Jeroen D. Hol
Xsens Technologies B.V.
Enschede,
The Netherlands
jeroen.hol@xsens.com

Thomas B. Schön
Fredrik Gustafsson
Division of Automatic Control
Linköping University,
Sweden
{schon,fredrik}@isy.liu.se

Modeling and Calibration of Inertial and Vision Sensors

Abstract

This paper is concerned with the problem of estimating the relative translation and orientation of an inertial measurement unit and a camera, which are rigidly connected. The key is to realize that this problem is in fact an instance of a standard problem within the area of system identification, referred to as a gray-box problem. We propose a new algorithm for estimating the relative translation and orientation, which does not require any additional hardware, except a piece of paper with a checkerboard pattern on it. The method is based on a physical model which can also be used in solving, for example, sensor fusion problems. The experimental results show that the method works well in practice, both for perspective and spherical cameras.

KEY WORDS—vision sensors, inertial sensors, sensor fusion, calibration, gray-box system identification

1. Introduction

This paper is concerned with the problem of estimating the relative translation and orientation between a camera and an inertial measurement unit (IMU) that are rigidly connected. The algorithm is capable of handling both perspective and spherical cameras. Accurate knowledge of the relative translation and orientation is an important enabler for high-quality sensor fusion using measurements from both sensors.

The sensor unit used in this work is shown in Figure 1. For more information about this particular sensor unit, see Hol (2008)[1]. The combination of vision and inertial sensors is very suitable for a wide range of robotics applications and a solid introduction to the technology is provided by Corke et al. (2007). The high-dynamic motion measurements of the IMU are used to support the vision algorithms by providing accurate predictions where features can be expected in the upcoming frame. This facilitates development of robust real-time pose estimation and feature detection/association algorithms, which are the cornerstones in many applications, including for example augmented reality (AR) (Chandaria et al. 2007; Bleser and Stricker 2008) and simultaneous localization and mapping (SLAM) (Bailey and Durrant-Whyte 2006; Durrant-Whyte and Bailey 2006).

The main contribution of this work is a calibration algorithm which provides high-quality estimates of the relative translation and orientation between a camera and an IMU. The proposed calibration algorithm is fast and, more importantly, it is simple to use in practice. We also provide a quality measure for the estimates in terms of their covariance. The derived calibration algorithm requires accurate predictions of the sensor measurements. Hence, an additional contribution is a dynamic model and a measurement model for a combined camera and IMU sensor unit. This model can be used straightforwardly in solving, for example, sensor fusion problems. Early versions of the present work have previously been published in Hol et al. (2008a,b).

The International Journal of Robotics Research
Vol. 29, No. 2–3, February/March 2010, pp. 231–244
DOI: 10.1177/0278364909356812

Figures 1, 5 appear in color online: http://ijr.sagepub.com

1. See also http://www.xsens.com.

Fig. 1. The sensor unit, consisting of an IMU and a camera. In this photo a fisheye lens was attached to the camera. The camera calibration pattern is visible in the background.

Let us now very briefly introduce the approach used in order to solve the calibration problem at hand. In order to find the unknown translation and orientation we will of course need data from the sensor unit. It is sufficient to move the sensor unit over a checkerboard pattern (see Figure 1) for a couple of seconds and record the images and the inertial data during this motion. We will make use of a dynamic model of the motion

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}) + \boldsymbol{w}_t, \tag{1a}$$

$$\boldsymbol{y}_t = h(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{e}_t, \tag{1b}$$

where $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$ denotes the state, $\boldsymbol{u}_t \in \mathbb{R}^{n_u}$ denotes the input signals, $\boldsymbol{y}_t \in \mathbb{R}^{n_y}$ denotes the output signals, $\boldsymbol{\theta}$ denotes the unknown parameters, and $\boldsymbol{w}_t$ and $\boldsymbol{e}_t$ denote the noise processes. Finally, the functions $f(\cdot)$ and $h(\cdot)$ describe the dynamics and how the measurements are related to the states, respectively. Note that the model depends on the parameters $\boldsymbol{\theta}$ we are looking for, a fact which will be exploited. Based on model (1) and the measured input $\boldsymbol{u}_t$ and output $\boldsymbol{y}_t$ signals from the sensor unit (i.e. the inertial data and the images) we can use the extended Kalman filter (EKF) to compute a prediction of the measurement $\hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta})$. Finally, we can use the prediction to form an optimization problem and solve this for the parameters that best describe the measurements recorded during the experiment. This approach for identifying parameters in dynamic systems is a special case of gray-box system identification (Graebe 1990; Ljung 1999).

Current state-of-the-art when it comes to calibration of the relative translation and orientation between a camera and an IMU is provided by Lobo and Dias (2007) and Mirzaei and Roumeliotis (2008). The first paper presents a two-step algorithm. First, the relative orientation is determined and then the relative position is determined using a turntable. The drawbacks of this method are that it requires a turntable and that it is rather labor intensive. Both these drawbacks are eliminated by the algorithm proposed in Mirzaei and Roumeliotis (2008) and by the algorithm introduced in this work. The solution proposed by Mirzaei and Roumeliotis (2008) is fundamentally different to the solution proposed in the present work. Their approach is to transform the parameter estimation problem into a state estimation problem by augmenting the state vector $\boldsymbol{x}_t$ with the parameters $\boldsymbol{\theta}$ and then estimating the augmented vector using a Kalman filter. This is an often used approach for handling this class of problem, see, e.g., Ljung and Söderström (1983). Furthermore, the work of Foxlin and Naimark (2003) is worth mentioning, where a custom calibration rig is used together with a set of artificial landmarks. A closely related topic is that of hand-eye calibration, where the relative pose between a robot and a camera is determined, see, e.g., Tsai and Lenz (1989) and Daniilidis (1999).

