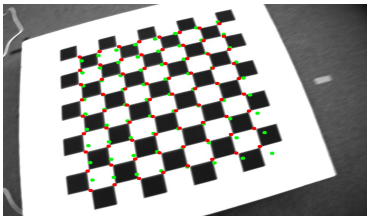

Figure 1. The reprojected chessboard corners after a continuous-time camera pose estimation. The red dots show the corner positions for our continuous-time rolling shutter model. The green dots show the results with a discrete-time global shutter model.

However, to perform SfM using a rolling shutter camera, the line delay must be known accurately. For many consumer devices, one may not have access to the data sheet or specific configuration information. Hence, the line delay must be discovered through a calibration process.

This paper presents an offline calibration procedure to determine the rolling shutter line delay for use in high accuracy SfM, such as the work of [9]. Existing autocalibrating or calibration-free algorithms (as presented by Grundman *et al*. [8] and Baker *et al*. [3]) aim at removing the rolling shutter wobble. [9] shows that SfM on wobble-corrected images performs worse (in terms of pose estimation accuracy) when compared to a continuous motion parametrisation operating on raw images with a calibrated RS model. In light of this, high-accuracy rolling shutter calibration that does not need specialised hardware (as in [14, 15]) is highly desirable.

The current state of the art for line delay calibration

was proposed by Geyer *et al*. [14]. The sensor is exposed with an LED flashing at high frequency. The resulting images include light and dark lines whose spatial frequency is linked to the line delay and the known LED frequency. The method, however, requires expensive hardware and is prone to imprecision, especially if the camera has a fixed lens. We propose to use standard GS calibration means to also calibrate the RS line delay. First, the intrinsics and distortion coefficients of a RS camera are determined with GS methods using still images and the help of a known pattern. Next, RS distortion generating camera motion is produced in front of the known pattern. The resulting video sequence is used to simultaneously estimate the camera pose in continuous-time and the line delay. This method removes the dependency on expensive hardware and we are able to show that the resulting line delay estimates are more accurate than those produced by the method of Geyer *et al*. [14].

The contributions of this paper are as follows:

1. we propose to use a continuous-time trajectory model combined with a rolling shutter camera model;
2. we propose a new method for RS calibration that requires no additional hardware except for a known pattern e.g. chessboard or circle grid also required for GS camera calibration;
3. we solve a batch estimation problem to estimate the camera pose and the line delay; in batch optimisation, the error terms are commonly weighted by their inverse covariance; we derive the covariance matrix of the reprojection error terms under a RS camera model and discuss the motion dependence of the RS covariance terms;
4. we parametrise the pose of the camera as a fourth-order B-spline and propose the first scheme for adaptively choosing the number of B-spline basis functions needed to represent different parts of the trajectory.

2. Related Work

Along with the proposition of the general projection equations for RS cameras, Geyer *et al*. [14] focus on the analysis of the projection subject to constant velocity motion. Additionally, they propose the current state-of-the-art method for line delay calibration. The image sensor is exposed with an LED flashing at high frequency. The precise knowledge of the LED frequency is essential for a successful calibration and the authors suggest to remove the lens for best sensor illumination. Later, Ringaby and Forssén [15] refine the method to cope with partly illuminated sensors occurring if the lens is not removed. Besides the line delay calibration proposed by Geyer *et al*. [14], Karpenko *et al*. [1] propose a method to estimate the full calibration of a rigidly attached camera and gyroscope, including the camera's line delay.

Ait-Aider *et al*. [2] stop considering the RS artefacts as drawback and exploit them to simultaneously extract the pose and velocity of an object relative to the camera frame from a single view. With the pose extraction, they propose the first perspective-n-point solution for RS cameras under a linear motion. Their work shows that the higher temporal density of RS cameras can effectively be used in real applications and stands as an advantage over GS cameras.

Some authors [14, 15] have already noted that GS approaches for SfM may fail on RS imagery due to the unmodelled feature distortions. The first specialised propositions for RS motion estimation do not directly handle the RS but circumvent its effects by first undistorting the frames with methods as proposed in [15] and subsequently applying a state-of-the-art GS SfM method. Klein and Murray [12], for example, port their parallel tracking and mapping (PTAM) framework [11] to an Apple iPhone 3GS. They first estimate a constant camera velocity between two consecutive frames which is then used to undistort the RS keypoints. The new keypoints are afterwards used in their GS framework.

A more advanced approach is proposed by Hedborg *et al*. [10], the first RS-specific structure from motion algorithm. The camera trajectory is described by linearly interpolating between the camera poses at the beginning of each frame. The rotational interpolation is performed using SLERP (Spherical Linear intERPolation [16]). They also propose to first undistort the keypoints and then apply a GS SfM algorithm. Considering the observation, that the most significant RS distortions occur for rotational motion, they suggest to invert the projection equations with a rotation-only assumption to undistort the keypoints [15].

