Sensor-based camera tracking and video stabilization

Vos, B.J.

Award date:
2013

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science
Den Dolech 2, 5612 AZ Eindhoven
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
http://w3.win.tue.nl/nl/

Series title:
Graduation report,
Embedded Systems

Commissioned by Professor:
prof. dr. ir. G. de Haan

Group / Chair:
Electronic Systems

Date of graduation symposium:
March 19, 2013

Sensor-based camera tracking and video stabilization

by

Author: B.J. Vos

Internal supervisors: prof. dr. ir. G. de Haan

External supervisors: ir. F.J. de Bruijn
ir. W.P. Lee

Sensor-based camera tracking and video stabilization

Benji Vos
Eindhoven University of Technology
Eindhoven, The Netherlands
b.j.vos@student.tue.nl

***Abstract*—In applications that use signal detection from video recorded by mobile devices, two problems are identified: rolling shutter artifacts and shaky video. In this work we do an extensive evaluation on how to use sensor and image data to correct for both problems, i.e. to rectify the rolling shutter artifacts and stabilize the video. On the basis of our requirement to align all images in the video to a single reference frame, we design an evaluation metric that enables a fair comparison of different rectification and stabilization methods. This way we show that only employing sensor based rectification and stabilization has limitations in accuracy. The accuracy can be significantly improved using image data; our extension of a recent image based rectification and stabilization method produces better image alignment at the cost of higher complexity. We also present an improvement to a state-of-the-art rectification and stabilization method that uses both gyroscope and image data. This improved method shows similar results as our extended image based rectification and stabilization method, with the advantage, however, of a considerable reduction in complexity.**

I. Introduction

VIDEO cameras in mobile phones and tablets have become increasingly popular today because of their low price and portability. There are however two major problems in using these cameras for recording video sequences. First, compared to film cameras, mobile phones are usually significantly lighter. As a result, the recorded video sequences acquired on such a device suffer from frame to frame jitter due to camera shake. Second, most mobile device cameras use CMOS sensors [1] instead of CCD sensors due to cheaper manufacturing costs. In most CMOS sensor cameras, different rows in a frame are exposed sequentially from top to bottom. This is called *rolling shutter* (RS) readout, in contrast to *global shutter* (GS) readout, where an entire frame is acquired at once. This means that whenever there is relative fast motion between the device and scene, distortions occur. Fig.1 shows how an image is distorted when moving the camera rapidly from left to right during capture of a static scene using a rolling shutter sensor.

Since a range of applications use signal detection from video, there is a lot of interest in removing rolling shutter distortions and stabilizing the video at the same time. The inertial sensors (accelerometers and gyroscopes) together with magnetic sensors, which are present in most modern mobile devices, provide a method of doing this: using the orientation of the device during recording, the motion induced distortions can be compensated in a post-processing step.

(a) Original

(b) Rectified

Fig. 1. An example of a rolling shutter distortion (a) together with the rectified frame (b)

We limit this work to applications where a user tries to keep the camera as steady as possible in order to capture a fixed static scene. This scene is typically located at a distance of more than 1 m from the camera. Since there is a temporal signal at a fixed location within this scene, the detection of this signal is corrupted due to camera motion. In order to improve the signal detection, it is required to align all images to a single reference frame. The goal of this work is to evaluate different methods that use sensor and image data to remove camera motion and correct for rolling shutter artifacts in this defined application domain.

The contributions of this work are as follows:

- A recent image based video rectification and stabilization method of Ringaby et al. [2], which works with feature point correspondences in consecutive frames, is extended

to work with absolute feature point correspondences, i.e. feature point correspondences with respect to a reference frame. In the scope of this work, this extended method produces better image alignment compared to the original method.
- In the literature, one method employing both sensor and image data for the video rectification and stabilization is found. This method of Jia et al. [3] shows a flaw in some practical operating conditions. In this work, we present a solution to this flaw, resulting in a rectification and stabilization accuracy similar to our extension of Ringaby's method.
- On the basis of the requirement to align all images to a single reference frame, we design an evaluation method where physical markers are "hidden" in the scenes that are captured for benchmarking. Since these markers are easily tracked throughout the sequence, they are used to measure the similarity of each frame with respect to the reference frame. This enables a fair, numerical comparison of video rectification and stabilization methods.
- All discussed video rectification and stabilization methods are evaluated with our evaluation method using real sensor and image data.

The remainder of this work is organized as follows: In Section II an overview of related work is presented. A system overview is given in Section III. Section IV describes camera orientation estimation based on sensor and image data. In Section V the video rectification and stabilization method is described. Experiments and their results are presented in Section VI. Finally, Section VII gives a conclusion on the presented work followed by recommendations for future work in Section VIII.

II. Related Work

This work relates to multiple research topics, of which the two most important are sensor based orientation tracking and video rectification and stabilization. Related work is therefore categorized as follows.

A. Sensor based orientation tracking

Most modern mobile devices have incorporated inertial sensors (accelerometers and gyroscopes) and are provided with magnetic sensors for estimating the motion. In order to measure the motion in three dimensions, three-axis sensors consisting of 3 mutually orthogonal sensitive axes are required. A gyroscope incorporated in a mobile device, typically a microelectromechanical (MEMS) gyroscope, measures angular velocity. This angular velocity can be integrated over time to compute an orientation. However, due to the integration of measurement errors, this leads to an inaccurate orientation estimate. To improve this, one can use magnetic sensors (magnetometer) and accelerometers that measure earth's magnetic field and acceleration relative to free fall respectively. This way, both provide an absolute reference of orientation, earth's magnetic north and earth's gravity. However, magnetometers and accelerometers are likely to be subject to noise, for example, accelerations due to translations corrupt the measured direction of gravity. The process of combining the sensors into an optimal orientation estimate is called sensor fusion. The majority of sensor fusion methods to estimate orientation is based on Kalman filters [4]. The Kalman filter is the most popular data fusion algorithm. It is a recursive filter estimator, which estimates a state of a dynamic system based on its dynamics and incomplete or noisy measurements.

Some examples of orientation estimation based on Kalman filters are work of Luinge et al. [5], work of Foxlin [6], work of Marins et al. [7] and work of Törnqvist [8]. In work of Marins et al. [7] and work of Törnqvist [8] a quaternion representation to describe the coupled nature of orientations in three-dimensions is used. With quaternions, the problematic singularities associated with an Euler angle representation are avoided (Appendix A). Foxlin's work [6] provides a complementary bias filter that estimates gyroscope bias.

Besides Kalman filter based sensor fusion methods also other approaches exist, for example, work of Madgwick [9]. Madgwick's work describes a computational very efficient filter that uses a quaternion representation. In this filter the integration of gyroscope data is corrected each filter iteration based on accelerometer and magnetometer measurements. Furthermore, this filter incorporates magnetic distortion and gyroscope bias drift compensation.

B. Video rectification and stabilization

Most of the current mobile devices are not provided with optical or electrical image stabilization. This limits the scope of our work to digital image stabilization methods that incorporate the rolling shutter based image capture. We consider three classes of video rectification and stabilization methods within this context. Most methods are in the class of image based video rectification and stabilization methods. Very recently, however, some methods emerged in the class of sensor based video rectification and stabilization, and in the class that combines both sensor and image data for the rectification and stabilization.

1) Sensor based: In recent literature, two sensor based video rectification and stabilization methods are found. Hanning et al. present a method to use a Kalman filter based orientation estimator, which combines gyroscope and accelerometer data [8], for the video rectification and stabilization [10]. This method is based on a rotation-only camera model. Similar to this method, Karpenko et al. show satisfactory results by reconstructing the camera motion by integration of gyroscope data over time [11].

2) Image based: Image based video rectification and stabilization methods come in different variations. Some methods, which do rolling shutter rectification, model distortions as taking place in the image plane. Examples are work of Liang et al. [12] and work of Chun et al. [13]. Chun et al. assume that rectification can be done by an affine transformation, based on a constant motion across the image. In the method of Liang et al. each image row is given a different motion by interpolating a global inter-frame motion using a Bézier curve. Methods that only do video stabilization exist too. Some

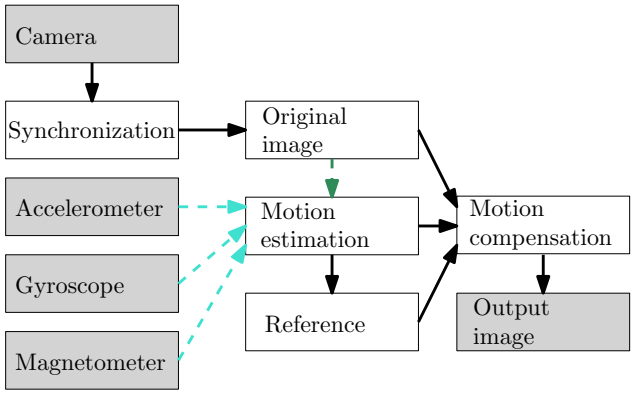


Fig. 2. An overview of the video rectification and stabilization system. Depending on the type of motion estimation (sensor/image based), the sensor data (– –), the image data (– –) or both are used in the *Motion estimation*. In the *Motion compensation* step the rectification and stabilization is done.

advanced video stabilization methods are based on structure-from-motion, for example, the method of Liu et al. [14]. Methods that do both rolling shutter rectification and video stabilization at the same time are limited. Two state-of-the-art methods are work of Ringaby et al [2] and work of Liu et al. [15]. Ringaby's work [2] uses a rotation-only rolling shutter camera model with image measurements from a feature tracker. Liu et al. [15] present a new stabilization approach based on subspace constraints on 2D feature trajectories. This method treats rolling shutter as noise.

3) Sensor and image based: In the literature, only one method that combines both sensor and image data for the video rectification and stabilization is found. This is the method of Jia et al. [3], using both gyroscope measurements and image data to rectify and stabilize the video. Similar to the approach of Hanning et al. [10], this method uses a rotation-only camera model.

III. System Overview

Fig.2 shows an overview of the complete rectification and stabilization system. This Section describes the model used for all different components of the complete system.

A. Camera

The camera is modeled using the pinhole camera model. The geometry related to the mapping is illustrated in Fig.3. In a pinhole camera, the relation between a 3D point in space $\mathbf{X} = (X, Y, Z)^T$ and its projection onto the 2D image plane, the homogeneous image point $\mathbf{x} = (x, y, z)^T$, is defined by:

$$\mathbf{x} = \mathbf{K}\mathbf{X} \text{ and } \mathbf{X} = \lambda \mathbf{K}^{-1}\mathbf{x} \tag{1}$$

where $\mathbf{K}$ is the *camera calibration matrix* and λ is a non-zero scale factor. $\mathbf{K}$ is estimated from images of a calibration pattern using Zhang's method [16] implemented in Matlab [17]. With the assumption of square pixels, the camera calibration matrix $\mathbf{K}$ has the following form:

$$\mathbf{K} = \begin{pmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

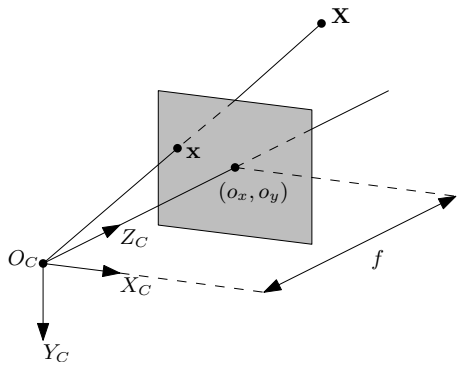
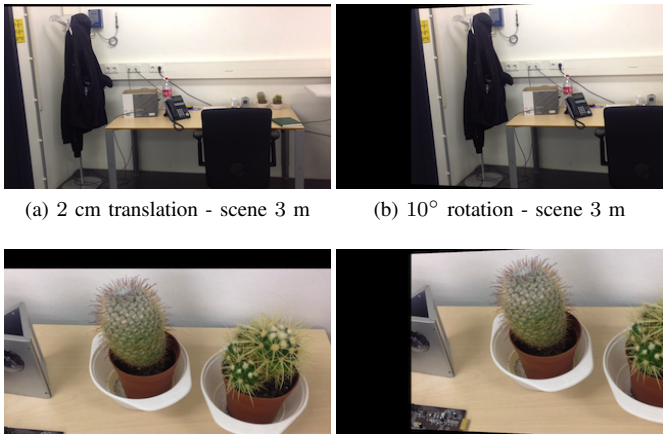


Fig. 3. Pinhole camera model. A ray from the camera center O_C to a point in the scene $\mathbf{X} = (X, Y, Z)^T$ intersects the image plane at $\mathbf{x} = (x, y)^T$. The relation between $\mathbf{X}$ and $\mathbf{x}$ is captured by $\mathbf{K}$ and depends on the focal length f and the location of the cameras axis (o_x, o_y) in the image plane.



(a) 2 cm translation - scene 3 m (b) 10° rotation - scene 3 m

(c) 2 cm translation - scene 40 cm (d) 10° rotation - scene 40 cm

Fig. 4. The effect of a camera translation and a camera rotation rotation on scenes at different distances. Case (a) and (c) show that the effect of a camera translation is different for scenes at different distances, while the effect of a camera rotation in (b) and (d) is independent of the scene's distance.

where (o_x, o_y) is the location of the camera axis in the image plane and f is the focal length.

B. Sensors

In order to estimate motion from a device, measurements of the current state of the system are needed. Most modern mobile devices are equipped with a three-axis accelerometer measuring acceleration relative to free fall $\mathbf{a} = (a_x, a_y, a_z)^T$, a three-axis gyroscope measuring angular velocity $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$, and a three-axis magnetometer measuring earth's magnetic field $\mathbf{m} = (m_x, m_y, m_z)^T$. We assume that the axes for the sensor measurements are similar to the axes of the pinhole camera (X_C, Y_C, Z_C) as defined in Fig.3.

C. Motion

Generally, motions can be described using rotations and translations. The effect of both on two different scenes is depicted in Fig.4. Based on the observation that objects at different depths move by different amounts as a result of a

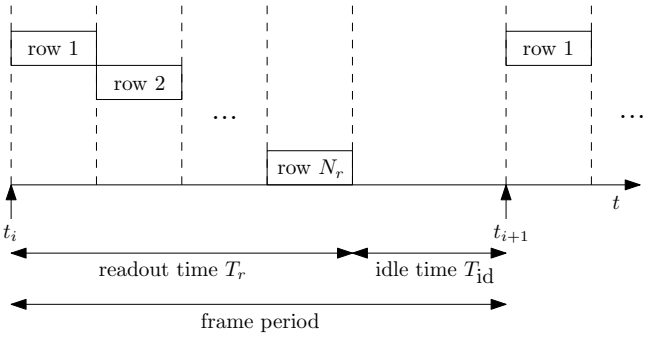


Fig. 5. Rolling shutter camera sequential row read out. Each row is read out at a different time instance. The total readout time of a single frame is represented by T_r. The time t_i represents the time instance when the image acquisition of frame i started, T_{id} represents the idle time and N_r represents the number of rows in the image.

camera translation, only camera rotations are modeled. This decision is supported in the commonly referenced comparative study of Forssen et al. [18]. In their work, better results are obtained by only modeling camera rotations compared to modeling both camera rotations and translations. Moreover, translations are also difficult to estimate using sensors. The only source of data to estimate translations, which is available in current mobile devices, is the accelerometer. However, to estimate the translations using the accelerometer, the measured acceleration requires double integration. This double integration includes the integration of measurement errors. In contrast, the measured angular velocity by gyroscopes only requires single integration to obtain the orientation. As a result, translation measurements are significantly less accurate than orientation measurements.

D. Rolling shutter model

In a rolling shutter camera, different rows are read out sequentially from top to bottom. This can be represented by assigning a time instance $t(\mathbf{x})$ to each point $\mathbf{x} = (x, y)$ in a frame, which can be calculated as:

$$t(\mathbf{x}) = t_i + T_r \frac{y}{N_r} \tag{3}$$

where t_i is the time instance when the image acquisition of frame i started, N_r is the number of rows in an image and T_r is the readout time as presented in Fig.5. The readout time T_r can be estimated by recording a flashing light source with a known frequency [2].

IV. Motion estimation

Video rectification and stabilization methods usually operate in two stages; motion estimation and motion compensation. In this Section we describe different methods to estimate the camera's motion. Since only camera rotations are modeled (Section III-C), motion estimation is considered as orientation estimation. The orientation estimation can be done based on sensor data, image data or based on both sensor and image data.

A. Sensor based

The goal of this work is to determine the feasibility of using sensor data for video rectification and stabilization in the context of our application domain. Sensor based approaches all have a low algorithm complexity in common due to the low number of data elements that need to be processed. This makes sensor based approaches very interesting for use on a mobile platform, which typically has limited processing power. Therefore, we choose to evaluate a number of different sensor based orientation estimators.

In the literature, two different sensor based orientation estimators are found, which are used in the context of video rectification and stabilization. The first orientation estimator, employed in Karpenko's work, uses integration of gyroscope data over time [11]. The second orientation estimator, employed in Hanning's work, estimates the camera's orientation with an Extended Kalman Filter (EKF)[10]. Since there are no limitations on the type of sensor fusion algorithm, and the motion estimation and motion compensation are independent steps, we also consider Madgwick's orientation filter [9]. Madgwick claims that better orientation results are obtained compared to Kalman filter based orientation estimation.

All discussed sensor based approaches have a unit quaternion representation of orientation in common. The unit quaternion representation is numerically more robust than, for example, Euler angles. In a unit quaternion representation, orientations can be described using an angle of rotation μ around a 3D axis of unit length $\mathbf{n}$ as depicted in Fig.6. More details about unit quaternions are presented in Appendix A.

1) Integrator: Karpenko et al. reconstruct the camera's orientation by integration of the measured angular velocity $\boldsymbol{\omega}_t$ [11]. Under the assumption that the angular velocity is constant during the sample intervals $\triangle t$, and $||\boldsymbol{\omega}_{t-1}||\triangle t$ is small, this integration can be done directly in quaternion representation [8] with the following equations:

$$\boldsymbol{\omega}_{q,t} = (0, \omega_{x,t}, \omega_{y,t}, \omega_{z,t})^T \tag{4}$$

$$\dot{\mathbf{q}}_{\text{int},\omega,t} = \frac{1}{2}\mathbf{q}_{\text{int},t-1} * \boldsymbol{\omega}_{q,t-1} \tag{5}$$

$$\mathbf{q}_{\text{int},t} = \mathbf{q}_{\text{int},t-1} + \dot{\mathbf{q}}_{\text{int},\omega,t}\triangle t \tag{6}$$

The integrator computes the quaternion derivative describing the rate of change in orientation as in (5). An orientation is obtained by integrating (6). This integration of the angular velocity is done with respect to a fixed system, the identity quaternion $q_{\text{int},0} = (1, 0, 0, 0)^T$ representing no rotation.