In Section 2 we provide a thorough problem formulation, where the necessary coordinate frames are introduced and the background for the calibration algorithm is provided. The dynamic model and the measurement models are then introduced in Section 3. These models allow us to obtain the predictor that is needed. The calibration algorithms are introduced in Section 4. The practical experiments are reported in Section 5, together with the results both for perspective and spherical cameras. Finally, the conclusions are given in Section 6.

2. Problem Formulation

In this section we will give a more formal formulation of the problem we are solving. We start by introducing the coordinate frames that are used:

- **Navigation frame (n):** The camera pose is estimated with respect to this coordinate frame, sometimes denoted as the Earth or world frame. The 3D feature positions are, without loss of generality, assumed to be constant and known in this frame. It is fixed to the environment and can be aligned in any direction; however, preferably it should be vertically aligned.

- **Camera frame (c):** This coordinate frame is attached to the moving camera. Its origin is located in the optical center of the camera, with the z-axis pointing along the optical axis.

- **Image frame (i):** The 2D coordinate frame of the camera images. It is located on the image plane, which is perpendicular to the optical axis.

- **Body frame (b):** This is the coordinate frame of the IMU and it is rigidly connected to the c frame. All the inertial measurements are resolved in this coordinate frame.

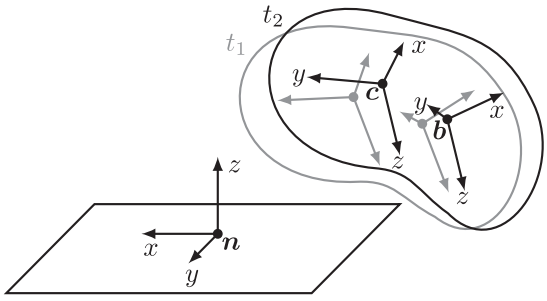


Fig. 2. The sensor unit, shown at two time instants, t_1 and t_2, consists of an IMU (b frame) and a camera (c frame). These frames are rigidly connected. The position of the sensor unit with respect to the navigation frame (n) changes over time as the unit is moved.

In the following sections, scalars are denoted with lowercase letters (u, ρ), vectors with bold letters ($\boldsymbol{b}$, $\boldsymbol{\theta}$), quaternions with bold letters with a ring on top ($\mathring{\boldsymbol{q}}$, $\mathring{\boldsymbol{e}}$), and matrices with boldface capitals ($\boldsymbol{A}$, $\boldsymbol{R}$). Coordinate frames are used to denote the frame in which a quantity is expressed as well as to denote the origin of the frame; for instance, $\boldsymbol{b}^n$ is the position of the body frame expressed in the navigation frame and $\boldsymbol{c}^b$ is the position of the camera frame expressed in the body frame. Furthermore, $\mathring{\boldsymbol{q}}^{bn}$, $\boldsymbol{\varphi}^{bn}$, $\boldsymbol{R}^{bn}$ are the unit quaternion, the rotation vector and the rotation matrix, respectively, all interchangeable and describing the rotation from the navigation frame to the body frame, see, e.g., Shuster (1993) and Kuipers (1999) for an overview of rotation parameterizations.

In Figure 2 the relationship between the coordinate frames is illustrated. The camera and the IMU are rigidly connected, i.e., $\boldsymbol{c}^b$ and $\boldsymbol{\varphi}^{cb}$ are constant. The position of the camera is in this setting defined as the position of its optical center. Although this is a theoretically well-defined quantity, its physical location is rather hard to pinpoint without exact knowledge of the design of the optical system and typically a calibration algorithm has to be used to locate it.

The goal of this work is to devise an algorithm that is capable of estimating the following parameters:

- the relative orientation of the body and the camera frames, parameterized using a rotation vector $\boldsymbol{\varphi}^{cb}$;
- the relative translation between these frames, parameterized as $\boldsymbol{c}^b$, i.e., the position of the camera frame expressed in the body frame.

In order to compute estimates of these parameters we need information about the system, provided by input and output data. This data is denoted as

$$\boldsymbol{Z} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M, \boldsymbol{y}_L, \boldsymbol{y}_{2L} \ldots, \boldsymbol{y}_{LN}\}, \tag{2}$$

where $\boldsymbol{u}_t$ denote the input signals and $\boldsymbol{y}_t$ denote the output signals (measurements). In the present work the data from the inertial sensors is modeled as input signals and the information from the camera is modeled as measurements. Note that the inertial sensors are typically sampled at a higher frequency than the camera, motivating the use of the multiplier L and the two different numbers of samples in (2), M and N, where $M = LN$.

The dataset does not have to satisfy any constraints other than that it should be informative, i.e., it should allow one to distinguish between different models and/or parameter vectors (Ljung 1999). It is very hard to quantify this notion of informativeness, but in an uninformative experiment the predicted output will not be sensitive to certain parameters of interest and this results in large variances of the obtained parameter estimates. For the calibration problem at hand, the presence of angular velocity is key. Its amplitude and duration, however, should match the intended application.

In order to be able to use the data in Equation (2) for our purposes we need a predictor, capable of predicting measurements. For the present problem, we can derive such a predictor based on a dynamic model in combination with an estimator; this is the subject of Section 3. More abstractly speaking, the predictor is a parameterized mapping $g(\cdot)$ from past input and output signals $\boldsymbol{Z}^{t-1}$ to the space of the model outputs,

$$\hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta}) = g(\boldsymbol{\theta}, \boldsymbol{Z}^{t-1}), \tag{3}$$

where $\boldsymbol{Z}^{t-1}$ is used to denote all the input and output signals up to time $t-1$. Here, $\boldsymbol{\theta}$ denotes all the parameters to be estimated, which of course include the relative translation $\boldsymbol{c}^b$ and orientation $\boldsymbol{\varphi}^{cb}$ of the camera and the IMU.