The preliminary undistortion of the keypoints significantly improves the results of SfM algorithms. However, the additional step is not necessary and error prone. Furthermore, it ignores the important velocity information contained in the RS data. Hedborg *et al*. make the same observation and generalise their approach. They remove the unnecessary undistortion step to propose the first RS bundle adjustment [9]. The pose between two consecutive frames is approximated with a linear interpolation for the position and SLERP for the rotation parameters. They adapt the bundle adjustment equations and propose the triangulation and PnP steps for RS cameras. For solving the perspective pose problem, they suggest to use a multi-frame PnP solver and simultaneously estimate the pose of the camera in multiple frames. The direct use of the RS images improves the performance and stability of the pose estimation.

The pose parametrisations proposed in earlier publications are however not optimal as they assume a constant velocity between two consecutive discrete camera poses. The camera motion is thus assumed to be linear, which stands in conflict with a general rapid motion. Recent publications in the domain of continuous-time estimation and their application to SLAM, as proposed in [7], allow a more general tra-

jectory description. This new approach does not imply any assumptions about the motion. The calibration approach proposed by Geyer *et al*. [14] and refined by Ringaby and Forssén [15] requires a complex hardware setup. The approach of Karpenko *et al*. [1] relies on a gyroscope. We propose a new calibration approach involving only a known pattern as also used for the intrinsics and distortion coefficient calibration.

3. Theory

We first introduce the concepts of continuous-time trajectory modelling in its most general form. Next, we propose a continuous-time perspective projection model for RS cameras derived from the general projection equations, and give the related perspective localisation theory. We derive the equations for line delay estimation and combine them with the perspective pose equations to simultaneously estimate the camera pose and the line delay based on a set of known landmarks. Finally, we derive the covariance matrix of the RS reprojection errors to build a standard maximum likelihood estimator.

3.1. Continuous-Time Pose Parametrisation

The following derivations are based on the concepts introduced by Furgale *et al*. [7]. Let the pose of the camera, $\mathbf{x}(t)$, be represented as the weighted sum of a set of known analytical temporal basis functions[1],

$$\mathbf{x}(t) := \mathbf{\Phi}(t)\mathbf{c}, \tag{1}$$

$$\mathbf{\Phi}(t) := \begin{bmatrix}\boldsymbol{\phi}_1(t) & \dots & \boldsymbol{\phi}_B(t)\end{bmatrix}, \tag{2}$$

where $\boldsymbol{\phi}_b(t)$ are the known analytical $D \times 1$ functions of time stacked in the $D \times B$ basis matrix $\mathbf{\Phi}(t)$. The $B \times 1$ coefficient matrix, $\mathbf{c}$, is the generating quantity that describes the shape of the D trajectories.

The continuous-time representation of a transformation matrix is obtained by first expressing the pose parameters as a vector-valued continuous-time function

$$\mathbf{x}(t) := \begin{bmatrix}\mathbf{t}(t)\\ \boldsymbol{\varphi}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_{\mathbf{t}}(t)\mathbf{c}_{\mathbf{t}}\\ \mathbf{\Phi}_{\boldsymbol{\varphi}}(t)\mathbf{c}_{\boldsymbol{\varphi}}\end{bmatrix}, \tag{3}$$

with $\mathbf{t}(t)$ the translational and $\boldsymbol{\varphi}(t)$ the rotational parametrisation. If we let $\mathbf{C}(\cdot)$ be a function converting a set of rotation parameters into the corresponding rotation matrix, we obtain the continuous-time transformation matrix:

$$\mathbf{T}(t) = \begin{bmatrix}\mathbf{C}(\boldsymbol{\varphi}(t)) & \mathbf{t}(t)\\ \mathbf{0}^T & 1\end{bmatrix}. \tag{4}$$

3.2. Rolling Shutter Camera Model

The general perspective projection model for cameras is given by

$$\mathbf{u}(t) := \boldsymbol{\pi}(\mathbf{T}(t)\mathbf{p}), \tag{5}$$

[1]Please note, that this is the most general form of a basis function parametrisation. The derivations in the following section remain independent of the basis function choice and the choice is up to the user.

where $\mathbf{p}$ is a homogeneous point expressed in world coordinates, $\boldsymbol{\pi}(\cdot)$ is the camera's projection function, $\mathbf{u}(\cdot)$ is the projection in the image plane and t the time of projection.

In a general RS setup, the timestamp of a feature is unknown, only its position in the image plane is observable. We assume that the start time of integration of the first line of the image, $\bar{t}$, is known. Beginning the integration at the first line, and assuming that the line delay, d, remains constant, the time of exposure of the vth line becomes: $t = \bar{t} + vd$. The projection of a point, $\mathbf{p}$, onto the image plane, $\mathbf{u}(t) = (u, v)$, expressed relative to the frame time, $\bar{t}$, is then written as:

$$\mathbf{u}(t) = \boldsymbol{\pi}(\mathbf{T}(\bar{t} + vd)\mathbf{p}). \tag{6}$$

Equation (6), with the proposed continuous-time parametrisation (4) is a fully analytic expression. This model is more general than the linear interpolation proposed in previous publications [1, 9, 10, 15]. we encode a more general continuous motion instead of linear interpolations between consecutive frames.