Apart from the low computational complexity, this method shows some drawbacks. First, the integration of gyroscope measurement errors causes drift of the orientation estimates. With only the (relative) gyroscope measurements, there is no way to correct for this drift. Second, there is no mechanism to correct for gyroscope bias, i.e. the measured angular velocity when the gyroscope is not undergoing any rotation.

2) Kalman filter: The Kalman filter used for orientation estimation in work of Hanning et al. [10] is based on work of Törnqvist [8]. The state in this filter is defined as the orientation in unit quaternion form $\mathbf{q}_{\text{kalman},t} = (q_{0,t}, q_{1,t}, q_{2,t}, q_{3,t})^T$. In Hanning's work only gyroscope data together with ac-

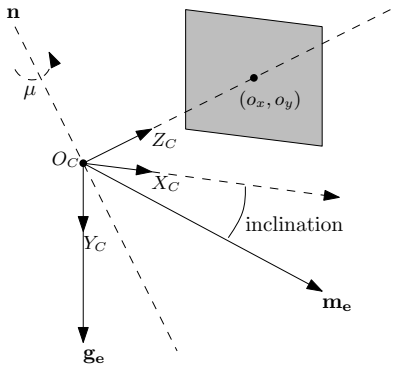


Fig. 6. Earth frame used in both the Kalman filter in Section IV-A2 and Madgwick's filter in Section IV-A3. The orientation estimated by both filters, is with respect to this earth frame. In the earth frame the y-axis is aligned with earth's gravity $\mathbf{g_e}$. The x-axis is aligned with the horizontal component of earth's magnetic north $\mathbf{m_e}$, which due to inclination of the earth, is not parallel to the earth's horizontal plane. The axis $\mathbf{n}$ and the angle of rotation μ describe an orientation with respect to this earth frame.

celerometer data are used, while the filter of Törnqvist incorporates gyroscope, accelerometer and magnetometer data.

Similar to the integration of gyroscope data in Section IV-A1, this Kalman filter estimates an orientation with respect to a fixed system. In this case however, the fixed system is the earth frame. The objective of the filter is to find an orientation of the sensor frame relative to the earth frame. Because the earth frame is fixed, obtaining an optimal orientation of the sensor frame relative to the earth frame implies obtaining an absolute orientation.

Finding such an absolute orientation is however quite difficult. By assuming the direction of an earth's field is known in the earth frame, a measurement of the field's direction within the sensor frame allows an orientation of the sensor frame relative to the earth frame to be calculated. However, for any given measurement, there is no unique orientation solution; all solutions represented by all those orientations obtained by the rotation of the true orientation around the axis parallel with the field are valid.

In this filter, gravity and the earth's magnetic field are used as known fields within the earth frame. The earth frame including both the earth's gravity vector $\mathbf{g_e}$ and earth's magnetic north $\mathbf{m_e}$ is presented in Fig.6. Gravity is measured using the accelerometer, which measures acceleration $\mathbf{a}$ relative to free fall. Under the assumption the user tries to keep the camera in a steady position, accelerations due to translations and rotations are ignored, so only the gravity component remains. Earth's magnetic field is measured using the magnetometer measurement $\mathbf{m}$.

As discussed above, using the measurement of gravity or earth's magnetic field alone does not provide a unique orientation of the sensor. However, both measurements can be combined to obtain a unique sensor orientation using the Kalman filter. In Appendix B the complete description of the used Kalman filter is given.

By incorporating the accelerometer data in addition to the gyroscope data, we can partially correct for the drift in orientation estimates compared to the integration of gyroscope data alone. This is done Hanning's work. A more accurate estimation could even be obtained by also incorporating the magnetometer data. The incorporation of magnetometer data reduces drift in the plane parallel to the earth's surface.

Besides the mechanism to correct for gyroscope drift, this Kalman filter based orientation estimator still shows some limitations. First, there is no mechanism to correct for gyroscope bias or magnetic distortions. Second, the assumption that translations or rotations are not measured with the accelerometer is not very realistic, camera shake inherently consists of accelerations due to translations and rotations.

3) Madgwick filter: The third sensor based orientation estimator under consideration is Madgwick's orientation filter [9]. Identical to the Kalman filter in Section IV-A2, this approach defines a state in unit quaternion form $\mathbf{q}_{\text{mw},t} = (q_{0,t}, q_{1,t}, q_{2,t}, q_{3,t})^T$. The state $\mathbf{q}_{\text{mw},t}$ represents the orientation of the sensor frame relative to the earth frame as depicted in Fig.6. Madgwick's orientation filter consists of the following equations, where $\boldsymbol{\omega}_{q,t}$ is defined in (4):

$$\dot{\mathbf{q}}_{\text{mw},\omega,t} = \frac{1}{2}\mathbf{q}_{\text{mw},t-1} * \boldsymbol{\omega}_{q,t-1} \quad (7)$$

$$\dot{\mathbf{q}}_{\text{mw},t} = \dot{\mathbf{q}}_{\text{mw},\omega,t} - \beta\dot{\mathbf{q}}_{\text{mw},\epsilon,t} \quad (8)$$

$$\mathbf{q}_{\text{mw},t} = \mathbf{q}_{\text{mw},t-1} + \dot{\mathbf{q}}_{\text{mw},t}\triangle t \quad (9)$$

Madgwick's orientation filter computes the estimated orientation rate in quaternion form as in (8). The filter computes (8) as the rate of change of orientation measured by the gyroscopes in (7) with the magnitude of the gyroscope measurement error, β, removed in the direction of the normalized estimated error $\dot{\mathbf{q}}_{\text{mw},\epsilon,t}$. The normalized estimated error $\dot{\mathbf{q}}_{\text{mw},\epsilon,t}$ is computed from accelerometer and magnetometer measurements. The estimated orientation is obtained by integration of (9). This algorithm is derived in [9] and uses the gradient descent algorithm to analytically derive the expressions in (7)-(9).

Next to the basic expressions describing Madgwick's filter, this filter also has the options to include on the fly magnetic distortion compensation and gyroscope bias drift compensation. This is an advantage compared to the Kalman filter in Section IV-A2, where no gyroscope bias compensation and magnetic distortion compensation is included.

There are also limitations of this filter. First, this filter uses the same assumption for the accelerometer measurements as in the Kalman filter; accelerations due to camera translations and rotations are ignored. Second, due to the fact that this filter is computational very efficient, the degree of freedom in usage is limited. In the basic configuration, without gyroscope bias correction, only the β parameter defines the operation of the filter. In this aspect the Kalman filter is more flexible; the noise terms associated with the gyroscope, accelerometer and magnetometer measurements control the operation of the filter.

B. Image based

Two experiments are performed in Section VI to determine the feasibility of using sensor data for the video rectification and stabilization. First, the orientation accuracy of all discussed sensor based orientation estimation methods is determined. Second, video rectification and stabilization is

evaluated using the discussed sensor based orientation estimation methods. The results of both experiments are presented in Section VI-A and Section VI-C respectively. For now it is sufficient to know that all sensor based orientation estimation methods have limited accuracy for the rectification and stabilization. To determine if the accuracy can be increased using image data, we also evaluate image based motion estimation methods.

In the literature, image based motion estimation methods are usually part of a stabilization method. These stabilization methods exist in different variations (Section II), however, methods doing both the video rectification and the video stabilization are limited. In this work we only consider one method that includes both the rectification and stabilization using a rotation-only camera model. This is the method of Ringaby at al. [2].

In Ringaby's method tracked features points extracted from the video are used to provide accurate geometric clues for the estimation of the camera's orientation. Their work describes a method to reconstruct the camera's orientation using non-linear least square optimization over inter-frame correspondences. In this work, Ringaby's method is referred to as the *relative* optimization method; feature point correspondences in consecutive frames are used for the optimization. As an extension of this relative optimization method, we design an *absolute* optimization method; feature points from a reference frame are matched with feature points in the current frame to reconstruct the camera's orientation.

In both the relative and the absolute optimization method, three parameters $\mathbf{r} = (r_1, r_2, r_3)^T$ are used to represent the orientation:

$$\mathbf{r} = \mu \mathbf{n} \tag{10}$$

where $\mathbf{n}$ represents the axis of rotation and μ the rotation angle as depicted in Fig.6. This results in fewer parameters that need to be resolved in the optimization step, while the same information is contained as with the quaternion representation. Also both methods require conversion to a rotation matrix and the other way around. This is done using the matrix logarithm and matrix exponent [2]:

$$\mathbf{r} = \text{logm}(\mathbf{R}) \tag{11}$$

$$\mathbf{R} = \text{expm}(\mathbf{r}) \tag{12}$$

Furthermore, in both methods the unit of time used for the rolling shutter model (Section III-D) is not expressed in seconds. Instead, for simplicity, the time is parameterized using the row number. This means that the inter-frame delay (idle time T_{id}) is expressed as a number of blank rows:

$$N_b = N_r(1 - T_r f) \tag{13}$$

where f is the video frame rate, N_r is the number of rows in the image and T_r is the readout time. By choosing time zero as the top row of the first frame, image point $\mathbf{x}_1 = (x, y)$ in frame one has time parameter $t(\mathbf{x}_1) = y$, where image point $\mathbf{x}_2 = (u, v)$ in the second frame has time parameter $t(\mathbf{x}_2) = N_r + N_b + v$.

1) Relative optimization: On the basis of feature point correspondences in consecutive frames, the relative optimization method of Ringaby et al. [2] reconstructs the camera's orientation. This is done by employing a sliding window mechanism that optimizes for short optimization intervals. This method is briefly described in Appendix C. A limitation of the relative optimization method is accumulation of errors; orientation errors resulting from the optimization step are propagated.

2) Absolute optimization: Aligning all images to a single reference frame does suffer from orientation errors being propagated throughout the image sequence. Therefore, it is interesting to extend the relative optimization method to work with absolute feature correspondences, i.e. feature point correspondences between a reference frame and the current frame.

Let the reference frame be indicated as frame 1. The absolute optimization method assumes that the reference frame contains no rolling shutter artifacts. We model this by fixing the camera's orientation during the read out interval of frame 1, as if the camera does not move during the image acquisition. This is equal to fixing the camera's orientation for the complete reference frame to $\mathbf{R} = \mathbf{I}$. With this reference frame, two points $\mathbf{x}_1$ and $\mathbf{x}_i$, that correspond in the reference frame and frame i, are expressed as:

$$\mathbf{x}_1 = \mathbf{K}\mathbf{X}_1 \text{ and } \mathbf{x}_i = \mathbf{K}\mathbf{R}(t(\mathbf{x}_i))\mathbf{X}_i \tag{14}$$

where $t(\mathbf{x}_i)$ is the time parameter for point $\mathbf{x}_i$. This gives the relation:

$$\mathbf{x}_1 = \mathbf{K}\mathbf{R}^T(t(\mathbf{x}_i))\mathbf{K}^{-1}\mathbf{x}_i \tag{15}$$

Each correspondence between the reference frame and the current frame, similar to (15), result in two equations where the unknowns are the rotations. Since the orientation of the reference frame is set to $\mathbf{I}$, this results in 3 unknowns. To restrict this number, the rotations are parameterized with an interpolating linear spline with a number of so called knots placed over an optimization interval. Within the optimization interval, intermediate rotations are found using spherical linear interpolation. Because the orientation of the reference frame is fixed, the minimum required optimization interval is one frame as depicted in Fig.7. As a result, with the interpolating spline, an optimization interval consisting of M knots has $3M$ unknowns.

The cost function that is minimized using non-linear least square optimization is defined as:

$$J_{\text{abs}}(\mathbf{r}_1, \ldots, \mathbf{r}_M) \tag{16}$$

where $\mathbf{r}_1, \ldots, \mathbf{r}_M$ represent the rotations belonging to knot $1, \ldots, M$. The function J_{abs} represents the (symmetric) image-plane residuals of the set of K corresponding points $\mathbf{x}_{1,k} \leftrightarrow \mathbf{x}_{i,k}$ in the reference frame and frame i [1]:

$$J_{\text{abs}} = \sum_k^K d(\mathbf{x}_{1,k}, \mathbf{H}\mathbf{x}_{i,k})^2 + d(\mathbf{x}_{i,k}, \mathbf{H}^{-1}\mathbf{x}_{1,k})^2 \tag{17}$$

$$\mathbf{H} = \mathbf{K}\mathbf{R}^T(t(\mathbf{x}_{i,k}))\mathbf{K}^{-1} \tag{18}$$

[1]This function only works for an optimization interval of one frame. However, by also incorporating feature point correspondences between the reference frame and other frames, the extension to longer optimization intervals is trivial.

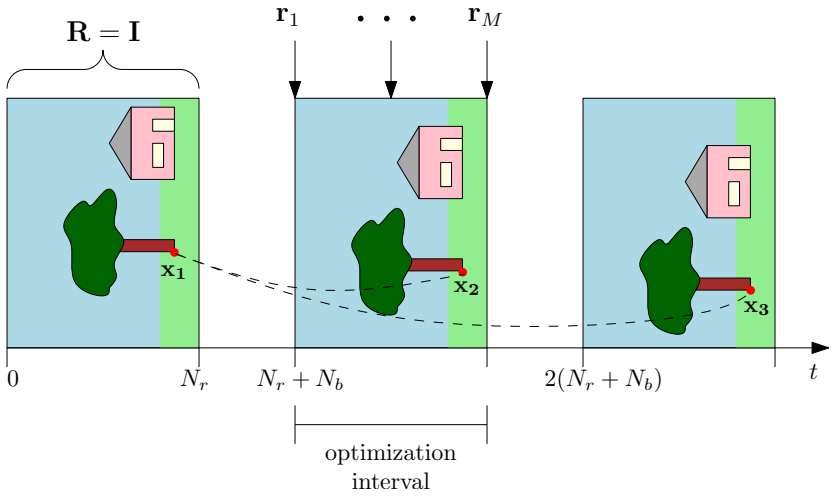


Fig. 7. Optimization interval for the absolute optimization method consisting of one frame using $M = 3$ knots. Rotations $\mathbf{r}_1, \ldots, \mathbf{r}_M$ are found with non-linear least square optimization. The readout time is represented using the number of rows N_r and the idle time is represented using the number of blank rows N_b. Because this method assumes that the reference frame contains no rolling shutter artifacts, the orientation for the complete reference frame is fixed to $\mathbf{I}$. An example feature correspondence, which is used for the optimization, is indicated with the dashed line between $\mathbf{x}_1$ and $\mathbf{x}_2$ or between $\mathbf{x}_1$ and $\mathbf{x}_3$. After rotations $\mathbf{r}_1, \ldots, \mathbf{r}_M$ for the first optimization interval starting from $N_r + N_b$ are found, the optimization interval shifts to the start of frame 2, which is $2(N_r + N_b)$.

and where the Euclidian distance function $d(\mathbf{x}, \mathbf{y})$ for (homogeneous) vectors $\mathbf{x} = (x, y, z)^T$ and $\mathbf{y} = (u, v, w)^T$ is defined as:

$$d(\mathbf{x}, \mathbf{y})^2 = (x/z - u/w)^2 + (y/z - v/w)^2 \tag{19}$$

Each optimization interval consists of M knots, similar to Fig.7. The position of a knot is called the knot time, which is a time instance. Let the time instance of knot m, be indicated with N_m. As mentioned above, the intermediate rotations are found using spherical linear interpolation (SLERP)[2], where the evaluation at a row with time parameter N_{curr} is denoted by:

$$\mathbf{R} = \text{SLERP}(\{\mathbf{r}_m, N_m\}_1^M, N_{\text{curr}}) \tag{20}$$

The rotations $\mathbf{r}_1, \ldots, \mathbf{r}_M$ that minimize (16) are found with non-linear least square optimization for a single optimization interval. To initialize a new optimization interval from the previous one, the following procedure is followed.

1) Shift the optimization interval one frame, by re-sampling the knots $\{\mathbf{r}_m\}_1^M$ with an offset of $N_r + N_b$:

$$\mathbf{R}_m = \text{SLERP}(\{\mathbf{r}_m, N_m\}_1^M, N_m + N_r + N_b) \tag{21}$$

2) Use the three parameter representation of orientation for $m = 1, \ldots, L$:

$$\mathbf{r}_m^{'} = \text{logm}(\mathbf{R}_m) \tag{22}$$

where N_L is the last time inside the optimization interval. Newly shifted-in rotations are copied from the last valid knot $\mathbf{r}_L^{'}$.

Compared to the relative optimization method, this method does not suffer from accumulation of orientation errors. The applicability of this method is however limited. In contrast to finding a camera orientation for all time instances, which is done in the relative optimization method, this absolute optimization method finds the camera orientations that map each frame to a reference frame in the least square sense.

C. Sensor and image based

In this work, experiments are performed to determine the feasibility of using sensor data or image data for the video rectification and stabilization. These experiments, together with the results, are presented in Section VI. For now it is sufficient to know that: 1) Video rectification and stabilization based on only sensor data has limited accuracy. 2) Compared to sensor based video rectification and stabilization, image based video rectification and stabilization offers better image alignment. However, this is at the cost of significantly higher algorithm complexity. To determine if we can combine the low complexity of sensor based orientation estimation with the high accuracy of image based orientation estimation, we also evaluate orientation estimation methods based on both sensor and image data.

Most existing algorithms in the literature, which deploy sensor and image data together, assume a unique camera pose for each complete frame, as if the image were taken using a global shutter camera. This assumption cannot be maintained for rolling shutter cameras. Yet, to this time one state-of-the-art method is found that incorporates the rolling shutter based image capture into an orientation estimation method employing both visual and inertial measurements. This is the method of Jia et al. [3], combining gyroscope measurements with feature point correspondences into an Extended Kalman Filter (EKF).