Finally, in order to compute an estimate of the parameters $\boldsymbol{\theta}$ we need a way of determining which parameters are best at describing the data. This is accomplished by posing and solving an optimization problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} V_N(\boldsymbol{\theta}, \boldsymbol{Z}). \tag{4}$$

Here, the cost function $V_N(\boldsymbol{\theta},\boldsymbol{Z})$ is of the form

$$V_N(\boldsymbol{\theta}, \boldsymbol{Z}) = \frac{1}{2}\sum_{t=1}^{N} \|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta})\|^2_{\boldsymbol{\Lambda}_t}, \tag{5}$$

where $\boldsymbol{\Lambda}_t$ is a suitable weighting matrix. The details regarding the formulation and solution of this optimization problem are given in Section 4. The problem of computing estimates of $\boldsymbol{\theta}$ based on the information in $\boldsymbol{Z}$ according to the above formulation is a so-called gray-box system identification problem, see, e.g., Graebe (1990) and Ljung (1999).

3. Modeling

The calibration method introduced in the previous section requires a predictor. We aim at providing a self-contained deriva-

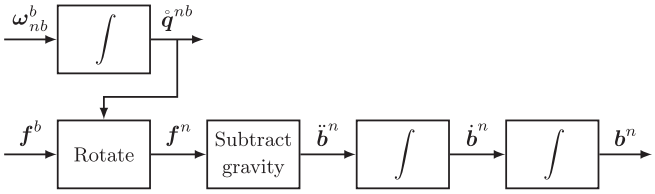


Fig. 3. Strapdown inertial navigation algorithm.

tion of such a predictor and start with a rather thorough analysis of inertial and vision sensors in Sections 3.1 and 3.2, respectively. Together with the dynamics discussed in Sections 3.3 these constitute a state-space description of the sensor unit which forms the basis of the predictor described in detail in Sections 3.4.

3.1. Inertial Sensors

An IMU consists of accelerometers and rate gyroscopes. The gyroscopes measure angular velocity or rate-of-turn $\boldsymbol{\omega}$. The accelerometers do not measure accelerations directly, but rather the external specific force $\boldsymbol{f}$. Both linear acceleration $\ddot{\boldsymbol{b}}$ and the Earth's gravitational field $\boldsymbol{g}$ contribute to the specific force.

The measurements from the accelerometers and gyroscopes can be used to compute the position and orientation of an object relative to a known starting point using inertial navigation (Chatfield 1997; Titterton and Weston 1997; Woodman 2007). In a strapdown configuration such as the sensor unit, the measurements are resolved in the body coordinate frame, rather than in an inertial reference frame. Hence, the orientation $\mathring{\boldsymbol{q}}^{nb}$ can be calculated by integrating the angular velocity $\boldsymbol{\omega}^b_{nb}$. The position $\boldsymbol{b}^n$ can be obtained by double integration of the external specific force $\boldsymbol{f}^b$, which has been rotated using the known orientation and corrected for gravity. This procedure is illustrated in Figure 3.

In practice, the angular velocity $\boldsymbol{\omega}^b_{nb}$ and the external specific force $\boldsymbol{f}^b$ are obtained from the gyroscope and the accelerometer measurements. These include bias and noise terms which cause errors in the calculated position and orientation. This integration drift is inherent to all inertial navigation. Moreover, using microelectromechanical systems (MEMS) inertial sensors, the integration drift is relatively large. Hence, the orientation estimate and especially the position estimate, are only accurate and reliable for a short period of time.

Summarizing the above discussion, the gyroscope measurements are modeled as

$$\boldsymbol{y}_\omega = \boldsymbol{\omega}^b_{nb} + \boldsymbol{\delta}^b_\omega + \boldsymbol{e}^b_\omega. \tag{6}$$

Here, $\boldsymbol{\omega}^b_{nb}$ is the angular velocity, body to navigation, expressed in the body coordinate frame, $\boldsymbol{\delta}^b_\omega$ is a slowly time-varying bias term and $\boldsymbol{e}^b_\omega$ is independent and identically distributed (i.i.d.) Gaussian noise. Furthermore, the accelerometer measurements are modeled as

$$\boldsymbol{y}_a = \boldsymbol{f}^b + \boldsymbol{\delta}^b_a + \boldsymbol{e}^b_a = \boldsymbol{R}^{bn}(\ddot{\boldsymbol{b}}^n - \boldsymbol{g}^n) + \boldsymbol{\delta}^b_a + \boldsymbol{e}^b_a, \tag{7}$$

where $\boldsymbol{f}^b$ is the external specific force expressed in the body coordinate frame, $\boldsymbol{\delta}^b_a$ is a slowly time-varying bias and $\boldsymbol{e}^b_a$ is i.i.d. Gaussian noise. The second expression in (7) splits the specific force into its contributions from the linear acceleration of the sensor $\ddot{\boldsymbol{b}}^n$ and the gravity vector $\boldsymbol{g}^n$, both expressed in the navigation coordinate frame. These vectors have been rotated to the body coordinate frame using the rotation matrix $\boldsymbol{R}^{bn}$.

3.2. Vision

A vision sensor is a rather complex device composed of a number of sub-systems. The optical system bundles incident rays of light and forms an analog image, which is digitized by the image sensor. Image processing then extracts distinct 2D features in the image and associates them to 3D points in the scene. The correspondences obtained this way are considered as the measurements from the vision system in this paper. The remainder of this section is devoted to describing these vision measurements and the associated camera models.

3.2.1. Camera Models

The image formation process is accomplished by two elements: the optical system or objective and the image sensor. Models for both are briefly discussed in this section.