3.3. Reprojection Error Modelling

In the following, we derive the reprojection error terms for the RS case and build the perspective localisation problem. Perspective localisation is widely known and aims at minimising the squared reprojection error of a set of known landmarks (k) in every frame (i), weighted by the inverse covariance matrix of the error, $\bar{\mathbf{R}}_{k,i}$. Following the approach described in the previous section, we estimate the column of basis function coefficients, $\mathbf{c}$, and a set of time-invariant parameters[2], $\boldsymbol{\theta}$. The perspective localisation problem becomes

$$\mathbf{c}^*, \boldsymbol{\theta}^* = \underset{\mathbf{c},\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i,k} \mathbf{e}_{k,i}^T \bar{\mathbf{R}}_{k,i}^{-1} \mathbf{e}_{k,i}, \tag{7}$$

where the reprojection errors are defined as the difference between an observed feature, $\mathbf{y}_k^i = (u_k^i, v_k^i)$, generated by a landmark, $\mathbf{p}_k$, in the ith frame and the reprojected landmark given the camera position $\mathbf{T}(\cdot)$:

$$\mathbf{y}_k^i = \boldsymbol{\pi}(\mathbf{T}(\bar{t}_i + v_k^i d)\mathbf{p}_k) + \mathbf{n}_{k,i}, \tag{8}$$

$$\mathbf{e}_{k,i} := \mathbf{y}_k^i - \boldsymbol{\pi}(\mathbf{T}(\bar{t}_i + v_k^i d)\mathbf{p}_k). \tag{9}$$

Every $\mathbf{n}_{k,i}$ is statistically independent Gaussian noise: $\mathbf{n}_{k,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,i})$.

The minimum of (7) is found using a Gauss-Newton based optimisation which requires the linearisation of the error terms with respect to the estimated quantity, $\mathbf{c}$:

$$\mathbf{e}_{k,i} = \mathbf{y}_k^i - \boldsymbol{\pi}\left(\mathbf{T}(\mathbf{\Phi}(\bar{t}_i + v_k^i d)(\bar{\mathbf{c}} + \delta\mathbf{c}))\mathbf{p}_k\right) \tag{10}$$

$$\approx \bar{\mathbf{e}}_{k,i} - \mathbf{J}_{\boldsymbol{\pi}}\mathbf{\Phi}(\bar{t}_i + v_k^i d)\delta\mathbf{c}. \tag{11}$$

We define the nominal error $\bar{\mathbf{e}}_{k,i} = \mathbf{y}_k^i - \boldsymbol{\pi}\left(\mathbf{T}(\mathbf{\Phi}(t_k^i)\bar{\mathbf{c}})\mathbf{p}_k\right)$ and use a simplified notation for the time of exposure of the kth landmark in the ith frame $t_k^i := \bar{t}_i + v_k^i d$, which is a function of the line delay. $\mathbf{J}_{\boldsymbol{\pi}} = [\mathbf{j}_{\boldsymbol{\pi},u}\ \mathbf{j}_{\boldsymbol{\pi},v}]^T$ is the Jacobian of the projection equation (6) with respect to small

[2]The parameter vector, $\boldsymbol{\theta}$, may e.g. include landmarks or the line delay.

changes in the pose parameters, $\mathbf{x}(t)$, with $\mathbf{j}_{\boldsymbol{\pi},u}^T$ and $\mathbf{j}_{\boldsymbol{\pi},v}^T$ the components in the u and v direction, respectively.

3.4. Shutter Calibration

To estimate the line delay, we linearise the error term (9) with respect to small changes in the line delay. With the nominal error $\bar{\mathbf{e}}_{k,i} = \mathbf{y}_k^i - \boldsymbol{\pi}\left(\mathbf{T}(\boldsymbol{\Phi}(\bar{t}_i + v_k^i\bar{d})\mathbf{c})\mathbf{p}_k\right)$ we obtain:

$$\begin{aligned}\mathbf{e}_{k,i} &= \mathbf{y}_k^i - \boldsymbol{\pi}\left(\mathbf{T}(\boldsymbol{\Phi}(\bar{t}_i + v_k^i(\bar{d} + \delta d))\mathbf{c})\mathbf{p}_k\right) && (12)\\ &\approx \bar{\mathbf{e}}_{k,i} - \mathbf{J}_\pi\dot{\boldsymbol{\Phi}}(\bar{t}_i + v_k^i\bar{d})\mathbf{c}v_k^i\delta d. && (13)\end{aligned}$$

To simplify the expressions equation (13) is merged with (11) to obtain a single equation for simultaneous line delay and pose estimation:

$$\mathbf{e}_{k,i} \approx \bar{\mathbf{e}}_{k,i} - \mathbf{J}_{\boldsymbol{\pi}}\begin{bmatrix}\boldsymbol{\Phi}(\bar{t}_i + v_k^i\bar{d}) & \dot{\boldsymbol{\Phi}}(\bar{t}_i + v_k^i\bar{d})v_k^i\end{bmatrix}\begin{bmatrix}\delta\mathbf{c}\\ \delta d\end{bmatrix} \quad (14)$$

As the trajectories are expressed with analytic basis functions, the evaluation of $\dot{\boldsymbol{\Phi}}(\cdot)$ is analytical.