Jia's method uses the same rolling shutter rotation-only camera model as described in Section III. By correcting the gyroscope measurements using matched feature points, the drift caused by integration of measurement errors can be removed. Comparing this with the sensor based orientation estimation methods, where gyroscope drift is removed using

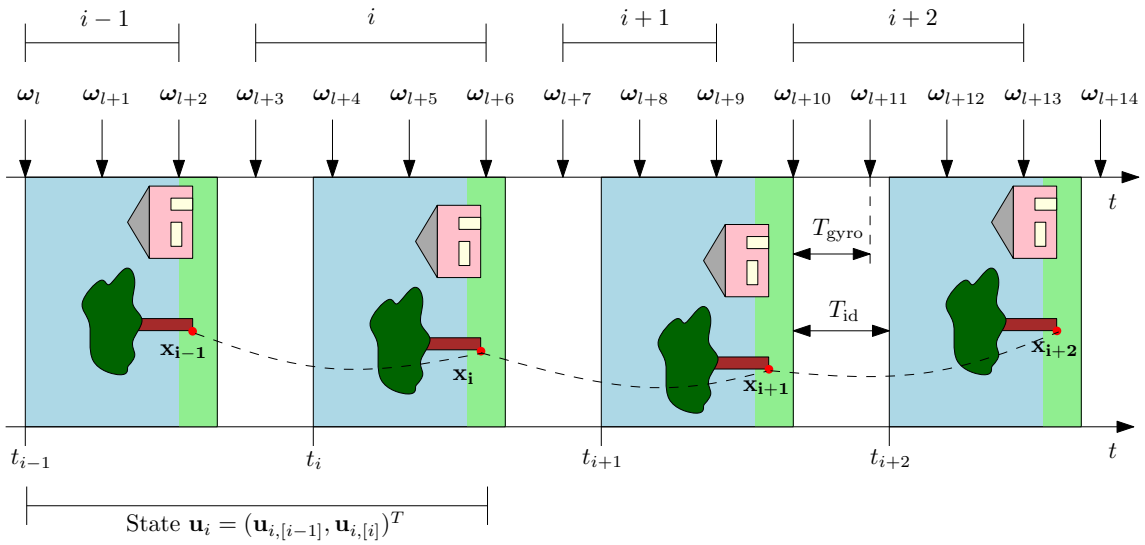


Fig. 8. Gyroscope measurements $\boldsymbol{\omega}_l, \ldots \boldsymbol{\omega}_{l+14}$ together with frame timestamps $t_{i-1}, \ldots t_{i+2}$. Gyroscope measurements $\boldsymbol{\omega}_l, \ldots, \boldsymbol{\omega}_{l+2}$ are related to frame $i-1$, while gyroscope measurements $\boldsymbol{\omega}_{l+3}, \ldots, \boldsymbol{\omega}_{l+6}$ are related to frame i. The relative camera rotation between two image points $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ in consecutive frames can be calculated using the gyroscope measurements as in (24). In this example, the assumption of Jia's method [3] is not violated, since the idle time T_{id} is larger than the gyroscope sample period T_{gyro}. Therefore, grouping the gyroscope measurements by frames is allowed.

accelerometer or magnetometer data, Jia et al. claim that matched feature points offer a higher accuracy.

However, depending on the parameters of the mobile platform, Jia's method contains a false assumption; they assume that *the gyroscope sampling period T_{gyro} is smaller than the idle time T_{id}*. In order to understand how this assumption affects their method, we need to discuss their method in more detail. In the Jia's method matched feature points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ in consecutive frames i and $i+1$, can be related using the relative rotation between them:

$$\mathbf{x}_{i+1} = \mathbf{K}\mathbf{R}\left(t(\mathbf{x}_{i+1})\right)\mathbf{R}^T\left(t(\mathbf{x}_i)\right)\mathbf{K}^{-1}\mathbf{x}_i \tag{23}$$

where $t(\mathbf{x}_i)$ and $t(\mathbf{x}_{i+1})$ are timestamps for point $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ respectively.

The relative rotation in (23) is similar to the relation described in (71), which is used by the relative optimization method of Ringaby et al. [2], in Appendix C. In contrast to Ringaby's method, where rotations are found using non-linear least square optimization, Jia et al. compute this relative rotation using gyroscope measurements.

In Fig.8 several gyroscope readings are grouped together into so called angular velocity groups since they are used to compute the camera rotations for the same frame during its corresponding read out time. Fig.8 also shows that the gyroscope sampling frequency is higher than the video frame rate; usually around 50 to 100 Hz. Let the timestamp of gyroscope reading $\boldsymbol{\omega}_l$ be indicated with $t(\boldsymbol{\omega}_l)$. Under the assumption that the idle time T_{id} is large enough so that no pixels from frame $i-1$, but only several pixels in frame i are read after $t(\boldsymbol{\omega}_{l+3})$, $\boldsymbol{\omega}_{l+3}$ is related to frame i. Note that this assumption is similar to stating that $T_{gyro} \leq T_{id}$.

Let the point $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ be matched feature points in frame $i-1$ and i, with corresponding 3D points $\mathbf{X}_{i-1}$ and $\mathbf{X}_i$ respectively. Without loss of generality, assume that $t(\mathbf{x}_{i-1})$ is between $t(\boldsymbol{\omega}_{l+2})$ and $t(\boldsymbol{\omega}_{l+3})$ and $t(\mathbf{x}_i)$ is between $t(\boldsymbol{\omega}_{l+5})$ and $t(\boldsymbol{\omega}_{l+6})$ as indicated in Fig.8. Under the assumption that the angular velocity is constant during its sample interval, the relative rotation relating the points $\mathbf{X}_{i-1}$ and $\mathbf{X}_i$, in terms of gyroscope measurements is:

$$\mathbf{R}(t(\mathbf{x}_i))\mathbf{R}^T(t(\mathbf{x}_{i-1})) = \prod_{n=l+2}^{l+5} \triangle\mathbf{R}(\boldsymbol{\omega}_n \triangle t_n) \tag{24}$$

where $\triangle t_{l+2}$ to $\triangle t_{l+5}$ are equal to $t(\boldsymbol{\omega}_{l+3}) - t(\mathbf{x}_{i-1})$, $t(\boldsymbol{\omega}_{l+4}) - t(\boldsymbol{\omega}_{l+3})$, $t(\boldsymbol{\omega}_{l+5}) - t(\boldsymbol{\omega}_{l+4})$ and $t(\mathbf{x}_i) - t(\boldsymbol{\omega}_{l+5})$ respectively. Each sub-relative rotation matrix is computed by:

$$\triangle\mathbf{R}(\boldsymbol{\omega}_n \triangle t_n) = \text{expm}(\boldsymbol{\omega}_n \triangle t_n) \tag{25}$$

With the relative rotation expressed in terms of gyroscope measurements in (24), two angular velocity groups $i-1$ and i, are enough to represent the relative rotation matrix between any pair of matching feature points in frame $i-1$ and frame i. Because of this observation, Jia et al. define the state of the used Extended Kalman Filter as:

$$\mathbf{u}_i = \begin{pmatrix} \mathbf{u}_{i,[i-1]} \\ \mathbf{u}_{i,[i]} \end{pmatrix} \tag{26}$$

where $\mathbf{u}_{i,[i-1]}$ and $\mathbf{u}_{i,[i]}$ correspond to angular velocity groups for frame $i-1$ and i respectively.

The goal of the state at time step i defined in (26) for the Extended Kalman Filter in Jia's method is to estimate the angular velocities associated with frame $i-1$ and frame i. By using the gyroscope readings of a group as control inputs, the state can be predicted as:

$$\mathbf{u}_i = \begin{pmatrix} \mathbf{u}_{i,[i-1]} \\ \mathbf{u}_{i,[i]} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{i-1,[i]} \\ \mathbf{y}_{[i]} \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{v}_{[i]} \end{pmatrix} \tag{27}$$

where $\mathbf{y}_{[i]}$ are the gyroscope readings associated with frame i with normally-distributed measurement noise $\mathbf{v}_{[i]}$.

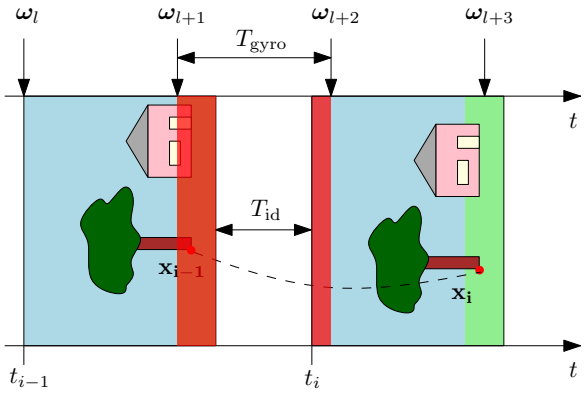


Fig. 9. Gyroscope measurements $\boldsymbol{\omega}_l, \ldots \boldsymbol{\omega}_{l+3}$ together with frame timestamps t_{i-1} and t_i. In this example, the assumption that $T_{\text{gyro}} \leq T_{\text{id}}$ holds, is violated. Both pixels in frame $i-1$ and in frame i are read out after $t(\boldsymbol{\omega}_{l+1})$ and before $t(\boldsymbol{\omega}_{l+2})$ (indicated in red). Therefore, it is not clear whether $\boldsymbol{\omega}_{l+1}$ should be related to frame $i-1$ or to frame i; in both frames $\boldsymbol{\omega}_{l+1}$ is needed to compute the relative rotation for pixels marked in red.

The feature point correspondences between frames $i-1$ and i are used to correct the angular velocities included in the state variable $\mathbf{u}_i$. The k-th feature point $\mathbf{x}_{i-1,k}$ in frame $i-1$ relates to point $\mathbf{x}_{i,k}$ in frame i with the following relation:

$$\mathbf{x}_{i,k} = g\left(\mathbf{K}\triangle\mathbf{R}\mathbf{K}^{-1}\begin{pmatrix}\mathbf{x}_{i-1,k}+\mathbf{e}_{i-1,k,1}\\ 1\end{pmatrix}\right)+\mathbf{e}_{i,k,2} \quad (28)$$

where g is the function to convert a 3D (homogeneous) vector into a 2D (inhomogeneous) vector and $\mathbf{e}_{i-1,k,1}$ and $\mathbf{e}_{i,k,2}$ represent the normally-distributed measurement noise in feature point detection for $\mathbf{x}_{i-1,k}$ and $\mathbf{x}_{i,k}$ respectively. Similar to (24), the relative rotation $\triangle\mathbf{R}$ is expressed, in terms of the state $\mathbf{u}_i$, as:

$$\prod_{l=1}^{N_i}\triangle\mathbf{R}(\boldsymbol{\omega}_l\triangle t_{k,l}) \quad (29)$$

where N_i is the number of angular velocities in the state $\mathbf{u}_i$. The duration time $\triangle t$ for each angular velocity can be computed in the same way as in (24). Note that some of them can be zero. In order to correct the state, which is predicted based on the gyroscope measurements in (27), the entire measurement equation is expressed as:

$$\mathbf{z}_i = \begin{pmatrix}\mathbf{x}_{i,1}\\ \mathbf{x}_{i,k}\\ \vdots\\ \mathbf{x}_{i,K}\end{pmatrix} = \begin{pmatrix}h_1(\mathbf{u}_i,\mathbf{x}_{i-1,1},\mathbf{e}_{i-1,1,1},\mathbf{e}_{i,1,2})\\ h_k(\mathbf{u}_i,\mathbf{x}_{i-1,k},\mathbf{e}_{i-1,k,1},\mathbf{e}_{i,k,2})\\ \vdots\\ h_K(\mathbf{u}_i,\mathbf{x}_{i-1,K},\mathbf{e}_{i-1,K,1},\mathbf{e}_{i,K,2})\end{pmatrix} \quad (30)$$

where $h_k()$ is defined as (28) and K is the number of feature point correspondences between frame $i-1$ and frame i. After linearization, the Extended Kalman Filter can now be applied to (27) and (30) to obtain an optimal angular velocity for the complete image sequence. By integration of this estimated angular velocity in the quaternion domain (Section IV-A1), the estimated camera orientation is obtained. The complete derivation of this method can be found in [3].

As previously suggested, the assumption that $T_{gyro} \leq T_{id}$ holds, appears invalid on some mobile platforms. The mobile platform in our experiments shows values of T_{id} in the range of 1 ms to 8 ms[2]. With these values of T_{id}, the required gyroscope sampling frequency should be in the range of 125 to 1000 Hz in order to satisfy the assumption. However, since typical gyroscope sampling frequencies are in de range of 50 to 100 Hz, it is very likely that the assumption in Jia's method is violated. In that case, there is an angular velocity that maps to pixels in two consecutive frames as depicted in Fig.9. This angular velocity, $\boldsymbol{\omega}_{l+1}$ in Fig.9, maps to pixels in both frames $i-1$ and i. With the grouping of angular velocities per frame, it is not clear if $\boldsymbol{\omega}_{l+1}$ should be related to frame $i-1$ or frame i; in both frames $i-1$ and i, $\boldsymbol{\omega}_{l+1}$ is needed to compute the relative rotation for pixels marked in the red area. This results in an undefined computation of the relative rotation in (29), for pixels marked in red, when $\boldsymbol{\omega}_{l+1}$ is not included in the state defined in (26).

Jia's method does not deal with this specific case of $T_{\text{gyro}} > T_{\text{id}}$. By changing the state prediction equation in (27), there is however a way to solve this. We introduce this solution in the following Section.

1) Improvement: In Jia's method, angular velocities are grouped by frames, since they are used to compute the camera rotations for the same frame during its corresponding read out time. With the assumption of $T_{\text{gyro}} \leq T_{\text{id}}$ this is allowed. However, with the practical case that the gyro sample period T_{gyro} can be larger than T_{id}, this results in problems as illustrated in Fig.9.

By relaxing the constraint that angular velocities are only associated to a single frame, we can come up with a simple solution such that the relative rotation between any two points in two consecutive frames, is still defined in terms of the state. In Fig.10 an example is presented of our new improved method of grouping the angular velocities. In our improved method, one angular velocity can map to two frames, e.g. $\boldsymbol{\omega}_{l+3}$ and $\boldsymbol{\omega}_{l+5}$ map to pixels in two different frames. On the basis of this grouping, we redefine the state of the Kalman filter as:

$$\mathbf{u}_i = \begin{pmatrix}\mathbf{u}_{i,[i-2,i-1]}\\ \mathbf{u}_{i,[i-1]}\\ \mathbf{u}_{i,[i-1,i]}\\ \mathbf{u}_{i,[i]}\\ \mathbf{u}_{i,[i,i+1]}\end{pmatrix} \quad (31)$$

where the sub groups of the state $\mathbf{u}_{i,[i-2,i-1]}, \ldots \mathbf{u}_{i,[i,i+1]}$ represent angular velocities associated to the frames between [and], i.e. $\mathbf{u}_{i,[i-2,i-1]}$ is the angular velocity which maps to both frames $i-2$ and $i-1$. Note that some of these groups can be empty or have at most one element. In that case, the state $\mathbf{u}_i$ has a lower number of elements.

The goal of the state at time step i defined in (31) in our improved method is to estimate the optimal angular velocities associated with frame $i-1$ and frame i. Similar to (27), the angular velocities are used as control inputs. The state in our

[2]Measured with an iPad 2 and the iPad 3 using a 720p resolution at 29.97 fps.

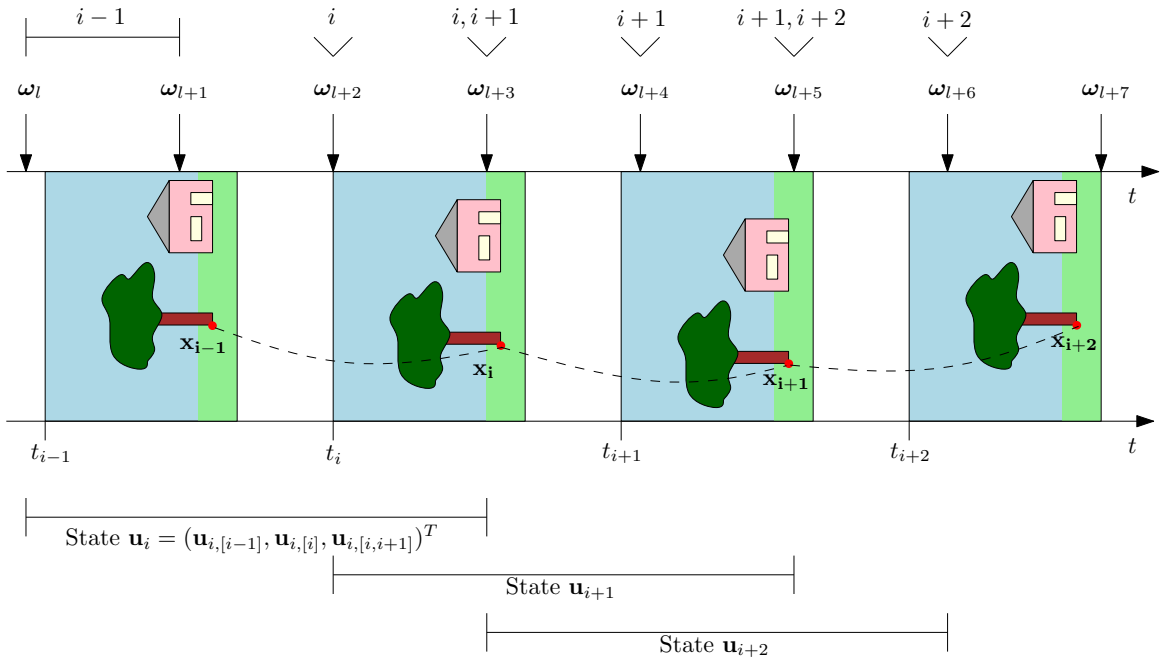


Fig. 10. Gyroscope measurements $\boldsymbol{\omega}_l, \ldots \boldsymbol{\omega}_{l+7}$ together with frame timestamps $t_{i-1}, \ldots t_{i+2}$. The gyroscope measurements are grouped as indicated by the labels above each gyroscope measurement. With this grouping and the state as defined in (31), the relative camera rotation between two image points $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ in consecutive frames, can be calculated in terms of the state as in (29).

improved method can be predicted as:

$$\mathbf{u}_i = \begin{pmatrix} \mathbf{u}_{i,[i-2,i-1]} \\ \mathbf{u}_{i,[i-1]} \\ \mathbf{u}_{i,[i-1,i]} \\ \mathbf{u}_{i,[i]} \\ \mathbf{u}_{i,[i,i+1]} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{i-1,[i-1,i]} \\ \mathbf{u}_{i-1,[i]} \\ \mathbf{u}_{i-1,[i,i+1]} \\ \mathbf{y}_{[i]} \\ \mathbf{y}_{[i,i+1]} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathbf{v}_{[i]} \\ \mathbf{v}_{[i,i+1]} \end{pmatrix} \tag{32}$$

here $\mathbf{y}_{[i]}$ and $\mathbf{y}_{[i,i+1]}$ represent the gyroscope readings associated to only frame i, and to both frames i and $i+1$, respectively. The terms $\mathbf{v}_{[i]}$ and $\mathbf{v}_{[i,i+1]}$ represent the normally-distributed measurement noise of both groups of gyroscope readings. In Fig.10 an example is given of the angular velocities that are associated to three consecutive states $\mathbf{u}_i$, $\mathbf{u}_{i+1}$ and $\mathbf{u}_{i+2}$. Apart from the modified state in (31) and the modified state prediction expression in (32), no other changes are needed to Jia's method.

This improvement to Jia's method shows a simple and effective way to deal with cases where $T_{\text{gyro}} > T_{\text{id}}$; our improvement proposes a new method of grouping angular velocities as illustrated in Fig.10. With our new state $\mathbf{u}_i$ as defined in (31) the relative rotation between any two corresponding points in frames $i-1$ and i can be computed from the state. Since this was not the case in Jia's original method, our improved method is more robust to practical operating conditions.