One of the most commonly used projection models is the pinhole model. According to this model the relation between an image point $\boldsymbol{p}^i_a = (u, v)^T$ of the analog image and its corresponding scene point $\boldsymbol{p}^c = (X, Y, Z)^T$ is given by

$$\lambda \begin{pmatrix} u \\ v \\ f \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{8}$$

where $\lambda > 0$ is a scale factor and f is the focal length. In order to use this homogeneous equation to predict $\boldsymbol{p}^i_a$, λ has to be eliminated. This yields the well-known equation

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{9}$$

Although widely used in computer vision, the pinhole camera model is only suitable for perspective objectives with limited field of view. A more generic model also suitable for wide

angle lenses and omnidirectional cameras is given by Scaramuzza et al. (2006),

$$\lambda \begin{pmatrix} u \\ v \\ f(\rho) \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{10a}$$

with $\lambda > 0$. The constant focal length has been replaced with an nth-order polynomial of radius $\rho \triangleq \sqrt{u^2 + v^2}$,

$$f(\rho) \triangleq \sum_{i=0}^{n} \alpha_i \rho^i. \tag{10b}$$

Note that this more general model (10) includes the pinhole model (8) as a special case. To change from homogeneous coordinates to Euclidean coordinates, we solve for λ using the last line in (10a). After some algebra, one obtains

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{\beta}{r} \begin{pmatrix} X \\ Y \end{pmatrix}, \tag{11a}$$

where $r \triangleq \sqrt{X^2 + Y^2}$ and β is the positive real root of the equation

$$\sum_{i=0}^{n} \alpha_i \beta^i - \frac{Z}{r}\beta = 0. \tag{11b}$$

Finding a closed form expression for this root can be very hard and is even impossible when $n > 4$. However, numerical evaluation is straightforward.

Several applications, including camera calibration and sensor fusion, require derivatives of (11). Of course, these can be calculated numerically. However, a closed form expression for the derivative is given by

$$\frac{\partial \boldsymbol{p}_a^i}{\partial \boldsymbol{p}^c} = \frac{\beta}{\gamma r^3} \begin{pmatrix} X \\ Y \end{pmatrix} \begin{pmatrix} XZ & YZ & -r^2 \end{pmatrix} + \frac{\beta}{r^3} \begin{bmatrix} Y^2 & -XY & 0 \\ -XY & X^2 & 0 \end{bmatrix}, \tag{12a}$$

with γ defined as

$$\gamma \triangleq Z - r \sum_{i=1}^{n} i \alpha_i \beta^{i-1}. \tag{12b}$$

Cameras deliver digital images with coordinates typically specified in pixels and indexed from the top left corner. Furthermore, there is the possibility of non-square as well as non-orthogonal pixels. These properties introduce (non-uniform) scaling and a principal point offset and can be accounted for by an affine transformation which transforms the analog image coordinates $\boldsymbol{p}_a^i = (u, v)^T$ into pixel coordinates $\boldsymbol{p}^i = (x, y)^T$,

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} s_x & s_\theta \\ 0 & s_y \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}. \tag{13}$$

Here, the transformation is composed of the pixel sizes s_x, s_y, the principal point coordinates x_0, y_0 and a skew parameter s_θ.

Equations (8)–(13) contain a number of parameters which have to be determined individually for every camera. The process of doing so is referred to as camera calibration and is a well-known problem in computer vision for which a number of toolboxes have been developed, see, e.g., Zhang (2000), Bouguet (2003) and Scaramuzza et al. (2006). Typically, these require images at several angles and distances of a known calibration object. A planar checkerboard pattern (see Figure 1) is a frequently used calibration object because it is very simple to produce, it can be printed with a standard printer, and has distinctive corners which are easy to detect. Hence, without loss of generality we assume that the camera has been calibrated.

3.2.2. Vision Measurements

The camera measurements $\boldsymbol{y}_t$ consist of the $k = 1, \ldots, K$ correspondences $\boldsymbol{p}_{t,k}^i \leftrightarrow \boldsymbol{p}_{t,k}^n$ between a 2D image feature $\boldsymbol{p}_{t,k}^i$ and its corresponding 3D position in the real world $\boldsymbol{p}_{t,k}^n$. Introducing the notation $\mathcal{P}$ for a projection function summarizing the models in Section 3.2.1, the correspondences are modeled as

$$\boldsymbol{p}_{k,t}^i = \mathcal{P}(\boldsymbol{p}_{k,t}^c) + \boldsymbol{e}_{k,t}^i, \quad \boldsymbol{p}_{k,t}^c = \mathcal{T}(\boldsymbol{p}_{k,t}^n), \tag{14}$$

where $\boldsymbol{e}_{k,t}^i$ is i.i.d. Gaussian noise. For a standard camera, the projection function $\mathcal{P}$ is composed of (9) and (13), whereas for a wide angle camera $\mathcal{P}$ is composed of (11) and (13). Note that $\mathcal{P}$ operates on $\boldsymbol{p}_{k,t}^c$, whereas the correspondences measurements give $\boldsymbol{p}_{k,t}^n$. The required coordinate transformation $\mathcal{T}$ will be discussed in Section 3.3.

In general, finding the correspondences is a difficult image processing problem where two tasks have to be solved. The first task consists of detecting points of interest or features in the image. Here, features are distinctive elements in the camera image, for instance, corners, edges, or textured areas. Common algorithms include the gradient-based Harris detector (Harris and Stephens 1988), the Laplace detector (Mikolajczyk et al. 2005), and the correlation-based Kanade–Lucas–Tomasi tracker (Shi and Tomasi 1994).