3.5. Error Term Standardisation

The standardisation of the error terms aims at scaling every term by its inverse covariance matrix such that all terms end up with unit variance [17]. Features with high variance—and thus a high probability of a large reprojection error—are offered less confidence by down-weighting their influence on the total error. A low-variance feature is offered more confidence by amplifying the error terms as it is precisely localised in the image. In the previous sections, we assumed that the measurement noise is Gaussian. The fundamental assumption for the following paragraphs is, that the effect of the measurement noise on the error terms is also approximately a Gaussian distribution.

As opposed to the GS case, the measurement noise $\mathbf{n}_{k,i}$ introduced in equation (8) is not directly fed through to the error term (9) such that $\mathbf{e}_{k,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k,i})$. The time at which a landmark is captured is a function of the frame time $\bar{t}$ and the measurement v, and v is affected by the second component of the measurement noise $\mathbf{n}_{k,i} = [n_{k,i}^u, n_{k,i}^v]^T$. In essence, our uncertainty in the image plane is translated into an uncertainty in the time the feature was observed and both effects must be considered.

We approximate the covariance of the error terms by linearisation of the measurement equations (8). This gives the approximate mapping between a small perturbation, $n_{k,i}^v$, of the vertical feature position, v, and the feature's position:

$$\boldsymbol{\pi}(\mathbf{T}(\bar{t}_i + (\bar{v}_k^i + n_{k,i}^v)d)\mathbf{p}_k) \approx \quad (15)$$

$$\boldsymbol{\pi}(\mathbf{T}(\bar{t}_i + \bar{v}_k^i d)\mathbf{p}_k) + \mathbf{j}_{\boldsymbol{\pi},v}^T(t_k^i)\dot{\boldsymbol{\Phi}}(t_k^i)\mathbf{c}dn_{k,i}^v. \quad (16)$$

With the measurement equation (8) and the error term (9) we approximate the effect of the noise, $\mathbf{n}_{k,i}$, on the error terms:

$$\mathbf{e}_{k,i} = \bar{\mathbf{e}}_{k,i} + \mathbf{j}_{\boldsymbol{\pi},v}^T(t_k^i)\dot{\boldsymbol{\Phi}}(t_k^i)\mathbf{c}dn_v^k + \begin{bmatrix}n_{k,i}^u\\ n_{k,i}^v\end{bmatrix} \quad (17)$$

$$= \bar{\mathbf{e}}_{k,i} + \underbrace{\left(\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + \begin{bmatrix}0 & 1\\ 0 & 1\end{bmatrix}\mathbf{j}_{\boldsymbol{\pi},v}^T d\dot{\boldsymbol{\Phi}}(t_k^i)\mathbf{c}\right)}_{\mathbf{A}_k(t_k^i)}\begin{bmatrix}n_{k,i}^u\\ n_{k,i}^v\end{bmatrix}. \quad (18)$$

The nominal error $\bar{\mathbf{e}}_{k,i} = \mathbf{y}_k^i - \boldsymbol{\pi}(\mathbf{T}(\bar{t}_i + \bar{v}_k^i d)\mathbf{p}_k)$ and $\mathbf{A}_k^i := \mathbf{A}_k(t_k^i)$ maps the feature variance onto the error term. It is important to note that in the linearised form, (18), the reprojection error terms fulfil the conditions under which the least squares solution is equivalent to the maximum likelihood solution. The expected value of the error terms is zero, $\mathbb{E}[\mathbf{e}_{k,i}] = 0$, and the second condition—Gaussian error term variance—is given, as the Gaussian noise, $\mathbf{n}_{k,i}$, is linearly mapped onto the error terms.

The GS relation of (18) is obtained if $d = 0$. As expected only a diagonal mapping, $\mathbf{A}_k^i = \mathbf{1}_2$, remains, with $\mathbf{1}_2$ the 2×2 identity matrix. For $d \neq 0$, the variance of the feature detector is mapped onto both error term components. In general the measurement noise—and thus the error covariance—is assumed to be constant over time. This assumption is no longer valid as the feature noise is mapped onto the error terms with the time-dependent matrix $\mathbf{A}_k^i$. The scalar $\mathbf{j}_{\boldsymbol{\pi},v}(t_k^i)\dot{\boldsymbol{\Phi}}(t_k^i)\mathbf{c}$ defines the scale factor of $\mathbf{n}_{k,i}$. The influence of the off-diagonal terms becomes significant for most fast motion patterns: small changes in the feature row may, under rapid motion, induce a large change in the camera position and propagate back to a large change in the feature position in the image plane. For a slow motion the matrix $\mathbf{A}_k^i$ tends to the GS case: identity. The covariance matrix of the error terms is finally obtained via:

$$\begin{aligned}\bar{\mathbf{R}}_{k,i} &= \mathbb{E}\left[(\bar{\mathbf{e}}_{k,i} - \mathbf{e}_{k,i})(\bar{\mathbf{e}}_{k,i} - \mathbf{e}_{k,i})^T\right] && (19)\\ &= \mathbb{E}\left[\mathbf{A}_k^i\mathbf{n}_{k,i}\mathbf{n}_{k,i}^T{\mathbf{A}_k^i}^T\right] && (20)\\ &= \mathbf{A}_k^i\mathbf{R}_{k,i}{\mathbf{A}_k^i}^T. && (21)\end{aligned}$$

Please note that the covariance is not constant in time and needs to be re-evaluated each time the line delay changes during estimation.