V. Video rectification and stabilization

The second step after the motion estimation is motion compensation, which means in this context, rolling shutter correction and video stabilization. All orientation estimators in Section IV provide the orientation in unit quaternion form or can be easily converted to this representation. Therefore, we use the quaternion based video rectification and stabilization method of Hanning et al. [10].

A. Rolling shutter correction

The rolling shutter rotation-only camera model, which is described in Section III, requires a camera orientation for all time instances during the image sequence. We obtain this camera orientation at time $t(\mathbf{x})$ by interpolating the orientation estimates in unit quaternion form. Let $\mathbf{q}_1$ and $\mathbf{q}_2$ be two consecutive orientation estimates belonging to times t_1 and t_2 such that $t_1 \leq t(\mathbf{x}) \leq t_2$. The orientation at time instance $t(\mathbf{x})$ is found using spherical linear interpolation (SLERP) [19]:

$$\mathbf{q}(\mathbf{x}) = \text{SLERP}(\mathbf{q}_1, \mathbf{q}_2, \frac{t(\mathbf{x}) - t_1}{t_2 - t_1}) \tag{33}$$

By having orientation estimates belonging to every row in each frame, correction for rolling shutter distortions is possible. Let $\mathbf{q}_m$ be the camera orientation for the middle row of frame i. The rotation from the row containing $\mathbf{x}$ to the middle row is:

$$\mathbf{q}_m(\mathbf{x}) = \mathbf{q}(\mathbf{x})^{-1} * \mathbf{q}_m \tag{34}$$

For each point in the image it's now possible to calculate it's position during the acquisition of the middle row in the frame:

$$\mathbf{X} = \lambda \mathbf{K}^{-1} \mathbf{x} \tag{35}$$

$$\begin{pmatrix} 0 \\ \mathbf{X}' \end{pmatrix} = \mathbf{q}_m(\mathbf{x}) * \begin{pmatrix} 0 \\ \mathbf{X} \end{pmatrix} * \mathbf{q}_m(\mathbf{x})^{-1} \tag{36}$$

$$\mathbf{x}' = \mathbf{K}\mathbf{X}' \tag{37}$$

here $\mathbf{x}'$ is the new, rectified position of $\mathbf{x}$. This technique aligns all rows to the orientation of the middle row. Another reference row can however be obtained by replacing $\mathbf{q}_m$ in (34).

B. Video stabilization

Video acquired on a mobile device suffers from frame to frame jitter due to camera shake. Therefore, the orientation estimates from Section IV vary for each individual frame. In order to remove these rapid changes in orientation, the camera rotations can be removed by using a single reference camera orientation $\mathbf{q}_{\text{ref}}$. This is different from other work that applies a low pass filter to smooth the camera rotations [10], [11]. The stabilized image point is calculated as:

$$\mathbf{q}_{\triangle} = \mathbf{q}_m^{-1} * \mathbf{q}_{\text{ref}} \tag{38}$$

$$\begin{pmatrix} 0 \\ \mathbf{X}'' \end{pmatrix} = \mathbf{q}_{\triangle} * \begin{pmatrix} 0 \\ \mathbf{X}' \end{pmatrix} * \mathbf{q}_{\triangle}^{-1} \tag{39}$$

$$\mathbf{x}'' = \mathbf{K}\mathbf{X}'' \tag{40}$$

here $\mathbf{x}''$ is the stabilized and rectified position of $\mathbf{x}$ using a reference orientation $\mathbf{q}_{\text{ref}}$. When aligning the video to the first frame, $\mathbf{q}_{\text{ref}}$ should be the orientation of the middle row of the first frame.

C. Synchronization

Whenever a sensor based orientation estimation is employed, synchronization to the image data is required. Both sensor and image data are timestamped, but there is an unknown delay between the two. This delay is represented by the term T_d added to (3):

$$t(\mathbf{x}) = t_i + T_d + T_r \frac{y}{N_r} \tag{41}$$

To find T_d, we use a number of K feature point correspondences for each pair of F consecutive frames. The rotation from $\mathbf{x}_{i+1}$ to $\mathbf{x}_i$, which are corresponding points in frame i and frame $i+1$, is expressed as:

$$\mathbf{q}_i = \mathbf{q}(\mathbf{x}_{i+1})^{-1} * \mathbf{q}(\mathbf{x}_i) \tag{42}$$

where $\mathbf{q}(\mathbf{x}_i)$ and $\mathbf{q}(\mathbf{x}_{i+1})$ are the orientations of the camera at time instances $t(\mathbf{x}_i)$ and $t(\mathbf{x}_{i+1})$ obtained by (33). The position of $\mathbf{x}_{i+1}$ at time instance $t(\mathbf{x}_i)$ is calculated as:

$$\mathbf{X}_{i+1} = \lambda \mathbf{K}^{-1} \mathbf{x}_{i+1} \tag{43}$$

$$\begin{pmatrix} 0 \\ \mathbf{X}'_{i+1} \end{pmatrix} = \mathbf{q}_i * \begin{pmatrix} 0 \\ \mathbf{X}_{i+1} \end{pmatrix} * \mathbf{q}_i^{-1} \tag{44}$$

$$\mathbf{x}'_{i+1} = \mathbf{K}\mathbf{X}'_{i+1} \tag{45}$$

here $\mathbf{x}'_{i+1}$ is the position of $\mathbf{x}_{i+1}$ at time instance $t(\mathbf{x}_i)$. To synchronize the sensor and image data a cost function is defined as:

$$J_{\text{sync}} = \sum^{F} \sum_{k}^{K} d(\mathbf{x}_{i,k}, \mathbf{x}'_{i+1,k})^2 \tag{46}$$

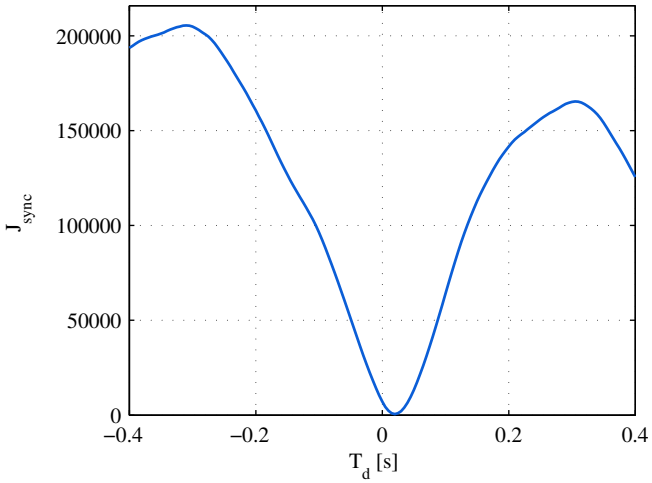


Fig. 11. The cost function J_{sync} in (46) evaluated for different values of T_d. The minimum J_{sync} is at $T_d = 0.01$.

where $d(\mathbf{x}, \mathbf{y})$ is the Euclidian pixel distance between $\mathbf{x}$ and $\mathbf{y}$ as defined in (19). Since J_{sync} could possibly have local minima, a grid is used to find the global minimum. Fig.11 shows J_{sync} as a function of T_d for a sample video with orientation estimates from the Integrator in Section IV-A1. There is a distinct minimum at $T_d = 0.01$.

VI. Experimental Results

In this Section, three experiments are performed to determine the feasibility of using sensor data, image data or combining both sensor and image data for the video rectification and stabilization. All these experiments are evaluated with different datasets using Matlab.

In our first experiment, we evaluate the sensor based orientation estimators of Section IV-A to determine their orientation accuracy. In our second experiment, we justify our usage of a rotation-only camera model as defined in III-C. In our third experiment, we evaluate our video rectification and stabilization method of Section V with orientation estimates from different orientation estimators in Section IV. Here we distinguish three classes of orientation estimators; sensor based, image based and sensor and image based orientation estimators. Finally, we do a comparison on complexity and accuracy of our video rectification and stabilization methods based on these three classes. This comparison also contains a justification of the used rolling shutter camera model, together with an analyses on how well camera translations can be approximated with camera rotations.

For all evaluations, sensor data together with image data is captured using an iPad 3. The captured sensor data is clocked at maximum sample frequencies of approximately 60, 90 and 60 Hz for the gyroscope, accelerometer and magnetometer respectively. In addition to the raw sensor data, the iOS SDK provides bias-corrected sensor data for both the gyroscope and magnetometer. This bias-corrected data is logged simultaneously with the raw sensor data. The image

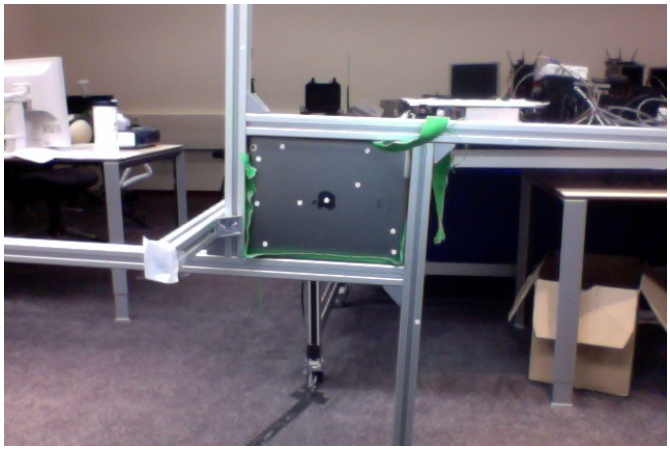


Fig. 12. 3D frame with iPad 3 attached for the experiments of Section VI-A and VI-B. Several infrared reflective markers are irregularly placed to track the orientation and position of the iPad 3.

data is captured at the resolution of 1280x720 pixels and logged together with the frame timestamps at 30 frames per second.

A. Sensor based orientation accuracy

Different sensor based orientation estimators of Section IV-A are evaluated to determine their orientation accuracy. The integration of gyroscope data in Section IV-A1 is referred to as Integrator [11]. The Kalman filter of Section IV-A2 is evaluated in two different configurations. First, configuration Kalman I, applied in Hanning's work [10], combining only gyroscope and accelerometer data. Second, configuration Kalman II, applied in Törnqvist's work [8], combining gyroscope, accelerometer and magnetometer data. Madgwick's filter [9] of Section IV-A3 is evaluated in a configuration combining gyroscope, accelerometer and magnetometer data.

In order to evaluate the accuracy of these orientation estimators, an experiment is conducted with a PS-Tech Optical Tracking System PST-55 [20]. This optical tracker uses infrared markers to track the iPad with two infrared cameras and can be sampled at 55 Hz. To maximize tracking accuracy the 3D frame of Fig.12 is attached to the iPad, however, the accuracy of the tracker's orientation estimates is given as less than 1°.

1) Experiment: In the experiment a single dataset is captured in which the iPad is rotated along all axes sequentially, while at the same time recording the ground truth orientation with the PS-Tech Optical Tracker. We synchronize the ground truth orientation data with the estimated orientation by cross-correlation.

In the experiment we use the bias-corrected gyroscope and magnetometer data provided by the iOS SDK; evaluation of both raw and bias-corrected sensor data for the estimation of orientation indicate a significant deterioration when using the raw sensor data. The accelerometer data is not bias-corrected by the iOS SDK. This is not needed; improvements of manually bias-correcting the accelerometer measurements, from data where the sensors are at rest, are negligible. Next to the bias-corrected sensor data provided by the iOS SDK, the iOS SDK also provides an orientation estimate relative to the earth frame for each time instance. This iOS orientation data together with the ground truth data is used to tune parameters of both Kalman filter configurations and Madgwick's filter.

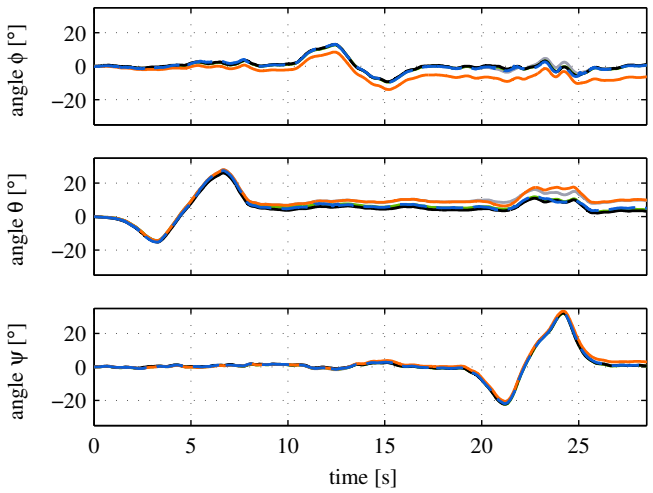


Fig. 13. Estimated Euler angles of sensor based orientation estimators. The estimated orientations of the Integrator [11] (—), Kalman I [10] (—), Kalman II [8] (—) and Madgwick [9] (—) are compared to the ground truth data (- - -) obtained by the PST-55 optical tracker.

TABLE I
INDIVIDUAL ANGULAR ERRORS ϕ_ϵ, θ_ϵ, ψ_ϵ AND GLOBAL ANGULAR ERROR δ FOR DIFFERENT ORIENTATION ESTIMATION METHODS.

Method	mean ϕ_ϵ	mean θ_ϵ	mean ψ_ϵ	mean δ
Integrator [11]	3.93°	2.76°	0.76°	4.98°
Kalman I [10]	0.48°	2.62°	0.14°	2.72°
Kalman II [8]	0.28°	0.43°	0.15°	0.62°
Madgwick [9]	0.40°	0.99°	0.21°	1.16°

Since both Kalman filter configurations and Madgwick's filter combine sensor data sampled at different frequencies, a mechanism to deal with this is required. In all discussed orientation estimators, the assumption holds that gyroscope measurements are constant during their sampling intervals. This assumption is extended to the magnetometer measurements. With this extended assumption, it is allowed to estimate the orientation with the highest sampling frequency of about 90 Hz, employed by the accelerometer.

2) Results: In order to better analyze the results, the quaternion output is converted to Euler angles ψ, ϕ and θ, using the ZXY sequence employed by the iOS SDK. With this representation, the orientation is considered a result of three composite rotations by the angles ψ, ϕ and θ, around the Z, X and Y axis respectively. Fig.13 displays the estimated Euler angles of the different orientation estimators under consideration together with the ground truth data obtained by the PS-Tech system. The Euler angles in Fig.13 are with respect to the initial orientation at time $t = 0$.

To determine which sensor based orientation estimator is able to keep a global accurate orientation estimate, we

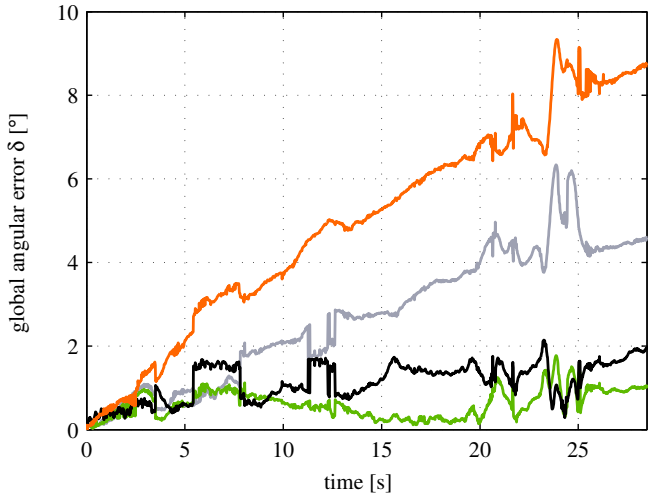


Fig. 14. Global angular error δ for the orientation estimated by the Integrator [11] (—), Kalman I [10] (—), Kalman II [8] (—) and Madgwick [9] (—) estimators. The errors are computed with respect to ground truth data obtained by the PST-55 optical tracker.

compare the estimated orientation with the ground truth data, by computing the errors in orientation. These errors, θ_ϵ, ψ_ϵ and ϕ_ϵ, are computed as the absolute difference between the Euler parameters of the ground truth data from the PS-Tech system and those of each of the corresponding estimated values. In Fig.14 the global angular error, δ, for each of the orientation estimators is presented. This global angular error, δ, is computed as the angle of the quaternion describing the rotation from the the ground truth measured orientation to the corresponding estimated orientation. The mean individual errors for each angle are summarized in Table I.

To maintain an accurate global orientation estimate over a long period of time, at least two different references of orientation in the earth frame are needed. This is clearly visible from the results, where the Integrator [11] has the largest global angular error due to integration of measurement errors. Furthermore, the Kalman I [10] configuration, combining only gyroscope and accelerometer data, is also not able to maintain an accurate global orientation. This is due to the lacking reference of magnetic north. Orientation estimators employing gyroscope, accelerometer and magnetometer measurements together, indicate that an accurate global estimate can be obtained with a maximum global angular error of about 2°. The effect of this error on the video rectification and stabilization is discussed in Section VI-C3.

B. Justification of rotation-only camera model

In Section III-C we choose to only model camera rotations. The justification of this decision depends on the type of application. Since the application domain is limited to situations where the user tries to keep the camera in a steady position (Section I), we use datasets in which this behavior is reflected.

1) Experiment: In the experiment datasets for *three sequences* in a single scene are captured in which we try to keep the camera in a steady position by focusing at a single point. While capturing these sequences, at the same time, we record the real camera rotations and translations with the PS-Tech Optical Tracking System. To determine the effect of camera rotations and translations, we first determine the maximum rotation angle and maximum translation for each of these sequences. Let the camera rotation and position at time instance t, relative to the reference orientation and position, be indicated with $\mathbf{q}_t$ and $\mathbf{p}_t$ respectively. The maximum rotation angle is then computed as the maximum angle of $\mathbf{q}_t$ over all time instances. The maximum camera translation is computed as the maximum distance of $\mathbf{p}_t$ to the reference position over all time instances.

TABLE II
MAXIMUM CAMERA ROTATION AND TRANSLATION IN SITUATIONS WHERE THE USER TRIES TO KEEP THE CAMERA IN A STEADY POSITION.

	Maximum rotation	**Maximum translation**
Sequence 1	1.37°	0.97 cm
Sequence 2	1.59°	1.00 cm
Sequence 3	0.99°	1.61 cm
Mean	1.32°	1.19 cm

2) Results: In Table II the maximum camera rotation and translation are listed for the three sequences captured in this experiment. Since the specifications of the PS-Tech optical tracking system are specified as <1 mm for the position and $< 1°$ for the orientation, it fair to state that within the used application domain, camera rotations are limited to 2° and translations are limited to 2 cm.