Once a feature has been found, it needs to be associated to a known 3D point in the scene in order to form a correspondence. This is the second task, which can be solved using probabilistic methods such as RANSAC (Fischler and Bolles 1981). However, it can be drastically simplified by making use

of some kind of descriptor of the feature which uniquely identifies it by providing information of the local image such as image patches or local histograms. This descriptor should preferably be invariant to scale changes and affine transformations. Common examples are SIFT (Lowe 2004) and more recently SURF (Bay et al. 2008) and FERNS (Ozuysal et al. 2007).

The measurement model (14) and hence our calibration algorithm works with any kind of correspondences. Without loss of generality we simplify the correspondence generation problem and work with checkerboard patterns of known size typically used for camera calibration. In this case, obtaining the correspondences is relatively easy due to the strong corners and simple planar geometry. The required image processing is typically implemented in off-the-shelf camera calibration software, e.g., Bouguet (2003) and Scaramuzza et al. (2006).

3.3. Dynamics

The inertial and vision measurement models are linked by a process model, which describes the motion of the sensor unit. Since it is hard to make informative assumptions regarding general sensor unit movement, the inertial sensors are used as inputs $\boldsymbol{u}_t$ for the process model instead of treating them as measurements. Following the derivation of Hol (2008), we have

$$\boldsymbol{b}^n_{t+1} = \boldsymbol{b}^n_t + T\dot{\boldsymbol{b}}^n_t + \frac{T^2}{2}\ddot{\boldsymbol{b}}^n_t, \tag{15a}$$

$$\dot{\boldsymbol{b}}^n_{t+1} = \dot{\boldsymbol{b}}^n_t + T\ddot{\boldsymbol{b}}^n_t, \tag{15b}$$

$$\mathring{\boldsymbol{q}}^{bn}_{t+1} = e^{-(T/2)\boldsymbol{\omega}^b_{nb,t}} \odot \mathring{\boldsymbol{q}}^{bn}_t, \tag{15c}$$

where $\boldsymbol{b}^n$ and $\dot{\boldsymbol{b}}^n$ denote the position and velocity of the b frame resolved in the n frame, $\mathring{\boldsymbol{q}}^{bn}$ is a unit quaternion describing the orientation of the b frame relative to the n frame and T denotes the sampling interval. Furthermore, $\odot$ is the quaternion multiplication and the quaternion exponential is defined as a power series, similar to the matrix exponential,

$$e^{(0,\boldsymbol{v})} \triangleq \sum_{n=0}^{\infty} \frac{(0,\boldsymbol{v})^n}{n!} = \left(\cos\|\boldsymbol{v}\|, \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\sin\|\boldsymbol{v}\|\right). \tag{16}$$

More details about unit quaternions and their use can be found in Kuipers (1999). The acceleration $\ddot{\boldsymbol{b}}^n_t$ and angular velocity $\boldsymbol{\omega}^b_{nb,t}$ are calculated from the accelerometer signal $\boldsymbol{u}_{a,t}$ and the gyroscope signal $\boldsymbol{u}_{\omega,t}$ according to

$$\ddot{\boldsymbol{b}}^n_t = \boldsymbol{R}^{nb}_t\boldsymbol{u}_{a,t} + \boldsymbol{g}^n - \boldsymbol{R}^{nb}_t\boldsymbol{\delta}^b_a - \boldsymbol{R}^{nb}_t\boldsymbol{e}^b_{a,t}, \tag{17a}$$

$$\boldsymbol{\omega}^b_{nb,t} = \boldsymbol{u}_{\omega,t} - \boldsymbol{\delta}^b_\omega - \boldsymbol{e}^b_{\omega,t}. \tag{17b}$$

The bias terms $\boldsymbol{\delta}^b_a$ and $\boldsymbol{\delta}^b_\omega$ are slowly time-varying and typically included in the process model (15). However, in this paper they are modeled as constants, since a few seconds of data are typically sufficient for calibration.

The process model (15) contains the pose of the body coordinate frame $\boldsymbol{R}^{bn}_t$, $\boldsymbol{b}^n_t$. Hence, a 3D scene point $\boldsymbol{p}^n_{t,k}$ can be expressed in camera coordinates using the transformation

$$\boldsymbol{p}^c_{t,k} = \mathcal{T}(\boldsymbol{p}^n_{k,t}) = \boldsymbol{R}^{cb}(\boldsymbol{R}^{bn}_t(\boldsymbol{p}^n_{t,k} - \boldsymbol{b}^n_t) - \boldsymbol{c}^b). \tag{18}$$

Here, $\boldsymbol{R}^{cb}$ is the rotation matrix which gives the orientation of the c frame with respect to the b frame and $\boldsymbol{c}^b$ is the position of the c frame with respect to the b frame.

3.4. The Predictor

Combining (14), (15), (17) and (18) we obtain a discrete-time non-linear state-space model

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}) + \boldsymbol{w}_t, \tag{19a}$$

$$\boldsymbol{y}_t = h(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + \boldsymbol{e}_t. \tag{19b}$$

This model is described by the state vector $\boldsymbol{x}_t$ and parameterized by the vector $\boldsymbol{\theta}$ which are defined as

$$\boldsymbol{x}_t = \begin{pmatrix} (\boldsymbol{b}^n)^T & (\dot{\boldsymbol{b}}^n)^T & (\mathring{\boldsymbol{q}}^{bn})^T \end{pmatrix}^T, \tag{20}$$

$$\boldsymbol{\theta} = \begin{pmatrix} (\boldsymbol{\varphi}^{cb})^T & (\boldsymbol{c}^b)^T & (\boldsymbol{\delta}^b_\omega)^T & (\boldsymbol{\delta}^b_a)^T & (\boldsymbol{g}^n)^T \end{pmatrix}^T. \tag{21}$$

Besides the relative pose $\boldsymbol{\varphi}^{cb}$ and $\boldsymbol{c}^b$, $\boldsymbol{\theta}$ contains several parameters that we are not directly interested in, so-called nuisance parameters, for example the gyroscope bias $\boldsymbol{\delta}^b_\omega$ and the accelerometer bias $\boldsymbol{\delta}^b_a$. Even though we are not directly interested in these nuisance parameters, they affect the estimated camera trajectory and they have to be taken into account to obtain accurate estimates of $\boldsymbol{\varphi}^{cb}$ and $\boldsymbol{c}^b$.