To visualise the effect of motion on the error covariance ellipse we plot several examples in Figure 2. In this example, the camera x-axis is aligned with the image u coordinates, parallel to the sensor rows, the camera y-axis is aligned with the v coordinates, aligned with the line scanning direction, and the camera z-axis points towards the landmarks in the scene. In Figure 2 (a-c) the landmark is placed directly along the optical axis. A perturbation in the vertical feature position changes the time of exposure of the feature. If the camera moves along the x-axis, the position of the camera also changes. The changed position produces a shift in the u coordinate of the feature, Figure 2 (a).

Figure 2 (b) shows the covariance ellipses if the cam-

era motion is parallel to the line scanning direction. If the camera motion has the same direction as the line scanning, the timespan during which a landmark is projected onto a specific line is enlarged. The variance in the v coordinate is thus reduced. The covariance matrix may, however, become degenerate for fast motion along the line scanning direction. This does not occur in practice but has to be taken into account during implementation. Moving the camera in the opposite direction reduces the timespan during which a feature is projected onto a same line. Consequently, the variance along the v-axis increases.

Placing a feature at an arbitrary position, as for example the lower right of the image plane, produces similar covariance plots. The projection equations with the nonlinear distortion model slightly rotate the covariance around the z-axis. A significant change is observed for motion along the z-axis (Figure 2 (d)) as the feature position is now influenced by the distance between the landmark and the image plane.

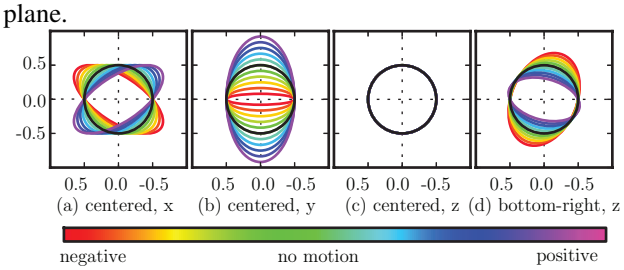

Figure 2. Covariance ellipse plots assuming a unit variance (black) of the feature detector, (21). Figures (a,b,c) are for a feature centred in the image plane. Figure (d) assumes a feature at the bottom right of the sensor. The motion is decomposed for all three translational axes and starts with a velocity in the negative direction (red), goes to zero-motion (green) and to a velocity in the positive direction (blue). The line scanning direction is the positive y-direction.

4. Implementation

In the following, we present our algorithm from an implementation-specific view. We first give our assumptions and present our setup. We show that a weak motion model is important in continuous-time estimation and discuss the problems related to a prior in the context of line delay calibration. Finally, we propose our adaptive knot placement algorithm which enables precise calibration results despite a motion prior.

4.1. Assumptions and Setting

We assume that (i) the camera intrinsics and the distortion coefficients are known, (ii) the line delay is constant, (iii) the correspondences between the features, $\mathbf{y}_k^i$, and the landmarks, $\mathbf{p}_k$, are known, (iv) the geometry of the calibration pattern is known and with that the coordinates of the landmarks, $\mathbf{p}_k$.

The perspective pose problem is initialised using the GS PnP implementations available in OpenCV. Even if one should not expect the PnP solution to be a good pose estimate, it still proves to be of sufficient precision to serve as initial guess. A RS-specific PnP solver, such as proposed by Hedborg *et al*. [9], may be implemented in future. Furthermore, we assume a zero frame-delay to initialise the line delay as $d_0 = \frac{1}{fps}\frac{1}{N_R}$, where *fps* is the average number of Frames Per Second and N_R the number of rows of the sensor. The trajectory parameters, $\mathbf{x}(t)$, are described with a 4th order B-spline and initialised using the linear solution proposed by Schoenberg and Reinsch ([6]). We chose a 4th order B-spline as it is the first form which describes a continuous, jump-less motion. Additionally, the 2nd order derivative is well defined which allows us to implement a motion prior based on accelerations. Given the nonlinear least-squares problem it is preferable to use a minimal parametrisation for the rotation to avoid any constraints on the estimated quantities. A good choice is the axis angle representation which has a singularity at $2k\pi$, $k \in [0, 1, ...]$, and an ambiguity in the combination of the sign of the angle and the vector direction. For discrete time estimation the ambiguity is of less importance as one is mainly interested in the encoded rotation matrix. If however a continuous parametrisation is chosen changes occurring in the direction of the axis and the sign of the angle introduce jumps in the supposedly continuous trajectories. To cope with that problem we assume a jump-less motion and select, for every PnP estimate, the rotation vector which is, in a geometric sense, closest to the vector in the previous time-step. The vector at time step i ($\mathbf{v}^i$), given by an axis ($\overline{\mathbf{v}}^i = \frac{\mathbf{v}^i}{\|\mathbf{v}^i\|}$) and angle ($\phi^i = \|\mathbf{v}^i\|$), is compared to the set of possible equivalent parametrisations $\mathbf{v}_k^i = \overline{\mathbf{v}}^i \cdot (\phi^i + 2k_i\pi)$ with $k_i \in [K_l, K_u]$ generating a plausible range of rotations. At each discrete measurement step, the calculated parametrisation is compared to the previous rotation vector and k_i is chosen as:

$$k_i = \underset{k_j}{\operatorname{argmin}} \|\mathbf{v}_{k_j}^i - \mathbf{v}_{k_{i-1}}^{i-1}\|. \tag{22}$$

With this preprocessing we obtain continuous trajectories even if the motion includes rotations passing 2π.