To evaluate the effect of a camera rotation or translation, similar to Fig.4 in Section III-C, a metric is needed. Because the result of a camera rotation or translation is reflected in the image plane, the average shift of the image, in terms of pixels, is a good indicator of how the image is changed due to a camera rotation or translation. To compute this, let the single image point $\mathbf{x}$ have corresponding 3D point $\mathbf{X}$. The new location of $\mathbf{x}$ undergoing a 3D camera rotation $\mathbf{R}$ and a 3D camera translation $\mathbf{t}$ is computed as:

$$\mathbf{x}' = \mathbf{KR}(\mathbf{X} + \mathbf{t}) \tag{47}$$

here $\mathbf{x}'$ is the new pixel location of image point $\mathbf{x}$. The average shift expressed in pixels can then be computed as:

$$J_{\text{shift}} = \frac{1}{N_c N_r} \sum_{c}^{N_c} \sum_{r}^{N_r} d(\mathbf{x}_{c,r}, \mathbf{x}'_{r,c}) \tag{48}$$

where N_c and N_r are the number of columns and rows in the image, image point $\mathbf{x}_{c,r}$ represents the image point at coordinate (c, r), and the distance function $d(\mathbf{x}, \mathbf{y})$ is defined in (19) as the Euclidian pixel distance.

Fig.15 and Fig.16 show the effect of camera rotations and translations as expressed in (48). Both rotations and translations are within the application domain where rotations are limited to the range of 0° to 2° and translations are limited to the range of 0 cm to 2 cm. In Fig.15 no camera translations are included, whereas in Fig.16 no camera rotations are included. In Section III-C we show that the distance to the scene only affects camera translations. Therefore, in Fig.16 scenes at different distances are evaluated. Note that both results

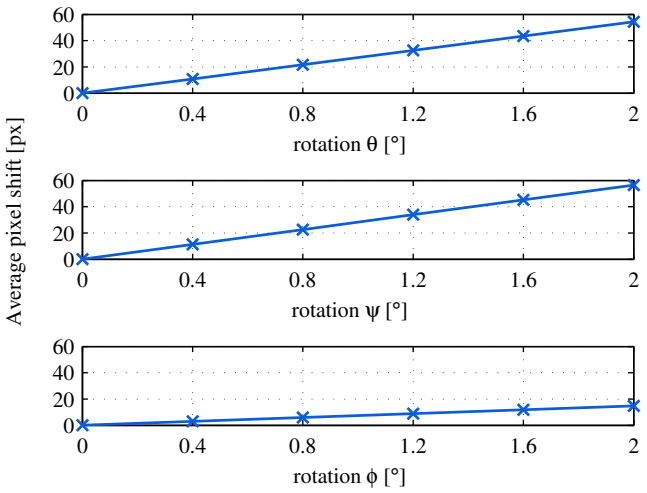


Fig. 15. Average pixel shift defined in (48) for different camera rotations within the application domain.

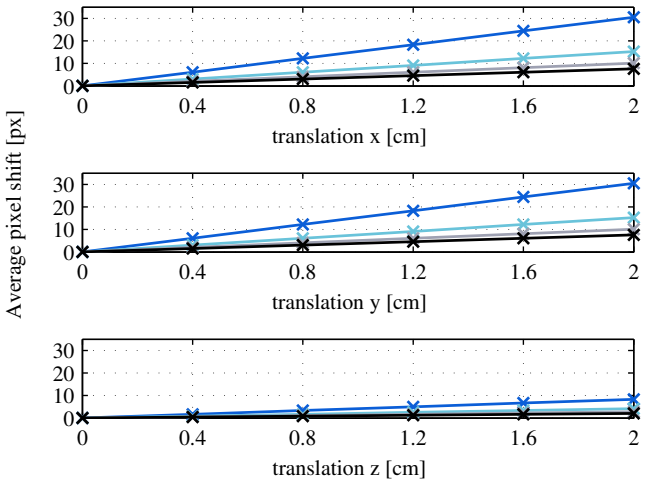


Fig. 16. Average pixel shift defined in (48) for different camera translations within the application domain. Because the effect of a camera translation is different for scenes at different distances, the effect of scenes at 1 m (—), 2 m (—), 3 m (—) and 4 m (—) are illustrated.

depend on the camera being used, since the camera matrix $\mathbf{K}$ is involved in all computations. However, using a different camera is expected to show similar results.

The results of Fig.15 and Fig.16 show that a camera rotation with an angle of 2° around the x-axis or y-axis result in approximately 60 pixels difference. On the other hand, the maximum pixel difference for a translation in the specified application domain is only 30 pixels. This is the pixel difference in the case of a 2 cm camera translation in the x-axis or y-axis with a scene distance of 1 m. Because the average scene distance is typically larger than 1 m in our application domain, the effect of translations are even lower. Combining this observation with the fact that camera translations are difficult to estimate, and that better rectification and stabilization results are obtained in the literature by only modeling rotations (Section III-C), modeling only camera rotations is justified. However, as camera translations are unavoidable, to theoretically remove all image distortions due to camera motion, modeling both camera rotations and translations is required.

C. Video rectification and stabilization accuracy

The video rectification and stabilization method described in Section V contains no tunable parameters. Therefore, the video rectification and stabilization quality depends solely on the quality of the orientation estimates. In this experiment all orientation estimators of Section IV are evaluated to show which method yields the best video rectification and stabilization.

The introduction of this Section describes how sensor and image data is acquired for the experiments. However, the image data still needs some additional processing to extract feature point correspondences before it can be used in any method employing image data. Since the quality of stabilization is significantly affected by the way feature point correspondences are found, the scope of this work is limited to Harris points [21] that tracked over the frames using the KLT tracker [22], [23]. The KLT tracker uses a spatial intensity gradient search that minimizes the Euclidian distance between the corresponding patches in the consecutive frames. To increase accuracy a crosschecking procedure is used [24]. When points are tracked from the first image to the other, the tracking is then reversed and only the points that return to the original position, within a threshold, are kept. This method doubles the computational cost, but removes outliers effectively. Next to the fact that this approach is commonly used in existing stabilization algorithms [10], [18], it also enables a fair comparison between relative and absolute methods; only feature points that are tracked throughout the entire sequence are preserved.

1) Metric: On the basis of the goal to align all images in the video to a single reference frame (Section I), a metric that measures the similarity of each frame compared to this reference frame is chosen. Several methods exist for measuring this similarity, for example mean squared error based methods like peak signal-to-noise ratio (PSNR) or structural similarity (SSIM). However, these methods are content dependent. In order to have a content independent metric, we compute how image points are reprojected into the reference frame, the so called reprojection error that is expressed in terms of pixels.

For simplicity, let the reference frame be the first frame in the image sequence. By having feature point correspondences $\mathbf{x}_{1,k}$ and $\mathbf{x}_{i,k}$, representing the k-th feature point in frame 1 and frame i, we can compute the average reprojection error between the reference frame and frame i as:

$$J_{\text{reproj}} = \frac{1}{K}\sum_{k}^{K} d(\mathbf{x}''_{1,k}, \mathbf{x}''_{i,k}) \tag{49}$$

where $\mathbf{x}''_{1,k}$ and $\mathbf{x}''_{i,k}$ represent the transformation of respectively $\mathbf{x}_{1,k}$ and $\mathbf{x}_{i,k}$, after rolling shutter correction and stabilization as defined in (40) and where the Euclidian distance function $d(\mathbf{x}, \mathbf{y})$ is defined in (19).

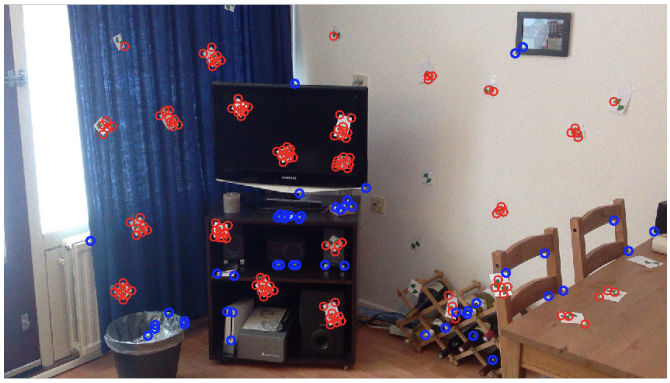


Fig. 17. Scene 3 used for benchmarking the video rectification and stabilization method. In this scene green markers are placed that are specifically used for the evaluation in (49). The features detected from these markers are indicated with red circles. The blue circles mark feature points used for the motion estimation.

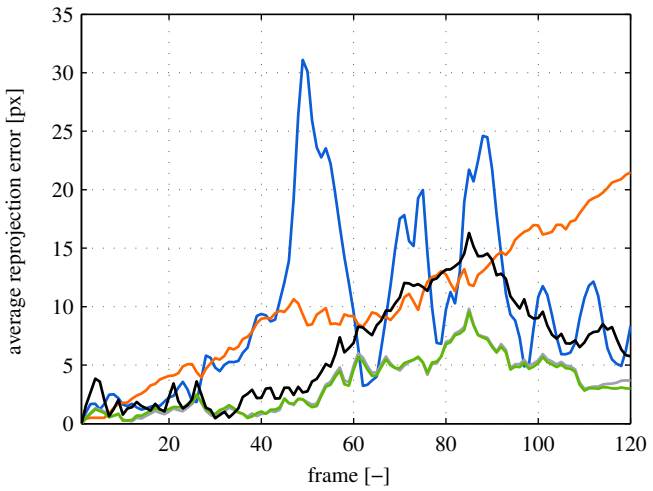


Fig. 18. Average reprojection error (49) for scene 3 in our benchmark set. The video rectification and stabilization is based on orientation estimates of respectively the Integrator [11] (—), Kalman I [10] (—), Kalman II [8] (—) and Madgwick's [9] (—) orientation estimator. The reprojection error of the original video is indicated with (—).

This method requires a set of K features being tracked throughout the complete sequence. It would not be fair to take the same set of features as used by motion estimation; their goal is to optimize for this distance used in (49). Fortunately, since we record our own benchmark data, additional markers that are tracked specifically for the evaluation can be "hidden" in the scene. Fig.17 shows such a scene with green markers attached. In this scene, the added markers are easily detected using Harris points together with the KLT-tracker. These markers are used in the computation of the reprojection error defined in (49). The other remaining features, not influenced by the added markers, are used for the orientation estimation. Note these markers can be added in *any scene*. Therefore, this metric enables a fair, content independent, numerical comparison of video rectification and stabilization method in the context of this work.

2) Experiment: In this experiment we capture three datasets, each for a different static *scene*. In these datasets we reflect the motions from the application domain by trying to keep the camera steady. The duration for each of the datasets is four seconds, corresponding to the maximum that can be stored in memory. To evaluate the video rectification and stabilization accuracy of the orientation estimators in Section IV, we use the average reprojection error defined in (49). However, this average reprojection is only given for a single scene. In order to make a fair comparison between different scenes, two important aspects are considered:

- **Drift:** In methods using only relative measurements, the accumulating measurement errors could result in a drift of the estimated camera orientation. This drift is naturally propagated in the average reprojection error in (49). In this experiment, we can approximate this drift by linear approximation[3], i.e. the average amount of pixels offset in each following frame. It is important to have a low drift for the image alignment, where a zero drift represents the ideal case.

[3]The linear approximation is computed by least-square fitting with the `polyfit` function in Matlab.

- **Stability:** Next to the drift of the resulting reprojection error in (49), the stability is also taken into account. Stability is expressed as the amount of variation in the video, the standard deviation σ of the average reprojection error without the approximated drift. Again, to obtain accurate image alignment, it is important to have a low variation in the sequence, where zero standard deviation represents the ideal case.

In Section IV three classes of orientation estimators are considered; sensor based, image based and both sensor and image based. In this experiment, each class of orientation estimators is evaluated individually.

3) Sensor based: The orientation accuracy found in the experiments of Section VI-A varied between each of the sensor based orientation estimators. To obtain accurate image alignment over a long period of time, an accurate global orientation estimate is required. This means that the Integrator [11] and Kalman I [10] orientation estimators are less suited for this application. The other two methods, the Kalman II [8] and Madgwick's [9] orientation estimator show that it is possible to maintain an accurate global orientation estimate within a maximum global error of 2°. This error in combination with the results of Section VI-B, where an orientation error of 2° translates to a reprojection error of 60 pixels, suggest that accurate image alignment is not possible using only sensor data.

Fig.18 shows the results for the video rectification and stabilization using sensor based orientation estimation methods for a single scene. The results expressed in terms of drift and stability for all three scenes are presented in Fig.19 and Table III. These results show that the sensor based rectified and stabilized videos, in almost all cases, improve over the original videos in terms of both drift and stability. The difference between the Kalman I [10] and Kalman II [8] configuration is not significant. This is probably due to the short time

TABLE III
THE AVERAGE REPROJECTION ERROR (49) EXPRESSED IN TERMS OF DRIFT, STANDARD DEVIATION AND THE MAXIMUM FOR THREE DIFFERENT SCENES. THE VIDEO RECTIFICATION AND STABILIZATION USES DIFFERENT SENSOR BASED ORIENTATION ESTIMATION METHODS.

Scene	Method	Drift	Stdev σ	Max
1	None	0.24 px	10.83 px	58.46 px
	Integrator [11]	0.21 px	1.16 px	25.66 px
	Kalman I [10]	0.05 px	1.15 px	11.39 px
	Kalman II [8]	0.04 px	1.12 px	9.23 px
	Madgwick [9]	0.04 px	1.49 px	10.92 px
2	None	0.22 px	4.07 px	32.74 px
	Integrator [11]	0.22 px	0.98 px	29.01 px
	Kalman I [10]	0.07 px	1.61 px	10.79 px
	Kalman II [8]	0.06 px	1.76 px	9.56 px
	Madgwick [9]	0.09 px	1.92 px	15.53 px
3	None	0.08 px	6.79 px	31.10 px
	Integrator [11]	0.16 px	1.27 px	22.06 px
	Kalman I [10]	0.05 px	1.39 px	9.47 px
	Kalman II [8]	0.04 px	1.41 px	9.24 px
	Madgwick [9]	0.09 px	2.99 px	16.16 px

interval of our captured video, where only 120 frames could be captured due to memory limitations. Furthermore, the results indicate that despite the larger drift of the Integrator [11] method, this method shows a relative good stability in terms of the standard deviation. The fact that the reprojection error is lower than 60 pixels, corresponding with an angular error of 2°, is due to both the relatively small motion, causing lower gyroscope scaling errors, and the short capture interval.

Additional experiments have been performed to determine the effect of different types of angular motions around the X, Y and Z-axis. In these experiments we expected that methods employing gyroscope, accelerometer and magnetometer data together, outperform the other sensor based orientation estimators depending on the type of motion. For example, we expected that an angular motion around the Y-axis, would have a lower drift in the methods that include the magnetometer. However, these differences were not visible in terms of our reprojection error in (49), probably caused by the limited time capture interval and the simple linear approximation of the drift. Therefore, these experiments did not contribute to extra insight.

To summarize, sensor data can be used to rectify and stabilize the video in order to better align the images compared to the original video. The accuracy is however limited to the orientation accuracy of the sensors. For higher accuracy incorporating image data is required.

4) Image based: The image based orientation estimation method in Section IV-B exists in two variations; the *relative* optimization method of Ringaby et al. [2], optimizing for feature point correspondences in consecutive frames, and our extension of Ringaby's method, the *absolute* method, optimizing for point correspondences with respect to a single reference frame. We use the Matlab function `lsqnonlin` for both methods to implement the non-linear least square optimization.

Both the relative and absolute optimization method allow two configuration parameters to be set; the length of the optimization interval and the number of knots in the optimization interval. The choice of both parameter settings depends heavily on the application domain. According Ringaby's work, the shorter the optimization interval, the better the rotation-only assumption holds. This is also confirmed in their experiments; their best results are obtained with optimization intervals of two or three frames. Therefore, we only consider optimization intervals of two and three frames for the relative optimization method. The minimum optimization interval for the absolute optimization method is one frame, due to the fact that orientations are found with respect to the reference frame, for which the orientation is fixed. Moreover, longer optimization intervals are not useful for the absolute optimization method; experiments with longer optimization intervals show similar rectification and stabilization results compared to an optimization interval of one frame. The explanation for this behavior is simple: while the absolute optimization finds the best camera orientations mapping the current frame to the reference frame, the relative method aims at finding the right camera orientation over time, i.e. not only for the individual frames.

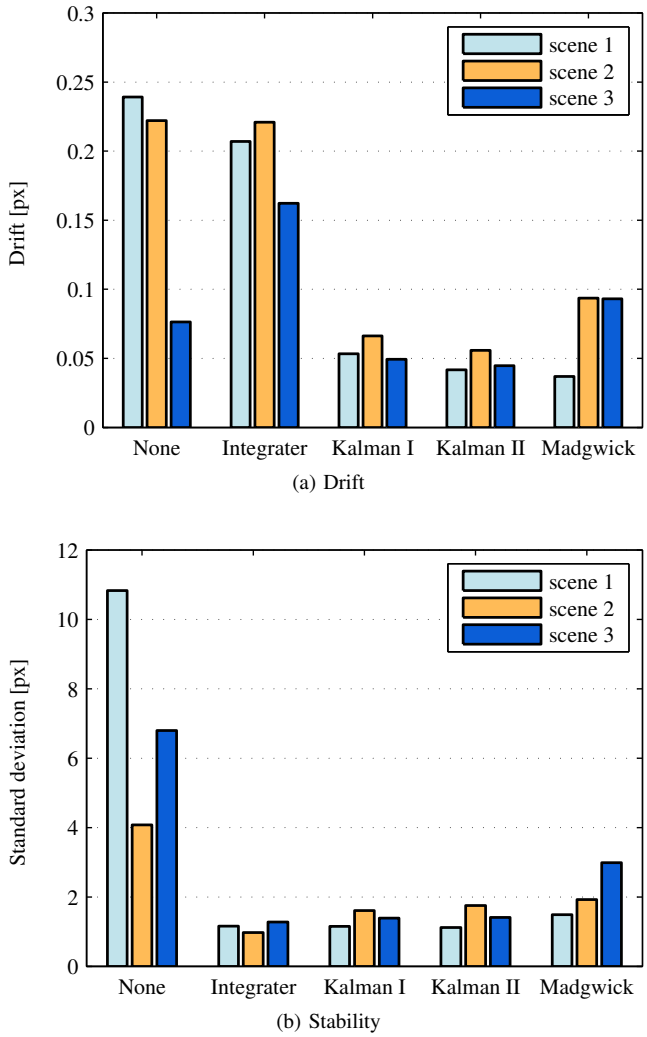


Fig. 19. Video rectification and stabilization results for three different scenes. Results are grouped per sensor based orientation estimation method. In (a) the drift of each method is depicted. In (b) the standard deviation is given as an indicator of the stability of the video.