For a given $\boldsymbol{\theta}$, the state-space model (19) is fully specified and can be used in sensor fusion methods such as the EKF (Kailath et al. 2000). In an EKF, the state estimate $\hat{\boldsymbol{x}}_{t|t}$ and its covariance $\boldsymbol{P}_{t|t}$ are recursively calculated using the time update

$$\hat{\boldsymbol{x}}_{t|t-1} = f(\hat{\boldsymbol{x}}_{t-1|t-1}, \boldsymbol{\theta}), \tag{22a}$$

$$\boldsymbol{P}_{t|t-1} = \boldsymbol{F}_t\boldsymbol{P}_{t-1|t-1}\boldsymbol{F}^T_t + \boldsymbol{Q}_{t-1}, \tag{22b}$$

together with the measurement update

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{K}_t\left(\boldsymbol{y}_t - h(\hat{\boldsymbol{x}}_{t|t-1}, \boldsymbol{\theta})\right), \tag{23a}$$

$$\boldsymbol{P}_{t|t} = \boldsymbol{P}_{t|t-1} - \boldsymbol{P}_{t|t-1}\boldsymbol{H}^T_t\boldsymbol{S}^{-1}_t\boldsymbol{H}_t\boldsymbol{P}_{t|t-1}, \tag{23b}$$

$$\boldsymbol{S}_t = \boldsymbol{H}_t\boldsymbol{P}_{t|t-1}\boldsymbol{H}^T_t + \boldsymbol{R}_t, \tag{23c}$$

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{H}^T_t\boldsymbol{S}^{-1}_t. \tag{23d}$$

Here, $\boldsymbol{F}_t = \mathrm{D}_{\boldsymbol{x}_t} f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta})$, $\boldsymbol{H}_t = \mathrm{D}_{\boldsymbol{x}_t} h(\boldsymbol{x}_t, \boldsymbol{\theta})$, $\boldsymbol{Q}_t = \mathrm{Cov}\,\boldsymbol{w}_t$ and $\boldsymbol{R}_t = \mathrm{Cov}\,\boldsymbol{e}_t$. The different sampling rates of the inertial and the vision sensors are handled straightforwardly. Time

updates (22) are performed at the high data rate of the IMU, whereas the measurement updates (23) are only applied when a new image is available.

Note that, as a part of its processing, the EKF computes a one-step ahead predictor by applying the measurement model $h(\cdot)$ to the state prediction $\hat{\boldsymbol{x}}_{t|t-1}$. This predictor defines exactly the type of mapping we are looking for, and hence we define the predictor introduced in (3) to be

$$\hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta}) \triangleq h(\hat{\boldsymbol{x}}_{t|t-1}(\boldsymbol{\theta}), \boldsymbol{\theta}), \tag{24}$$

where h is the measurement model and $\hat{\boldsymbol{x}}_{t|t-1}(\boldsymbol{\theta})$ is the one-step ahead state prediction of the EKF whose dependency on $\boldsymbol{\theta}$ is here explicitly denoted.

4. Calibration Algorithms

A calibration algorithm determines the model parameters $\boldsymbol{\theta}$ which provide the best match between the predicted and the observed measurements. Based on the prediction error method, extensively used in the system identification community (Ljung 1999), we will derive in Siction 4.1 our calibration algorithm. This algorithm relies on a reasonable initial guess of the parameters to be estimated, which motivates Section 4.2, where an algorithm for finding a suitable initial guess is provided.

4.1. Gray-box Calibration

With the predictor (24) derived in the previous section we are now ready to pose the optimization problem (4), which will allow us to find the relative pose between the camera and the IMU. This problem is posed using the prediction error method. The goal of this method is to find the parameter vector $\boldsymbol{\theta}$ that minimizes the prediction error

$$\boldsymbol{\varepsilon}_t(\boldsymbol{\theta}) = \boldsymbol{y}_t - \hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta}), \tag{25}$$

i.e., the difference between the one-step ahead prediction from the model $\hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta})$ and the observed measurement $\boldsymbol{y}_t$. Note that for the used predictor (24), the prediction errors (25) are commonly referred to as the innovations. In order to find the parameter $\boldsymbol{\theta}$ that provides the smallest (in terms of variance) prediction error we will employ the well-known quadratic cost function,

$$V_N(\boldsymbol{\theta}, \boldsymbol{\varepsilon}) = \frac{1}{2}\sum_{t=1}^{N} \boldsymbol{\varepsilon}_t^T \boldsymbol{\Lambda}_t \boldsymbol{\varepsilon}_t, \tag{26}$$

where $\boldsymbol{\Lambda}_t$ is a suitable weighting matrix. For a correctly tuned EKF, the prediction errors $\boldsymbol{\varepsilon}_t$ are normal distributed according to

$$\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\boldsymbol{0},\ \boldsymbol{S}_t), \tag{27}$$