Continuous-time estimation requires enough information in every time segment to estimate the spline coefficients, $\mathbf{c}$. A lack of measurements in one segment results in an under-constrained problem. To guarantee a well defined problem, even if no measurements are available, a weak motion prior is included [7]. We chose a prior of the form $\ddot{\boldsymbol{x}} \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}\delta(t-t'))$, with the covariance function $\mathbf{Q}\delta(t-t')$ and $\delta(\cdot)$ is Dirac's delta function. The zero-mean GP is the natural choice: in the absence of measurements, it is assumed that the sensor moves with minimum acceleration. The motion prior takes a visible influence only if no information on the pose is available. The weight, $\mathbf{Q} := diag\left(10^{-5}, 10^{-5}, 10^{-5}, 10^{-2}, 10^{-2}, 10^{-2}\right)$, was selected once and kept constant for all experiments. Figure 3

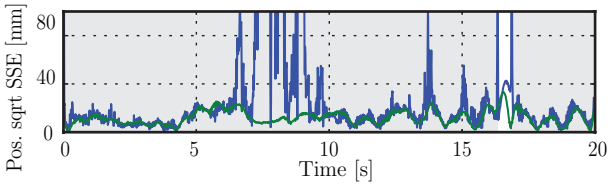

Figure 3. Influence of the motion prior on the position Sum of Squared Errors (SSE). The grey area shows the timestamps at which valid video data is available. The blue curve shows the position SSE for an unregularised estimate. The green curve shows a regularised estimate, which gives a more stable and less noisy estimate, especially in the regions with no measurements.

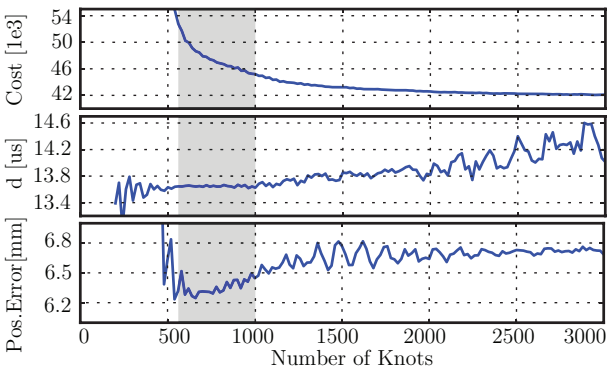

Figure 4. Optimisation results as a function of the number of uniformly spaced knots. The cost is constantly reduced by overfitting the reprojection errors while the estimated line delay starts to deviate from the nominal value. At the same time the RMS position error increases. The grey area shows the range in which the results are unbiased.

shows how a motion prior improves the stability of the estimated camera position in a general setup.

To solve the nonlinear least-squares problem, we used a DogLeg implementation as suggested by Lourakis and Argyros [13]. The sparse matrix equations were solved using the sparse Cholesky solver of CHOLMOD [5].

4.2. Adaptive Knot Placement

As explained in the previous section, the optimisation problem requires a weak motion prior to deal with measurement-poor time intervals. However, the motion prior can introduce a bias. Figure 4 shows the line delay estimates for an increasing number of uniformly spaced knots. As the line delay is a physical parameter, we expect its value to be independent of the motion parametrisation as long as enough representational power is available to accurately describe the motion. However, in Figure 4 we see that the line delay estimate stabilises between 600 and 1000 knots but then begins to diverge as more knots allow the curve to over fit. The same effect is seen for the RMS-position error (see Section 5), which starts to increase, while the estimator cost is constantly reduced. We conclude that the selection of an appropriate number of knots becomes important for line delay calibration.

We propose an adaptive knot placement method that avoids over-fitting by adaptively adjusting the number of knots until the residuals agree with their theoretical expected value. Each inverse-covariance-weighted error in (7) can be considered standardised, in the sense that the weighting whitens each error to follow a standard normal distribution ([4], chapter 5.4). This implies that ([4], section 1.4.17)

$$\mathbb{E}\left[\mathbf{e}_{k,i}^T \bar{\mathbf{R}}_{k,i}^{-1} \mathbf{e}_{k,i}\right] = \mathbb{E}\left[\tilde{\mathbf{e}}_{k,i}^T \tilde{\mathbf{e}}_{k,i}\right] = n, \tag{23}$$

where $\tilde{\mathbf{e}}_{k,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ and n is the dimension of the error term, in our case $n = 2$. Summing up, we obtain the expected cost in (7) to be

$$\mathbb{E}[J] = \sum_{i=0}^{N} \mathbb{E}\left[\mathbf{e}_{k,i}^T \bar{\mathbf{R}}_{k,i}^{-1} \mathbf{e}_{k,i}\right] \tag{24}$$

$$= 2N, \tag{25}$$

where N is the number of vector-valued reprojection error terms. To detect over-fitting, we can compare J, evaluated for a dataset, with its expected value. When $J > \mathbb{E}[J]$ for some segment of the trajectory, this implies that the spline does not have enough knots to be able to represent the motion in that segment. Consequently, the number of knots should be increased. This adaptive knot picking scheme was crucial for us to be able to estimate the line delay over a wide range of datasets.