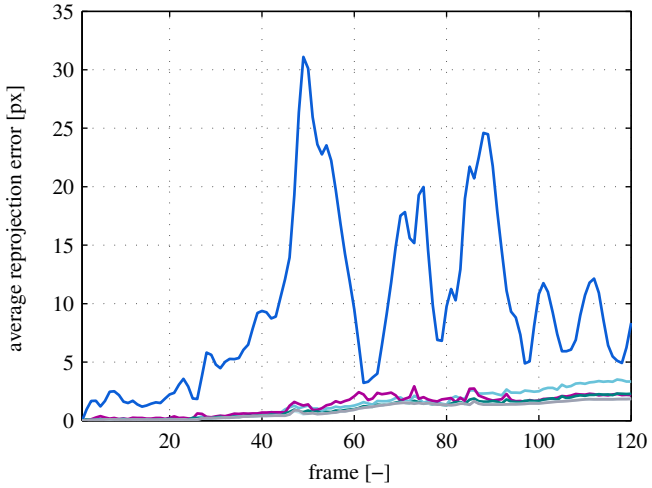


Fig. 20. Average reprojection error (49) for scene 3 in our benchmark set. The video rectification and stabilization uses image based orientation estimation methods. The relative [2] optimization method is evaluated with three configurations; three knots with two frames (—), four knots with two frames (—) and four knots with three frames (—). The absolute optimization method is evaluated using two configurations with overlapping lines; two knots with one frame (—) and three with knots one frame (—). The reprojection error of the original video is indicated with (—).

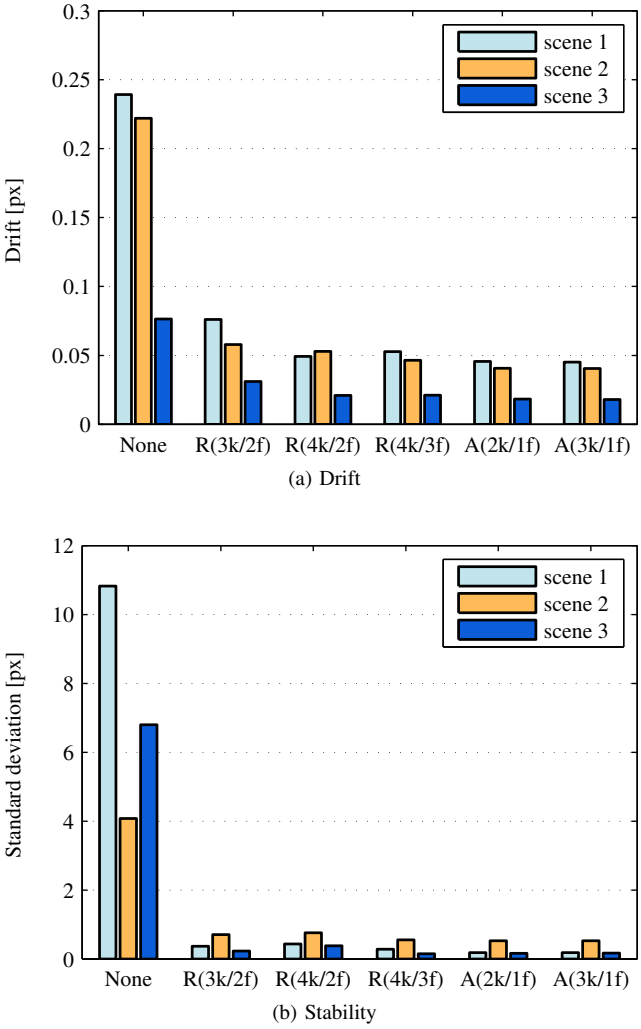


Fig. 21. Video rectification and stabilization results for three different scenes. Results are grouped per configuration of the relative (R) and the absolute (A) optimization method, where the configuration is given between brackets as (number of knots/number of frames). In (a) the absolute drift of each method is depicted. In (b) the standard deviation is given as an indicator of the stability of the video.

We refer back to our application domain, where motions originate from camera shake, for the choice on the number of knots. In our application domain camera shake is typical in the range of maximum 10 Hz. Therefore, only a limited number of knots is required to capture the camera's orientation. This minimum number of knots depends on the length of the optimization interval, e.g. a two frame optimization interval with three knots corresponds in our case to a 31 Hz frequency in which the camera's orientation can be captured [4]. The use of too many knots in the optimization interval would not only increase the computation time, the probability to have enough good points within the corresponding image region also declines.

Based on these choices for the optimization interval and the number of knots, different configurations are evaluated using our dataset. Fig.20 shows the results for the video rectification and stabilization for a single scene. Here, the same scene is used as in Fig.18, in which the video rectification and stabilization is done using sensor based orientation estimation. When both results are compared, we notice a significant difference; both image based methods produce a more stable result with a lower drift and stability. The results expressed in terms of drift and stability for all three scenes are presented in Fig.21 and Table IV. These results indicate the same; matched feature points provide a higher accuracy for the image alignment, compared to the sensor data. This comparison also shows that the drift is not always improved due to the simple linear approximation. In that case, the maximum value of the average reprojection error can also be taken into account.

The comparison between both image based orientation estimation methods shows that the relative optimization method has a higher drift and maximum value compared to our absolute optimization method. This is in line with our expectations as described in Section IV-B. However, because the motion is only small, the difference is not significant. Furthermore, the results indicate that incorporating more knots than required for capturing the camera's orientation, does not improve our accuracy. This can be seen in two cases; first, two configurations of the relative optimization method, namely the configurations with three and four knots over two frames, and second, both configurations of our absolute optimization method. The drift that is still present in the absolute optimization method is probably due to the camera translations that are still present in our random motion. These translations cannot be compensated completely with a camera rotation. A discussion on how translations can be approximated with rotations is presented in Section VI-D4.

To summarize, compared to sensor based orientation estima-

[4]The obtained frequency depends on the camera readout time T_r and the idle time T_{id}.

TABLE IV
THE AVERAGE REPROJECTION ERROR (49) EXPRESSED IN TERMS OF DRIFT, STANDARD DEVIATION AND THE MAXIMUM FOR THREE DIFFERENT SCENES. THE VIDEO RECTIFICATION AND STABILIZATION USES DIFFERENT CONFIGURATIONS OF THE RELATIVE [2] AND OUR ABSOLUTE OPTIMIZATION METHOD, WHERE THE CONFIGURATION IS GIVEN BETWEEN BRACKETS AS (NUMBER OF KNOTS/NUMBER OF FRAMES).

Scene	Method	Drift	Stdev σ	Max
1	None	0.24 px	10.83 px	58.46 px
	Relative [2] (3k/2f)	0.08 px	0.37 px	8.23 px
	Relative [2] (4k/2f)	0.05 px	0.44 px	5.98 px
	Relative [2] (4k/3f)	0.05 px	0.28 px	5.81 px
	Absolute (2k/1f)	0.05 px	0.18 px	4.93 px
	Absolute (3k/1f)	0.05 px	0.18 px	4.82 px
2	None	0.22 px	4.07 px	32.74 px
	Relative [2] (3k/2f)	0.06 px	0.71 px	6.28 px
	Relative [2] (4k/2f)	0.05 px	0.76 px	6.20 px
	Relative [2] (4k/3f)	0.05 px	0.55 px	5.23 px
	Absolute (2k/1f)	0.04 px	0.53 px	4.59 px
	Absolute (3k/1f)	0.04 px	0.53 px	4.64 px
3	None	0.08 px	6.79 px	31.10 px
	Relative [2] (3k/2f)	0.03 px	0.23 px	3.50 px
	Relative [2] (4k/2f)	0.02 px	0.38 px	2.93 px
	Relative [2] (4k/3f)	0.02 px	0.15 px	2.30 px
	Absolute (2k/1f)	0.02 px	0.16 px	1.87 px
	Absolute (3k/1f)	0.02 px	0.17 px	1.85 px

tion methods, estimating the orientation based on image data results in a more accurate image alignment. Yet, this is at the cost of a higher complexity as discussed in Section VI-D1. In all our scenes, the maximum value of the average reprojection error with respect to the reference frame, is within 5 pixels for 120 consecutive frames using the absolute optimization method.

5) Sensor and image based: In Section IV-C the sensor and image based orientation estimation method of Jia et al. [3] is described. This method is implemented by Chao Jia and provided as a toolbox in Matlab [25]. In Section IV-C, we show that Jia's method contains a flaw when the gyroscope sampling frequency is too low, i.e. the gyroscope sampling period T_{gyro} is larger than the idle time T_{id}. This flaw is also present in their code, resulting in *some* cases where the relative rotation between feature point correspondences in consecutive frames is undefined in terms of the state (Fig.9 in Section IV-C). These undefined cases result in negative time intervals in the code provided by their Matlab toolbox. Since Jia does not handle the specific case of $T_{\text{gyro}} > T_{\text{id}}$, which is justified using their data set, we choose to replace the incorrectly computed time intervals by zero. However, as a result of this decision, the comparison may not be fair. By keeping this in mind, our improvement to Jia's method is able to compute the relative rotation between feature point correspondences in consecutive frames in terms of the state in *all* cases.

Besides the original method and our improvement, we can also choose which gyroscope data to use. In our sensor based orientation estimation we have chosen to use bias-corrected sensor data provided by the iOS SDK for the estimation. However, in this experiment both bias-corrected data and raw

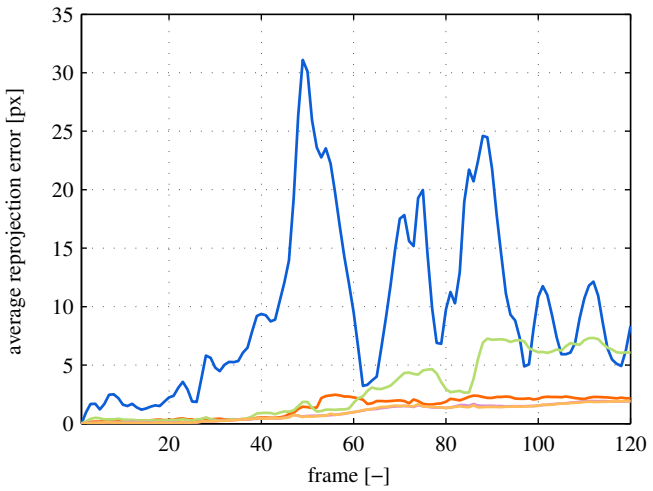


Fig. 22. Average reprojection error (49) for scene 3 in our benchmark set. The video rectification and stabilization uses orientation estimates from the sensor and image based method of Jia et al. [3]. The four configurations represent the original method, with (—) and without (—) bias-corrected gyroscope data, and the improved method, with (—) and without (—) bias-corrected gyroscope data. The reprojection error of the original video is indicated with (—).

TABLE V
THE AVERAGE REPROJECTION ERROR (49) EXPRESSED IN TERMS OF DRIFT, STANDARD DEVIATION AND THE MAXIMUM FOR THREE DIFFERENT SCENES. THE VIDEO RECTIFICATION AND STABILIZATION USES ORIENTATION ESTIMATES FROM THE SENSOR AND IMAGE BASED METHOD OF JIA ET AL. [3]. RESULTS ARE FOR THE ORIGINAL (O) AND THE IMPROVED (I) METHOD, WHERE RAW (R) OR BIAS-CORRECTED (C) GYROSCOPE DATA IS USED.

Scene	Method	Drift	Stdev σ	Max
1	None	0.24 px	10.83 px	58.46 px
	Jia [3] (r/o)	0.03 px	0.67 px	4.83 px
	Jia [3] (c/o)	0.07 px	0.81 px	8.24 px
	Jia [3] (r/i)	0.04 px	0.23 px	4.36 px
	Jia [3] (c/i)	0.04 px	0.19 px	4.52 px
2	None	0.22 px	4.07 px	32.74 px
	Jia [3] (r/o)	0.06 px	0.79 px	7.18 px
	Jia [3] (c/o)	0.04 px	0.57 px	5.11 px
	Jia [3] (r/i)	0.04 px	0.28 px	5.17 px
	Jia [3] (c/i)	0.04 px	0.57 px	4.99 px
3	None	0.08 px	6.79 px	31.10 px
	Jia [3] (r/o)	0.02 px	0.39 px	2.47 px
	Jia [3] (c/o)	0.07 px	1.00 px	7.33 px
	Jia [3] (r/i)	0.02 px	0.15 px	1.93 px
	Jia [3] (c/i)	0.02 px	0.16 px	1.93 px

sensor data are considered; we expect that Jia's method can also correct for gyroscope bias.

Fig.22 shows the results for the video rectification and stabilization using Jia's method for a single scene. Here, the same scene is used as in Fig.18 and Fig.20. Even though the comparison is unfair, the results indicate a significant deterioration for Jia's original method compared to our improvement to Jia's method. The results expressed in terms of drift and stability for all three scenes are presented in Table V and Fig.23. Again, these results indicate a significant

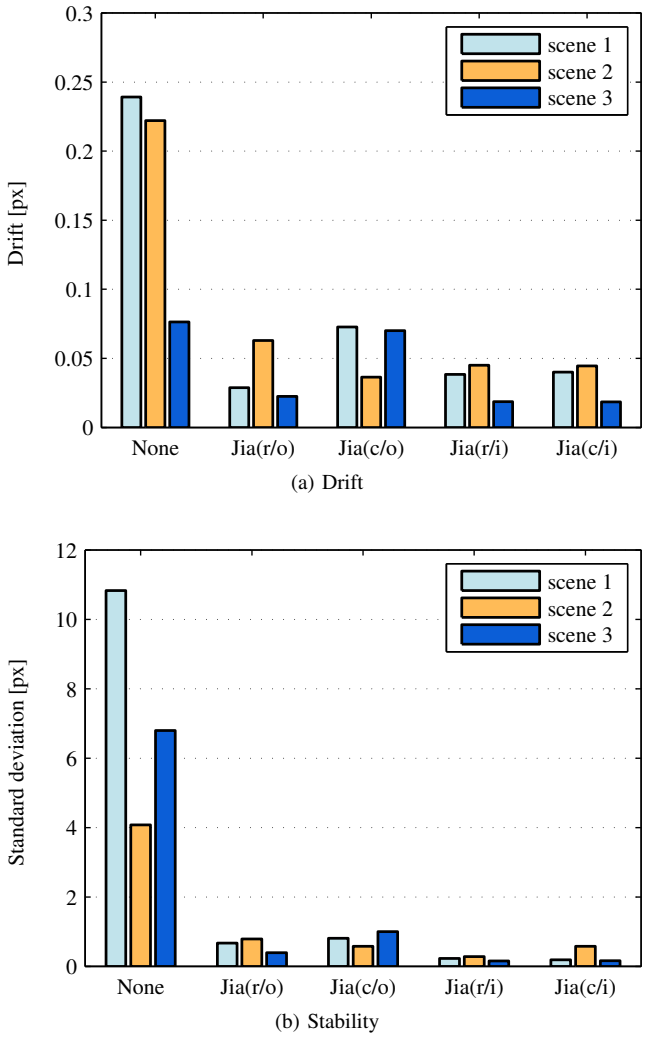


Fig. 23. Video rectification and stabilization results for three different scenes. The video rectification and stabilization uses orientation estimates from the sensor and image based method of Jia et al. [3]. Results are for the original (o) and the improved (i) method, where raw (r) or bias-corrected (c) gyroscope data is used. In (a) the absolute drift of each method is depicted. In (b) the standard deviation is given as an indicator of the stability of the video.

difference; our improvement to Jia's method shows that it is required to deal with the low gyroscope sampling frequency in the right way. Furthermore, both cases of our improvement to Jia's method, using raw or bias-corrected gyroscope data, give similar results. This means that Jia's method effectively corrects for gyroscope bias.

The comparison to our extension of the method of Ringaby et al. [2], the absolute optimization method, shows that similar results are obtained using Jia's method; the difference in terms of drift and stability compared to our improvement to Jia's method, using raw or bias-corrected gyroscope data, is negligible. However, Jia's method only makes use of relative measurements that inherently drift over time. Therefore, it is expected that the absolute optimization method works better over a longer period.

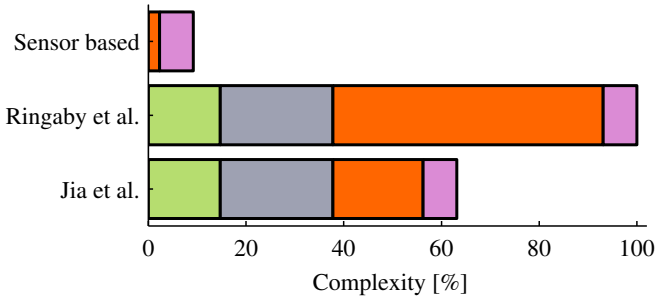


Fig. 24. Overview of algorithm complexity. The width of each box roughly corresponds to the fraction of time the algorithm spends during one frame. Different steps in the algorithm are feature point detection (), tracking (), orientation estimation () and rectification and stabilization ().

D. Comparison and discussion

In Section VI-C, all orientation estimators of Section IV are evaluated for the video rectification and stabilization accuracy using the same benchmark set consisting of three different scenes. These orientation estimators originated from three different classes; sensor based, image based and both sensor and image based. In this Section, the goal is to make an inter-class comparison. This requires that, next to the video rectification and stabilization accuracy, we also have to take into account the algorithm complexity. Therefore, we first determine the algorithm complexity. Then, after we do the comparison based on both complexity and video rectification and stabilization accuracy, we justify the used rolling shutter model of Section III-D. Finally, we conclude with an analyses on how well camera translations can be approximated with camera rotations.

1) Algorithm complexity: The algorithm complexity differs per class of orientation estimation. In our comparison we are interested in the relative differences in complexity between each class of orientation estimation. In work of Ringaby et al. [2], percentages are given for different steps in their algorithm. We use these percentages as a reference when determining the algorithm complexity for different methods. The video rectification and stabilization consists of four relevant steps: feature point detection, tracking, orientation estimation and rectification and stabilization.

The steps *feature point detection* and *tracking* are not needed when using the video rectification and stabilization method with sensor based orientation estimates; orientation estimates only originate from gyroscope, accelerometer and magnetometer measurements. The only step that differs among each class is the orientation estimation step. This step in Ringaby's method, referred to in this work as the *relative* optimization method, has a time complexity of $\mathcal{O}(N \times I \times M \times F)$ per frame. Here N is the number of feature point correspondences, I is the number of iterations in the non-linear least square optimization, M is the number of knots within the optimization interval and F is the number of frames within the optimization interval. This time complexity is similar to our extension of Ringaby's method, the *absolute* optimization method. Therefore, the complexity of Ringaby's method, is

a good indicator of our discussed methods in the class of image based orientation estimation. The time complexity for all discussed sensor based orientation estimators is $\mathcal{O}(S)$, where S is the number of sensor measurements within a frame period. In the class of both sensor and image based orientation estimators, we only consider the method of Jia et al. [3]. This method has a time complexity per frame of $\mathcal{O}(N \times S)$, where N is the number of feature point correspondences and S is the number of gyroscope measurements within a frame period.