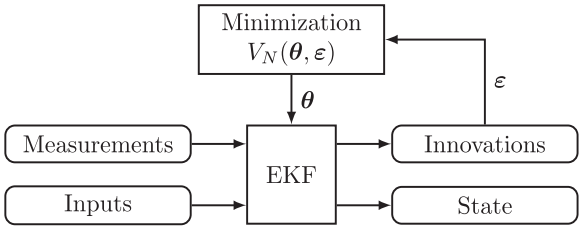


Fig. 4. Gray-box system identification using EKF innovations as prediction errors. The parameter vector $\boldsymbol{\theta}$ is adjusted to minimize the cost function $V_N(\boldsymbol{\theta}, \boldsymbol{\varepsilon})$ given in (28).

with S_t defined in (23c). Inserting (27) into (26) results in

$$V_N(\boldsymbol{\theta}, \boldsymbol{\varepsilon}) = \frac{1}{2}\sum_{t=1}^{N} \boldsymbol{\varepsilon}_t^T \boldsymbol{S}_t^{-1} \boldsymbol{\varepsilon}_t = \frac{1}{2}\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}, \tag{28}$$

where the Nn_y-dimensional vector $\boldsymbol{\epsilon} = \left(\boldsymbol{\epsilon}_1^T, \ldots, \boldsymbol{\epsilon}_N^T\right)$ is constructed by stacking the normalized innovations

$$\boldsymbol{\epsilon}_t = \boldsymbol{S}_t^{-1/2} \boldsymbol{\varepsilon}_t \tag{29}$$

on top of each other. Here it is worth noting that in the resulting cost function (28) the prediction errors $\boldsymbol{\varepsilon}_t$ are weighted by their corresponding inverse covariance. This is rather intuitive, since the covariance contains information about the relative importance of the corresponding component $\boldsymbol{\varepsilon}_t$. Finally, substituting (28) and (25) into (4), the optimization problem becomes

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\sum_{t=1}^{N} \|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta})\|^2_{S_t^{-1}(\boldsymbol{\theta})}. \tag{30}$$

The covariance of the obtained estimate $\hat{\boldsymbol{\theta}}$ (Ljung 1999) is given as

$$\operatorname{Cov}\hat{\boldsymbol{\theta}} = \frac{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}{N n_y} \left([D_{\boldsymbol{\theta}}\boldsymbol{\epsilon}][D_{\boldsymbol{\theta}}\boldsymbol{\epsilon}]^T\right)^{-1}, \tag{31}$$

where $D_{\boldsymbol{\theta}}\boldsymbol{\epsilon}$ is the Jacobian of the normalized prediction error $\boldsymbol{\epsilon}$ with respect to the parameters θ. Note that both the normalized prediction error $\boldsymbol{\epsilon}$ and the Jacobian $D_{\boldsymbol{\theta}}\boldsymbol{\epsilon}$ are evaluated at the obtained estimate $\hat{\boldsymbol{\theta}}$.

An overview of the approach is given in Figure 4. Note that all the quantities in (30) are computed by processing the complete dataset with the EKF, given an estimate $\hat{\boldsymbol{\theta}}$. This makes it an offline calibration, which does not constrain its applicability. The optimization problem (30) is a non-linear least-squares problem and standard methods, such as Gauss–Newton and Levenberg–Marquardt apply, see, e.g., Nocedal and Wright (2006). These methods only guarantee convergence to a local minimum and it is a well-known fact that it can be hard to find the global optimum of $V_N(\boldsymbol{\theta},\boldsymbol{Z})$ for physically parameterized

Algorithm 1 Relative Pose Calibration

1. Place a camera calibration pattern on a horizontal, level surface, e.g., a desk or the floor.
2. Acquire inertial measurements $\{\boldsymbol{u}_{a,t}\}_{t=1}^{M}, \{\boldsymbol{u}_{\omega,t}\}_{t=1}^{M}$ as well as images $\{\boldsymbol{I}_t\}_{t=1}^{N}$:
   - Rotate about all three axes, with sufficiently exciting angular velocities.
   - Always keep the calibration pattern in view.
3. Obtain the point correspondences between the 2D feature locations $\boldsymbol{p}_{t,k}^{i}$ and the corresponding 3D grid coordinates $\boldsymbol{p}_{t,k}^{n}$ of the calibration pattern for all images $\{\boldsymbol{I}_t\}_{t=1}^{N}$.
4. Solve the gray-box identification problem (30), starting the optimization from $\theta_0 = ((\hat{\varphi}_0^{cb})^T, \mathbf{0}, \mathbf{0}, \mathbf{0}, (\boldsymbol{g}_0^n)^T)^T$. Here, $\boldsymbol{g}_0^n = (0, 0, -g)^T$ since the calibration pattern is placed horizontally and $\hat{\varphi}_0^{cb}$ can be obtained using Algorithm 2.
5. Validate the calibration result by analyzing the obtained state trajectory, normalized innovations and parameter covariance (31). If necessary, start over from Step 2.

models (Ljung 2008). This problem is reduced by exploiting the structure of the problem in order to derive a good initial guess for the parameters. This is the topic of the subsequent section.

We can now introduce Algorithm 1, a flexible algorithm for estimating the relative pose of the IMU and the camera.

The dataset is captured without requiring any additional hardware, except for a standard camera calibration pattern of known size that can be produced with a standard printer. The motion of the sensor unit can be arbitrary, provided it contains sufficient rotational excitation. A convenient setup for the data capture is to mount the sensor unit on a tripod and pan, tilt and roll it, but hand-held sequences can be used equally well.

Solving (30) yields relative position and orientation, as well as nuisance parameters such as sensor biases and gravity. The optimization is started in $\boldsymbol{\theta}_0 = ((\hat{\varphi}_0^{cb})^T, \mathbf{0}, \mathbf{0}, \mathbf{0}, (\boldsymbol{g}_0^n)^T)^T$. Here, $\boldsymbol{g}_0^n = (0, 0, -g)^T$ since the calibration pattern is placed horizontally and $\hat{\varphi}_0^{cb}$ can be obtained using Algorithm 2, which will be described shortly. It is worth noting that the optimization problem (30) is quite flexible and parameters can easily be removed if they are already known.