4.3. Calibration Algorithm

Finally, we can outline our complete algorithm for line delay calibration:

1. Collect a video sequence of a known calibration pattern and extract keypoint measurements.
2. Initialise $d = \frac{1}{fps}\frac{1}{N_R}$
3. Initialise a smoothing spline using the GS PnP solution at each image time. The spline should have a uniform knot sequence with an approximate number of knots, preferably too few knots.
4. Solve problem (7) with $\boldsymbol{\theta} = d$.
5. For every segment j and with J_j the actual cost and N_j the number of error terms in the jth segment check $2N_j \leq J_j$, if True: split the segment in half by adding a knot at $t_{new} = \frac{1}{2}(t_{j-1} + t_j)$, where t_j is the position of the jth knot.
6. if new knots were added, go to 4)

5. Experiments

In this section, we describe three experiment types. For all three types we collected a series of datasets by moving the camera in front of a calibration pattern for 30–60s. The presented results were obtained with a *Matrix Vision BlueFOX IGC202dG*. The camera is attached to a large rig equipped with Vicon motion capture system markers for ground truth. The calibration process as well as the

perspective localisation datasets used chessboard corners, with known relative positions as landmarks, extracted using OpenCV. The transformation between the rig and the camera is estimated with Tsai's method [18] applied to a series of static images. The method proposed by Liu and Xia [19] is applied to synchronise both Vicon and camera timestamps. The camera-rig transformation and the time synchronisation are only required for validation against the ground truth and have no influence on the calibration or localisation tasks themselves. For the calibration of the intrinsics and the distortion coefficients of the RS camera the OpenCV tools were used. To prevent RS distortions, the camera was kept still but not mounted to a fixed stand.

5.1. Continuous-Time Localisation

Our first experiment is designed to show that our algorithm is capable of estimating the motion of a rolling shutter camera; simultaneous calibration of the line delay is not performed. The line delay is first calibrated using the approach proposed by Geyer *et al*. [14]. We chose a low framerate ($\approx$ 7*fps*) for which the line delay lies around $d = 137\mu s$. The extremely large line delay, around 4 times larger than for common smart-phones, produces significant distortions, even if only slow motion is produced. Comparing the results obtained with our approach to a GS discrete-time PnP solution, Figure 5, shows the importance of using an appropriate RS model. The Sum of Squared Errors (SSE) of the position and orientation[3] are significantly reduced and the reprojection errors confirm those results. The grey areas mark the time-ranges during which valid measurements were obtained. In accordance with the constant-acceleration hypothesis introduced by regularisation (Section 4.1) the estimated pose parameters remain bounded even if no measurements are obtained. A significant increase of the position error is however unpreventable as no information on the camera pose is available.

5.2. Shutter Calibration

The second experiment was designed to test the accuracy of our algorithm in estimating the line delay. We calibrated our camera with forty different real datasets at four different pixel-clock values with our approach and again with forty different datasets using the Geyer calibration. As this publication only presents a calibration method a thorough analysis of the required motion patterns for a successful calibration is left for future research. Our results are consistent with the results obtained with the established Geyer approach and both are in accordance with the nominal values. The numerical values are available in Table 1. The nominal values were obtained using the relation between the pixel clock (P [Hz]) and the time required for reading a single line containing N_P pixels given in the data sheet:

[3]to quantify the errors an Euler Roll-Pitch-Yaw parametrisation is used

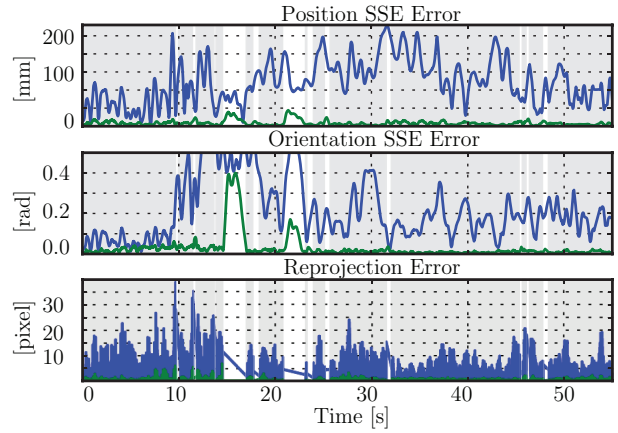

Figure 5. Continuous-time RS perspective localisation (green) and a discrete-time GS PnP solution (blue) compared to the Vicon ground truth. The grey areas show the time-ranges during which valid measurements were received. As expected, both solutions deviate from the ground truth if no measurements are available. The mean absolute errors for the GS and RS estimates, respectively, are 189 and 8.34 *mm* for the position, and 0.34 and 0.03 *rad* for the orientation.