In order to compare these different time complexities, we have to make assumptions on the quantities of F, S and M. Therefore, we only consider optimization intervals consisting of two frames ($F = 2$) for Ringaby's method with the assumption that the number of knots within this optimization interval is approximately the number of sensor measurements ($S \approx M$)[5]. In that case, compared to Ringaby's method, the time complexity differs by a factor $\mathcal{O}(N \times I)$ for sensor based orientation estimation methods and by a factor $\mathcal{O}(I)$ for Jia's method. These factors are used to compute the algorithm complexity as presented in Fig.24, where we assume a number of 3 iterations[6] for Ringaby's method in the orientation estimation step.

The algorithm complexity in Fig.24 shows that there is a significant difference between orientation estimation methods in different classes. The algorithm complexity of the sensor based methods is the lowest, here the overall computational cost with respect to Ringaby's method, is reduced by 90 percent. This reduction is due to two reasons; no feature points need to be detected and tracked for the orientation estimation, and the complexity of the orientation estimation step is considerably lower. The difference between the image based methods, represented by Ringaby's method, and the sensor and image based method, represented by Jia's method, is smaller. In this comparison, the overall computational cost reduces by approximately 35 percent, caused by the number of iterations to find the camera orientation. While the image based methods require multiple iterations over an optimization interval, Jia's method only requires one Kalman correction step. The fact that only one Kalman correction step is required in Jia's method, is due to the incorporation of gyroscope measurements. These measurements provide a good initial guess for the change in orientation of the camera.

2) Comparison: In this work, for each class of orientation estimation, different approaches or configurations are considered to evaluate the video rectification and stabilization accuracy. However, since most methods within a class share the same characteristics, we limit our comparison by choosing only one method or configuration representing each class. In the class of sensor based orientation estimators, we consider the Kalman II [8] estimator. This sensor based orientation estimator is able to keep an accurate global estimate as discussed in Section VI-A. The image based method we choose for the comparison is our extension of the method of Ringaby et al. [2], the *absolute* optimization method. This method does not drift due to errors in the orientation estimates in contrast to Ringaby's original method. Finally, in the class of both sensor and image based orientation estimation, we choose our improved method of Jia et al. [3] using raw gyroscope data.

[5]A gyroscope frequency of 60 Hz, corresponds approximately with 5 knots placed over a two frame optimization interval.

[6]The average number of iterations in our experiments to find the camera orientations using Ringaby's method.

TABLE VI
THE AVERAGE REPROJECTION ERROR (49) EXPRESSED IN TERMS OF DRIFT, STANDARD DEVIATION AND THE MAXIMUM FOR THREE DIFFERENT SCENES. THE VIDEO RECTIFICATION AND STABILIZATION USES ORIENTATION ESTIMATES OF RESPECTIVELY THE KALMAN II [8] ESTIMATOR, OUR EXTENSION OF THE METHOD OF RINGABY ET AL. [2], REFERRED TO AS THE ABSOLUTE OPTIMIZATION METHOD, AND THE IMPROVED METHOD OF JIA ET AL.[3].

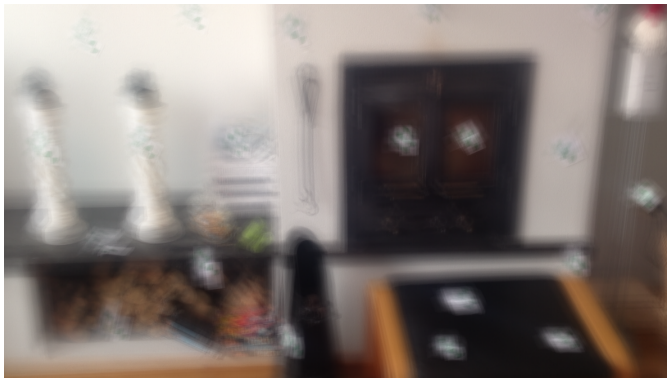
Scene	Method	Drift	Stdev σ	Max
1	None	0.24 px	10.83 px	58.46 px
	Kalman II [8]	0.04 px	1.14 px	9.25 px
	Absolute (2k/1f)	0.05 px	0.18 px	4.93 px
	Jia [3] (r/i)	0.04 px	0.23 px	4.36 px
2	None	0.22 px	4.07 px	32.74 px
	Kalman II [8]	0.06 px	1.76 px	9.56 px
	Absolute (2k/1f)	0.04 px	0.53 px	4.59 px
	Jia [3] (r/i)	0.04 px	0.28 px	5.17 px
3	None	0.08 px	6.79 px	31.10 px
	Kalman II [8]	0.05 px	1.58 px	9.56 px
	Absolute (2k/1f)	0.02 px	0.16 px	1.87 px
	Jia [3] (r/i)	0.02 px	0.15 px	1.93 px

The video rectification and stabilization results for these different methods are summarized in Table VI. In terms of our linear approximated drift, all methods show good results. In terms of stability, however, the video rectification and stabilization shows significant improvements when orientation estimates from the absolute optimization method or from Jia's method are used. With both methods, the standard deviation improves in almost all cases by a factor 4 over the Kalman II [8] estimator. To see the results visually, the temporal average is computed in Fig.25 for the original image sequence and the rectified and stabilized image sequences[7] for the methods in our comparison. The temporal average image for Jia's method is not depicted. This temporal average image is visually similar to the temporal average image of the absolute optimization method.

Fig.25 confirms the results in Table VI. By taking into account both complexity and accuracy, the following conclusions can be made for the video rectification and stabilization in our application domain.

- On platforms with low processing power and applications that do not require high rectification and stabilization accuracy, using sensor based orientation estimates for the rectification and stabilization improves the image alignment over the original image sequence. The algorithm complexity compared to any orientation estimation method that uses image data is significantly lower; no feature points need to be detected and tracked over the images. In our complexity analyses, we show that compared to image based orientation estimation, the overall computational cost can be reduced by up to 90 percent.

[7]The rectified and stabilized images are computed by forward interpolation with the `griddata` function in Matlab.



(a) Original

(b) Kalman II [8]

(c) Absolute (2 knots/1 frame)

Fig. 25. The temporal average images for scene 2 in our benchmark set. All images are cropped to hide black borders introduced by the rectification and stabilization method. In (a) the temporal average image of the original image sequence is given. In (b) and (c) the temporal average images of rectification and stabilization based on respectively the Kalman II [8] sensor based orientation estimator and our extension of the method of Ringaby et al. [2], the absolute optimization method, are given.

- Our extension of Ringaby's method, the absolute optimization method, offers similar accuracy as Jia's method for recordings within a limited time interval. In terms of complexity, however, there is a difference; while the absolute optimization method requires multiple iterations per frame to find an optimal camera orientation, the EKF-based method of Jia et al. only requires one Kalman correction step. Our complexity analyses shows that this can reduce the overall computational by up to 35 percent.

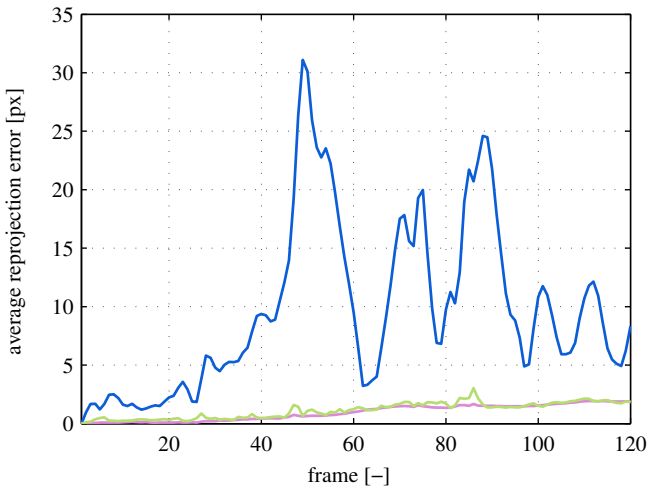


Fig. 26. Average reprojection error (49) for scene 3 in our benchmark set. The video rectification and stabilization is done using the sensor and image based orientation estimation method of Jia et al. [3]. The two configurations represent the video rectification and stabilization based on a rolling shutter camera model (—) and a global shutter camera model (—). The reprojection error of the original video is indicated with (—).

TABLE VII
THE AVERAGE REPROJECTION ERROR (49) EXPRESSED IN TERMS OF DRIFT, STANDARD DEVIATION AND THE MAXIMUM FOR THREE DIFFERENT SCENES. THE VIDEO RECTIFICATION AND STABILIZATION IS DONE USING THE SENSOR AND IMAGE BASED ORIENTATION ESTIMATION METHOD OF JIA ET AL. [3]. RESULTS ARE FOR THE ROLLING SHUTTER CAMERA MODEL (RS) AND THE GLOBAL SHUTTER CAMERA MODEL (GS).

Scene	Method	Drift	Stdev σ	Max
1	None	0.24 px	10.83 px	58.46 px
	Jia [3] (r/i) RS	0.04 px	0.23 px	4.36 px
	Jia [3] (r/i) GS	0.04 px	0.39 px	4.58 px
2	None	0.22 px	4.07 px	32.74 px
	Jia [3] (r/i) RS	0.04 px	0.28 px	5.17 px
	Jia [3] (r/i) GS	0.04 px	0.35 px	5.39 px
3	None	0.08 px	6.79 px	31.10 px
	Jia [3] (r/i) RS	0.02 px	0.15 px	1.93 px
	Jia [3] (r/i) GS	0.02 px	0.29 px	3.02 px

3) Justification of rolling shutter model: Throughout the complete report, we assume that rolling shutter artifacts are considered a major problem for which correction is required. To show that this is indeed the case in our application domain, we evaluate the video rectification and stabilization method for two camera models; the rolling shutter camera and the global shutter camera. We model the global shutter camera in our video rectification and stabilization method, by using a zero readout time T_r in our rolling shutter camera model of Section III-D. This way, all pixels in the image are read out at the same time instance, which is the case in a global shutter camera. Fig.26 shows the video rectification and stabilization results for a single scene using both camera models. The results expressed in terms of drift and stability for all three scenes are presented in Table VII. These results indicate a more unstable video when the global shutter camera model is used; the stability, which is expressed as the standard deviation,

is significantly deteriorated. Therefore, the use of the rolling shutter camera model as described in Section III-D is justified.

Note that in this work, we only have to correct for motions that originate from camera shake. In our application domain typical camera shake is in the range of maximum 10 Hz. Therefore, only a limited amount of orientation estimates per frame are required to capture the camera's motion. In this work we always satisfy this minimum amount of required orientation estimates per frame. However, in applications with motions that have different frequencies, it is required to take the minimum required orientation estimates per frame into account.

4) Rotations and translations in the image plane: In the experiments of Section VI-B we have derived the range of camera rotations and translations in our application domain using three recorded sequences. In these three sequences, camera rotations and translations are recorded with the PS-Tech optical tracking system. We can use this ground truth orientation data for image rectification and stabilization. It is not possible to decouple the effect of camera translations and rotations in terms of our metric in (49). Therefore, we assume that the average reprojection errors originate from camera translations. By also evaluating the image rectification and stabilization based on our extension of the method of Ringaby et al. [2], referred to as the absolute optimization method, we obtain an idea of how well camera translations can be approximated using rotations.

Under the assumption that both the ground truth data and the absolute optimization method do not contain drift, the results for the video rectification and stabilization are presented in Fig.27 for the three recorded sequences. These results are expressed in terms of the mean value and standard deviation of the average reprojection error in (49). Fig.27 shows that we can indeed partially approximate translations with rotations by incorporating image data. However, there are still camera translations that cannot be approximated with camera rotations only, as the results in Fig.27 indicate.

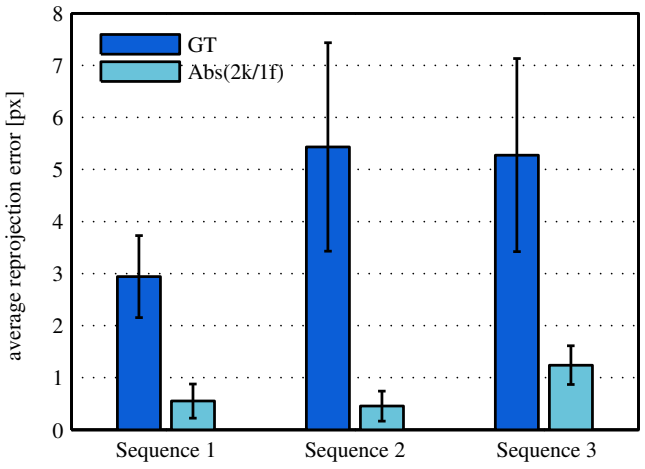


Fig. 27. Mean and standard deviation of the average reprojection error (49) for three sequences. The image rectification and video stabilization is done using the measured ground truth orientation data (GT) and our extension to the method of Ringaby et al. [2], referred to as the absolute optimization method (Abs(2k/1f)).

VII. Conclusion

In this work we considered the application domain of signal detection from video recorded by mobile devices. The scope of this work has been limited to situations where the user tries to keep the camera as steady as possible in order to capture a fixed static scene. This scene is typically located at a distance of more than 1 m from the camera. Two problems have been identified in this application domain, namely rolling shutter artifacts and shaky video due to the device's light weight. In order to correct for both problems, it is required to align all images in the video to a single reference frame.

We did an extensive evaluation on how to use sensor and image data to correct for rolling shutter artifacts and remove camera motion to obtain alignment of all images to a single reference frame. The four main contributions in this work are:

- We designed an evaluation method that measures the similarity of each frame with respect to a reference frame. This method is content independent, therefore enabling a fair numerical comparison of different video rectification and stabilization methods in our application domain.
- A recent and commonly referenced image based video rectification and stabilization method of Ringaby et al. [2] has been extended. This extended method produces better image alignment compared to the original method in our application domain.
- In the literature, only one method has been found that combines sensor and image data for the correction of rolling shutter artifacts and stabilization of the video. This is the method of Jia et al. [3], combining gyroscope measurements with image data. We improved this method, by identifying and correcting a flaw in some practical operating conditions.
- We evaluated a number of existing and improved video rectification and stabilization methods with our evaluation method using real sensor and image data.

These contributions together with their accompanied results, lead to the following conclusions in the context of our application domain:

- In the correction for camera motion, most motion induced distortions can be removed by rotations only. In addition, it is essential to incorporate the rolling shutter based image capture. Results without incorporating the rolling shutter based image capture, show lower video rectification and stabilization accuracy.
- Sensor based video rectification and stabilization produces better image alignment compared to the original image sequence. However, the overall accuracy is limited to the orientation accuracy of the sensors. The algorithm complexity of the sensor based video rectification and stabilization methods is relatively low; no feature point detection and tracking is needed to estimate the camera's orientation.
- Image based video rectification and stabilization offers high accuracy image alignment. However, compared to sensor based video rectification and stabilization, this

comes at the cost of significant higher algorithm complexity. We have shown that it is reasonable to expect that the overall computational cost increases by a factor of ten compared to sensor based video rectification and stabilization.

- The low complexity of sensor based video rectification and stabilization can be combined with the high accuracy of image based video rectification and stabilization. So far known, this has only been done in the method of Jia et al. [3]. The key is to use the sensor measurements as an initial prediction for the camera orientation, while using the image data to correct this prediction. We have shown that it is reasonable to expect that the overall computational cost can be reduced by up to 35 percent when incorporating the sensor data together with the image data for the video rectification and stabilization. This reduction is without sacrificing the high accuracy of image based rectification and stabilization.

VIII. Future work

In order to limit the scope of of this work, our application domain is limited to a specific range of applications. The applications we consider are limited to situations where a user keeps the camera in a steady position in order to capture a fixed static scene. However, future work can focus on situations where the scene contains motion of independent objects, i.e. the region of interest in the image plane moves during the capture interval.

Second, in this work we limited the extraction of feature point correspondences from the image sequence. In our work, we only considered Harris points [21] that are tracked over the frames by a KLT-tracker [22], [23]. However, a lot of research has been done in this field for image based stabilization methods. Our work can be extended by evaluating different methods of feature point detection and tracking.

Finally, during our research, Jia et al. made an improvement to their method incorporating gyroscope and image data [3]. In this proposed improvement [26], the correlation between feature point correspondences in consecutive frames, previously ignored, is taken into account. However, the same flaw that occurs in their original method is still present. Therefore, it would be interesting to extend their improved method with our proposed improvements.

Acknowledgment

I would like to thank my direct supervisors, ir. W.P. Lee and ir. F.J. de Bruijn, for their enthusiasm, useful advice and criticism. In addition, I would like to thank prof. dr. ir. G. de Haan for his advice and guidance throughout the project.

References

[1] A. El Gamal and H. Eltoukhy, "Cmos image sensors," *Circuits and Devices Magazine, IEEE*, vol. 21, no. 3, pp. 6–20, May/June 2005.

[2] E. Ringaby and P. Forssén, "Efficient video rectification and stabilisation for cell-phones," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 335–352, February 2012.

[3] C. Jia and B. Evans, "Probabilistic 3-d motion estimation for rolling shutter video rectification from visual and inertial measurements," in *Proc. IEEE Intl. Workshop on Multimedia Signal Processing*, Banff, Canada, September 2012, pp. 203–208.

[4] R. Faragher, "Understanding the basis of the kalman filter via a simple and intuitive derivation," *Signal Processing Magazine, IEEE*, vol. 29, no. 5, pp. 128–132, September 2012.

[5] H. Luinge, P. Veltink, and C. Baten, "Estimation of orientation with gyroscopes and accelerometers," in *Proc. First Joint BMES/EMBS Conference*, vol. 2, Atlanta, USA, October 1999, p. 844.

[6] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter," in *Proc. Virtual Reality Annual International Symposium*, Santa Clara, USA, March/April 1996, pp. 185–194.

[7] J. Marins, X. Yun, E. Bachmann, R. McGhee, and M. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, Maui, USA, October/November 2001, pp. 2003–2011.

[8] D. Törnqvist, "Estimation and detection with applications to navigation," Ph.D. dissertation, Linköping University, Department of Electrical Engineering, Automatic Control, November 2008.

[9] S. Madgwick, "An efficient orientation filter for inertial and inertial/-magnetic sensor arrays," University of Bristol (UK), Department of Mechanical Engineering, Tech. Rep., April 2010.

[10] G. Hanning, N. Forslow, P. Forssen, E. Ringaby, D. Tornqvist, and J. Callmer, "Stabilizing cell phone video using inertial measurement sensors," in *Proc. IEEE International Conference on Computer Vision Workshops*, Barcelona, Spain, November 2011, pp. 1–8.

[11] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes," Stanford University, Computer Graphics Laboratory, Tech. Rep., March 2011.

[12] C.-K. Liang, L.-W. Chang, and H. Chen, "Analysis and compensation of rolling shutter effect," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1323–1330, August 2008.

[13] J.-B. Chun, H. Jung, and C.-M. Kyung, "Suppressing rolling-shutter distortion of cmos image sensors by motion vector detection," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1479–1487, November 2008.

[14] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," in *Proc. ACM Transactions on Graphics*, vol. 28, no. 3, New Orleans, USA, August 2009, p. 44.