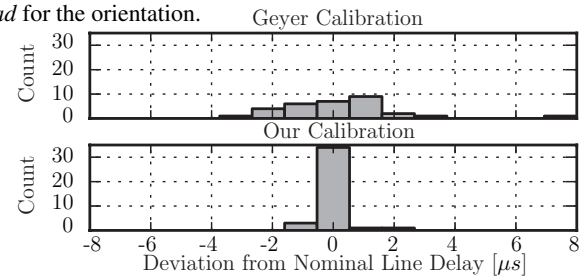

Figure 6. Deviation of the calibration results with our approach and the Geyer approach from the nominal values based on 40 real datasets at different values of pixel-clock.

$t_l = N_P \frac{1}{P}$. In our case the number of pixels including horizontal blanking is $N_P = 1650$. The numerical values in Table 1 and the error histogram in Figure 6 both confirm that our approach delivers line delay estimates close to the nominal values. As the nominal values might not be the camera's exact value we take an additional comparison step. We expect a linear relation between the line delay and the pixel timing (μs) passing through the origin. The results of a linear least squares fit of the line delays estimated by both methods are shown in Table 2. The behaviour of the line delay as a function of the pixel clock is more accurately described with our results. In conclusion, our method significantly outperforms the state-of-the-art technique in terms of reproducibility, while requiring a much simpler setup.

5.3. Smart-Phone Calibration Results

We performed a series of calibration runs for different smart-phones using our proposed method, see Table 3. Smart-phones typically have large exposure times. For that reason, we suggest using circle patterns for calibrating smart-phone cameras as blob detection algorithms are

	Nominal	Median		Std. Deviation	
		New	Geyer	New	Geyer
12MHz	137.5	137.7	136.86	0.56	3.12
20MHz	82.5	82.55	83.54	0.15	1.23
32MHz	51.56	51.35	51.48	0.19	1.70
40MHz	41.25	41.14	41.09	0.21	0.82

Table 1. Numerical values of the line delay estimation. Comparison between our approach, the Geyer calibration and the nominal values. The line delay units are μs.

	Nominal	New	Geyer
Slope	1650.0	1650.02	1642.54
Intercept	0	-0.0016	0.39
Residual	0	$3.90e^{-6}$	1.40

Table 2. Comparison of the nominal slope and intercept to a least squares fit of our results and the results with the Geyer calibration.

	fps	Resolution	d	σ	N
iPhone3GS	30	640×480	64.41	0.11	5
iPhone4S	30	1920×1080	24.12	0.52	12
Galaxy S3	30	1920×1080	30.25	0.68	7

Table 3. Calibration results for three common smart-phones. The line delay d and the standard deviation σ are given in μs. The estimates are based on N datasets.

in general more robust to motion blur than chessboard corner detectors. We used the detection algorithms offered by OpenCV 2.4. The length of the recorded video sequences is $50 - 60s$.

6. Conclusion and Future Work

In this paper, we derived a new method of estimating the line delay of a rolling shutter camera, using only video images of a chessboard or other calibration pattern with known geometry. The estimation is performed using the batch, continuous-time approach of [7]. The line delay calibration method outperforms the current state-of-the-art technique [14] without requiring any specialised hardware.

The complexity of the simultaneous calibration of the camera intrinsics and distortion parameters remains an open problem which should be addressed by further research. Besides that, a series of RS cameras influence the line delay by tuning the exposure while recording a video. We believe that our approach could be modified to an online line delay estimation to take full advantage of the sensors.

Acknowledgement

This work was supported by the EU project V-Charge (FP7-269916).

References

[1] J. B. A. Karpenko, D. Jacobs and M. Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. Technical report, Stanford University, march 2011.

[2] O. Ait-aider, N. Andreff, and J. M. Lavest. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *In Proc. European Conference on Computer Vision*, pages 56–68, 2006.

[3] S. Baker, E. P. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. 2010.

[4] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *State Estimation in Discrete-Time Linear Dynamic Systems*. John Wiley & Sons, Inc., 2002.

[5] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.*, 35(3):22:1–22:14, Oct. 2008.

[6] C. de Boor. A practical guide to splines. Applied Mathematical Sciences, chapter XIV. Springer, 2001.

[7] P. Furgale, T. Barfoot, and G. Sibley. Continuous-time batch estimation using temporal basis functions. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2088 –2095, may 2012.

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

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

[10] J. Hedborg, E. Ringaby, P.-E. Forssén, and M. Felsberg. Structure and motion estimation from rolling shutter video. In *ICCV Workshops*, pages 17–23, 2011.

[11] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[12] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *Proc. Eigth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*, Orlando, October 2009.

[13] M. Lourakis and A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1526 –1531 Vol. 2, oct. 2005.

[14] M. Meingast, C. Geyer, and S. Sastry. Geometric models of rolling-shutter cameras. *CoRR*.

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

[16] K. Shoemake. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph.*, 19(3):245–254, July 1985.

[17] 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.

[18] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *Robotics and Automation, IEEE Transactions on*, 5, jun 1989.

[19] L. Zhang, Z. Liu, and C. Honghui Xia. Clock synchronization algorithms for network measurements. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 160 – 169 vol.1, 2002.