[15] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Transactions on Graphics*, vol. 30, no. 1, p. 4, January 2011.

[16] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, November 2000.

[17] J. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/, 2003, [Online; accessed 1 March 2013].

[18] P. Forssen and E. Ringaby, "Rectifying rolling shutter video from hand-held devices," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, June 2010, pp. 507–514.

[19] K. Shoemake, "Animating rotation with quaternion curves," *ACM SIGGRAPH computer graphics*, vol. 19, no. 3, pp. 245–254, 1985.

[20] PS-Tech, "Optical tracking system pst-55," http://ps-tech.com/tracking/pst-55/, [Online; accessed 1 March 2013].

[21] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of Fourth Alvey Vision Conference*, vol. 15, Manchester, UK, August/September 1988, p. 50.

[22] B. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proc. of the 7th international joint conference on Artificial intelligence*, vol. 2, Vancouver, Canada, April 1981, pp. 674–679.

[23] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 1994, pp. 593–600.

[24] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Proc. of the IEEE International Conference on Computer Vision*, Rio de Janeiro, Brasil, Oktober 2007.

[25] C. Jia, "Rolling Shutter Video Rectification Toolbox for MATLAB," http://users.ece.utexas.edu/~bevans/projects/dsc/software/rollingShutter/, 2012, [Online; accessed 29 January 2013].

[26] C. Jia and B. L. Evans, "Camera tracking and video rectification in rolling shutter cameras using visual and inertial measurements," 2012, [Submitted 15 November 2012 to IEEE Transactions on Image Processing].

[27] J. Kuipers, *Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 1999.

APPENDIX A
QUATERNION BASICS

A quaternion is a four-dimensional complex number that can be used to represent the orientation of a coordinate frame in three-dimension space. The quaternion representation of orientation is numerically more robust than, for example, Euler angles or rotation matrices. An orientation is a rotation with respect to a reference frame. Any orientation can be represented with an angle of rotation μ around a fixed axis $\mathbf{n}$ of unit length. Quaternions give a simple way to encode this axis-angle representation into four numbers, the unit quaternion:

$$\mathbf{q} = \begin{pmatrix} \cos(\mu/2) \\ \sin(\mu/2)\mathbf{n} \end{pmatrix} \tag{50}$$

In work of Kuipers et al. [27] a comprehensive description of unit quaternions is given, here only important aspects in the context of this work are given. An important operation is the quaternion inverse, denoted by $^{-1}$. When a quaternion $\mathbf{q}$ describes the rotation from frame B to frame A, the quaternion inverse $\mathbf{q}^{-1}$, describing the rotation from frame A to frame B, is defined as:

$$\mathbf{q}^{-1} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{pmatrix} \Big/ \operatorname{norm}(\mathbf{q}) \tag{51}$$

where $\operatorname{norm}(\mathbf{q})$ is defined as:

$$\mathbf{q} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = q_0^2 + q_1^2 + q_2^2 + q_3^2 \tag{52}$$

The quaternion product, denoted by $*$, can be used to compound rotations. For example, two quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$ can be compounded into $\mathbf{q}_3$, as if $\mathbf{q}_3$ is the rotation of $\mathbf{q}_1$ followed by $\mathbf{q}_2$. This compounded rotation of $\mathbf{q}_3$ is defined as:

$$\mathbf{q}_3 = \mathbf{q}_2 * \mathbf{q}_1 \tag{53}$$

where $\mathbf{a} * \mathbf{b}$ is defined as:

$$\mathbf{a} * \mathbf{b} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} * \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4 \\ a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3 \\ a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2 \\ a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1 \end{pmatrix} \tag{54}$$

A three dimensional vector $\mathbf{v}$ can be rotated by a quaternion $\mathbf{q}$ using the following relation:

$$\begin{pmatrix} 0 \\ \mathbf{v}' \end{pmatrix} = \mathbf{q} * \begin{pmatrix} 0 \\ \mathbf{v} \end{pmatrix} * \mathbf{q}^{-1} \tag{55}$$

here $\mathbf{v}'$ is the rotated vector.

Other operations used in this work concern conversion from and to other rotation representations. For example, the conversion from quaternion $\mathbf{q}$ to the rotation matrix $R(\mathbf{q})$:

$$R(\mathbf{q}) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ q(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{56}$$

Furthermore, we use the conversion from quaternion $\mathbf{q}$ to Euler angles ψ, ϕ and θ. This conversion is defined as:

$$\psi = \operatorname{arctan2}(2q_0q_3 - 2q_1q_2, q_0^2 - q_1^2 + q_2^2 - q_3^2) \tag{57}$$

$$\phi = \arcsin(2q_0q_1 + 2q_2q_3) \tag{58}$$

$$\theta = \operatorname{arctan2}(2q_0q_2 - 2q_1q_3, q_0^2 - q_1^2 - q_2^2 + q_3^2) \tag{59}$$

here, the rotation described by the quaternion $\mathbf{q}$ can now be considered a result of three composite rotations by the angles ψ, ϕ and θ around the Z, X and Y axis respectively.

APPENDIX B
KALMAN ORIENTATION FILTER

In this work we only consider camera rotations. Therefore, the state of the used Kalman filter is defined as the orientation in unit quaternion form[8] $\mathbf{q}_t = (q_{0,t}, q_{1,t}, q_{2,t}, q_{3,t})^T$. In order to reduce the dimension of the state space and to avoid nonlinearity, gyroscope measurement $\boldsymbol{\omega}_t$ is used as input to the motion model. This approach would only be correct if $\boldsymbol{\omega}_t$ could be measured without noise, but since the measurement noise from gyroscopes is small, $\boldsymbol{\omega}_t + \mathbf{v}_t$ is used as angular velocity measurements, where $\mathbf{v}_t$ is normally-distributed measurement noise. Under the assumption that the angular velocity is constant during sampling intervals of length T, and that $||\boldsymbol{\omega}_{t-1}||T$ is small, we can predict the state with the following discrete-time dynamic model [8] using small angle approximation:

$$\mathbf{q}_t = \Big(I + \frac{T}{2}\mathbf{A}(\boldsymbol{\omega}_{t-1})\Big)\mathbf{q}_{t-1} + \frac{T}{2}\mathbf{B}(\mathbf{q}_{t-1})\mathbf{v}_{t-1} \tag{60}$$

where

$$\mathbf{A}(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{pmatrix} \tag{61}$$

$$\mathbf{B}(\mathbf{q}) = \begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix} \tag{62}$$

Since the gyroscope measurements are already incorporated in the dynamic model, the measurement equations only consist of accelerometer and magnetometer measurements.

The accelerometer measures both free acceleration of the device and the earth's gravitational field into a single accelerometer measurement $\mathbf{a}_t$. However, when recording a video, the user usually tries to hold the camera in a steady

[8] In Section IV-A2 we refer to this state as $\mathbf{q}_{\text{kalman},t}$.

position. Therefore, the assumption is made that only the gravitational acceleration is affecting the accelerometer measurement $\mathbf{a}_t$. With this assumption, the measurement equation is defined as:

$$\mathbf{z}_{a,t} = \mathbf{a}_t + \mathbf{e}_{a,t} = R(\mathbf{q}_t)\mathbf{g}_e + \mathbf{e}_{a,t} = \mathbf{h}_a(\mathbf{q}_t) + \mathbf{e}_{a,t} \quad (63)$$

where $R(\mathbf{q}_t)$ is the rotation matrix corresponding to $\mathbf{q}_t$, $\mathbf{g}_e$ is the gravity vector in the earth frame and $\mathbf{e}_{a,t}$ is the measurement noise.

The normalized magnetometer measurement $\mathbf{m}_t$ points along the earth's magnetic field. Because the measurement is three dimensional, information is also given about the magnetic inclination. The inclination is the angle between the earth's tangent plane and the magnetic field vector and depends on the position on earth. The inclination is $0°$ at the magnetic equator and $90°$ at the magnetic poles, in Eindhoven the inclination is about $67°$. The measurement equation is written as:

$$\mathbf{z}_{m,t} = \mathbf{m}_t + \mathbf{e}_{m,t} = R(\mathbf{q}_t)\mathbf{m}_e + \mathbf{e}_{m,t} = \mathbf{h}_m(\mathbf{q}_t) + \mathbf{e}_{m,t} \quad (64)$$

where $\mathbf{m}_e$ is the vector pointing to earth magnetic north in the earth frame and $\mathbf{e}_{m,t}$ is the measurement noise. The earth frame including both the earth's gravity vector $\mathbf{g}_e$ and earth's magnetic north $\mathbf{m}_e$ is presented in Fig.6 in Section IV-A.

To do estimation using linear filter theory, the system must be linearized. The dynamic equation (60) is already in linear time-varying form, but both measurement equations (63) and (64) are nonlinear. Therefore, the measurement equations are linearized with a first order Taylor series. For the accelerometer part:

$$\begin{aligned}\mathbf{z}_{a,t} &= \mathbf{h}_a(\mathbf{q}_t) + \mathbf{e}_{a,t} \approx \\ &\mathbf{h}_a(\hat{\mathbf{q}}_{t|t-1}) + \mathbf{H}_a(\hat{\mathbf{q}}_{t|t-1})(\mathbf{q}_t - \hat{\mathbf{q}}_{t|t-1}) + \mathbf{e}_{a,t} \quad (65)\end{aligned}$$

where

$$\mathbf{H}_a(\hat{\mathbf{q}}_{t|t-1}) = \left.\frac{\partial \mathbf{h}_a(\mathbf{q})}{\partial \mathbf{q}}\right|_{\mathbf{q}=\hat{\mathbf{q}}_{t|t-1}} \quad (66)$$

The magnetometer equation is approximated as:

$$\begin{aligned}\mathbf{z}_{m,t} &= \mathbf{h}_m(\mathbf{q}_t) + \mathbf{e}_{m,t} \approx \\ &\mathbf{h}_m(\hat{\mathbf{q}}_{t|t-1}) + \mathbf{H}_m(\hat{\mathbf{q}}_{t|t-1})(\mathbf{q}_t - \hat{\mathbf{q}}_{t|t-1}) + \mathbf{e}_{m,t} \quad (67)\end{aligned}$$

where

$$\mathbf{H}_m(\hat{\mathbf{q}}_{t|t-1}) = \left.\frac{\partial \mathbf{h}_m(\mathbf{q})}{\partial \mathbf{q}}\right|_{\mathbf{q}=\hat{\mathbf{q}}_{t|t-1}} \quad (68)$$

To form the linearized measurement equations, a new measurement variable is formed by rearranging (65) and (67).

$$\begin{aligned}\tilde{\mathbf{z}_t} = \begin{pmatrix}\mathbf{z}_{a,t}\\ \mathbf{z}_{m,t}\end{pmatrix} - \begin{pmatrix}\mathbf{h}_a(\hat{\mathbf{q}}_{t|t-1})\\ \mathbf{h}_m(\hat{\mathbf{q}}_{t|t-1})\end{pmatrix} + \begin{pmatrix}\mathbf{H}_a(\hat{\mathbf{q}}_{t|t-1})\\ \mathbf{H}_m(\hat{\mathbf{q}}_{t|t-1})\end{pmatrix}\hat{\mathbf{q}}_{t|t-1} = \\ \begin{pmatrix}\mathbf{H}_a(\hat{\mathbf{q}}_{t|t-1})\\ \mathbf{H}_m(\hat{\mathbf{q}}_{t|t-1})\end{pmatrix}\mathbf{q}_t + \begin{pmatrix}\mathbf{e}_{a,t}\\ \mathbf{e}_{m,t}\end{pmatrix} \quad (69)\end{aligned}$$

Assume that the noise terms $\mathbf{v}_{t-1}$, $\mathbf{e}_{a,t}$ and $\mathbf{e}_{m,t}$ are normally-distributed random variables with covariance matrices $\mathbf{Q}_{t-1}$, $\mathbf{R}_{a,t}$ and $\mathbf{R}_{m,t}$, respectively. With these assumptions the EKF can be applied to (60), (63), (64) and (69) [8]. Note that the state $\mathbf{q}_t$ is normalized after both the prediction and correction step in order to maintain $\mathbf{q}_t$ as a unit quaternion.

APPENDIX C
RELATIVE OPTIMIZATION METHOD

In work of Ringaby et al. [2] two homogenous image points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, that correspond in consecutive frames i and $i+1$, are expressed as:

$$\mathbf{x}_i = \mathbf{KR}(t(\mathbf{x}_i))\mathbf{X}_i \text{ and } \mathbf{x}_{i+1} = \mathbf{KR}(t(\mathbf{x}_{i+1}))\mathbf{X}_{i+1} \quad (70)$$

where $t(\mathbf{x}_i)$ and $t(\mathbf{x}_{i+1})$ are time parameters for points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ respectively. This gives the relation:

$$\mathbf{x}_i = \mathbf{KR}\left(t(\mathbf{x}_i)\right)\mathbf{R}^T\left(t(\mathbf{x}_{i+1})\right)\mathbf{K}^{-1}\mathbf{x}_{i+1} \quad (71)$$

Each correspondence between two frames, similar to (71), result in two equations where the unknowns are the rotations. Since rotations are expressed using three parameters, this results in 6 unknowns. To restrict this number, the rotations parameterized with an interpolating linear spline with a number of so called knots placed over an optimization interval. Fig.28 shows an optimization interval of two frames and $M = 6$ knots. Within the optimization interval, intermediate rotations are found using spherical linear interpolation. Furthermore, because we need a reference world frame, the start of the first frame within the optimization interval is fixated to $\mathbf{R}_1 = \mathbf{I}$. As a result, an optimization interval consisting of M knots has $3(M-1)$ unknowns.

The cost function that is minimized using non-linear last square optimization is defined as:

$$J_{\text{rel}}(\mathbf{r}_1, \ldots, \mathbf{r}_M) \quad (72)$$

where $\mathbf{r}_1, \ldots, \mathbf{r}_M$ represent the rotations belonging to knot $1, \ldots, M$. The function J_{rel} represents the (symmetric) image-plane residuals of the set of K corresponding points $\mathbf{x}_{i,k} \leftrightarrow \mathbf{x}_{i+1,k}$ for frame i and $i+1$ respectively [9]:

$$J_{\text{rel}} = \sum_k^K d(\mathbf{x}_{i,k}, \mathbf{H}\mathbf{x}_{i+1,k})^2 + d(\mathbf{x}_{i+1,k}, \mathbf{H}^{-1}\mathbf{x}_{i,k})^2 \quad (73)$$

$$\mathbf{H} = \mathbf{KR}(t(\mathbf{x}_{i,k}))\mathbf{R}^T(t(\mathbf{x}_{i+1,k}))\mathbf{K}^{-1} \quad (74)$$

and where the Euclidian distance function $d(\mathbf{x}, \mathbf{y})$ for (homogeneous) vectors $\mathbf{x} = (x, y, z)^T$ and $\mathbf{y} = (u, v, w)^T$ is defined as:

$$d(\mathbf{x}, \mathbf{y})^2 = (x/z - u/w)^2 + (y/z - v/w)^2 \quad (75)$$

Each optimization interval consists of M knots, similar to Fig.28. The position of a knot is called the knot time, which is a time instance. Let the time instance of knot m, be indicated with N_m. As mentioned above, the intermediate rotations are found using spherical linear interpolation (SLERP)[2], where

[9] This function works for an optimization interval of two frames. However, by also incorporating feature correspondences between other consecutive frames, the extension to longer optimization intervals is trivial.

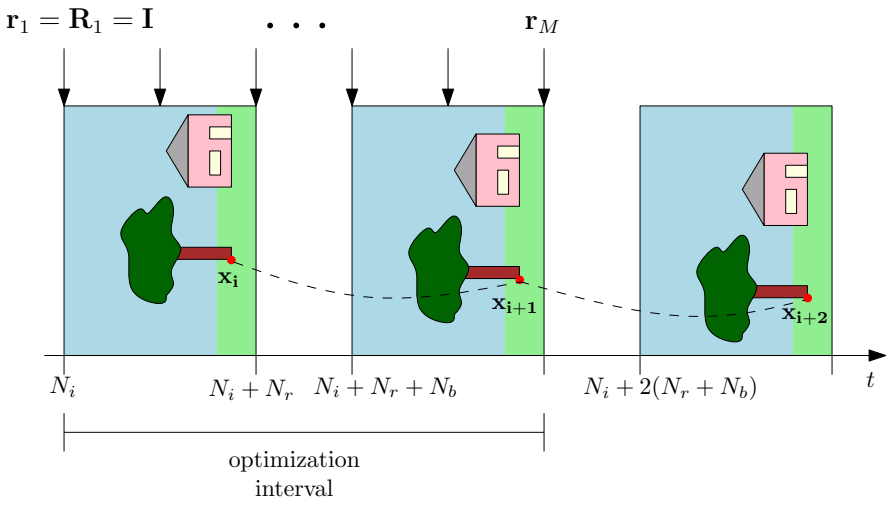


Fig. 28. Optimization interval for the relative optimization method consisting of two frames using $M = 6$ knots. Rotations $\mathbf{r}_1, \ldots, \mathbf{r}_M$ are found with non-linear least square optimization. The readout time is represented using the number of rows N_r and the idle time is represented using the number of blank rows N_b. An example feature correspondence, which is used for the optimization, is indicated with the dashed line between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ or between $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+2}$. After rotations $\mathbf{r}_1, \ldots, \mathbf{r}_M$ for the optimization interval starting from N_i are found, the optimization interval shifts to the start of frame $i + 1$, which is $N_i + N_r + N_b$.

the evaluation at a row with time parameter N_{curr} is denoted by:

$$\mathbf{R} = \text{SLERP}(\{\mathbf{r}_m, N_m\}_1^M, N_{\text{curr}}) \tag{76}$$

The orientations $\mathbf{r}_1, \ldots, \mathbf{r}_M$ which minimize (72) are found with non-linear least square optimization for a single optimization interval. To initialize a new optimization interval from the previous one, the following procedure is followed.

1) Change the origin of the camera to the next frame, by sampling:

$$\mathbf{R}_{\text{shift}} = \text{SLERP}(\{\mathbf{r}_m, N_m\}_1^M, N_r + N_b) \tag{77}$$

2) Shift the interval one step, by re-sampling the knots $\{\mathbf{r}_m\}_1^M$ with an offset of $N_r + N_b$:

$$\mathbf{R}_m = \text{SLERP}(\{\mathbf{r}_m, N_m\}_1^M, N_m + N_r + N_b) \tag{78}$$

3) Correct them for the change in origin for $m = 1, \ldots, L$:

$$\mathbf{r}_m^{'} = \text{logm}(\mathbf{R}_{\text{shift}}^T \mathbf{R}_m) \tag{79}$$

where N_L is the last time inside the optimization interval. Newly shifted-in rotations are copied from the last valid knot $\mathbf{r}_L^{'}$.