4.2. Initialization of Parameter Estimates

An initial estimate of the relative orientation can be obtained simply by performing a standard camera calibration, similar to Lobo and Dias (2007). Placing the calibration pattern on a horizontal, level surface, a vertical reference can be obtained from the extrinsic parameters. Furthermore, when holding the sensor unit still, the accelerometers measure only gravity. From these two ingredients an initial orientation can be obtained using Theorem 1, originally by Horn (1987). It has been extended with expressions for the Jacobian, facilitating computation of the covariance.

Theorem 1 (Relative orientation). *Suppose* $\{\boldsymbol{v}_t^a\}_{t=1}^{N}$ *and* $\{\boldsymbol{v}_t^b\}_{t=1}^{N}$ *are measurements satisfying* $\boldsymbol{v}_t^a = \mathring{\boldsymbol{q}}^{ab} \odot \boldsymbol{v}_t^b \odot \mathring{\boldsymbol{q}}^{ba}$. *Then the sum of the squared residuals,*

$$V(\mathring{\boldsymbol{q}}^{ab}) = \sum_{t=1}^{N} \|e_t\|^2 = \sum_{t=1}^{N} \|\boldsymbol{v}_t^a - \mathring{\boldsymbol{q}}^{ab} \odot \boldsymbol{v}_t^b \odot \mathring{\boldsymbol{q}}^{ba}\|^2, \quad (32)$$

is minimized by $\widehat{\mathring{\boldsymbol{q}}}^{ab} = \boldsymbol{x}_1$, *where* $\boldsymbol{x}_1$ *is the eigenvector corresponding to the largest eigenvalue* λ_1 *of the system* $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ *with*

$$\boldsymbol{A} = -\sum_{t=1}^{N} (\boldsymbol{v}_t^a)_L (\boldsymbol{v}_t^b)_R. \quad (33)$$

Here, the quaternion operators $\cdot_L, \cdot_R$ *are defined as*

$$\mathring{\boldsymbol{q}}_L \triangleq \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix},$$

$$\mathring{\boldsymbol{q}}_R \triangleq \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}. \quad (34)$$

Furthermore, the Jacobians of $\widehat{\mathring{\boldsymbol{q}}}^{ab}$ *with respect to the measurements are given by*

$$\mathrm{D}_{v_i^a} \widehat{\mathring{\boldsymbol{q}}}^{ab} = -[(\widehat{\mathring{\boldsymbol{q}}}^{ab})^T \otimes (\lambda_1 \boldsymbol{I}_4 - \boldsymbol{A})^{\dagger}] \times [\boldsymbol{I}_4 \otimes (\boldsymbol{v}_i^b)_R][\mathrm{D}_v \boldsymbol{v}_L], \quad (35a)$$

$$\mathrm{D}_{v_t^b} \widehat{\mathring{\boldsymbol{q}}}^{ab} = -[(\widehat{\mathring{\boldsymbol{q}}}^{ab})^T \otimes (\lambda_1 \boldsymbol{I}_4 - \boldsymbol{A})^{\dagger}] \times [\boldsymbol{I}_4 \otimes (\boldsymbol{v}_t^a)_L][\mathrm{D}_v \boldsymbol{v}_R], \quad (35b)$$

where $\otimes$ is the Kronecker product and † is the Moore–Penrose pseudo inverse. The Jacobians $\mathrm{D}_v v_L$ and $\mathrm{D}_v v_R$ are defined as

$$\mathrm{D}_v v_L = \begin{bmatrix} \mathring{e}_R^0 \\ \mathring{e}_R^1 \\ \mathring{e}_R^2 \\ \mathring{e}_R^3 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix}, \quad \mathrm{D}_v v_R = \begin{bmatrix} \mathring{e}_L^0 \\ \mathring{e}_L^1 \\ \mathring{e}_L^2 \\ \mathring{e}_L^3 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ I_3 \end{bmatrix},$$

where $\{\mathring{e}^i\}_{i=1}^4$ is the standard basis in $\mathbb{R}^4$.

Proof. See Appendix B.

The procedure to obtain an initial orientation estimate is shown in Algorithm 2. Note that $g^n = (0, 0, -g)^T$, since the calibration pattern is placed horizontally.

Algorithm 2 Initial Orientation

1. Place a camera calibration pattern on a horizontal, level surface, e.g., a desk or the floor.
2. Acquire images $\{I_t\}_{t=1}^N$ of the pattern while holding the sensor unit static in various poses, simultaneously acquiring accelerometer readings $\{u_{a,t}\}_{t=1}^N$.
3. Perform a camera calibration using the images $\{I_t\}_{t=1}^N$ to obtain the orientations $\{\mathring{q}_t^{cn}\}_{t=1}^N$.
4. Compute an estimate $\widehat{\mathring{q}}^{cb}$ from the vectors $g_t^c = R_t^{cn} g^n$ and $g_t^b = -u_{a,t}$ using Theorem 1.

5. Experiments and Results

The sensor unit introduced in Section 1 has been equipped with both perspective and fisheye lenses, see Figure 5. In both configurations the sensor unit has been calibrated according to Algorithm 1, using nothing but a planar checkerboard pattern of known size as in a standard camera calibration setup. The calibration data was gathered according to the following protocol:

1. The checkerboard pattern is placed on a horizontal, planar surface.
2. The sensor unit is held stationary in 8–12 different poses, similar to what is done during a standard camera calibration. For each pose, a single image is captured together with 1 s of inertial measurements at 100 Hz.

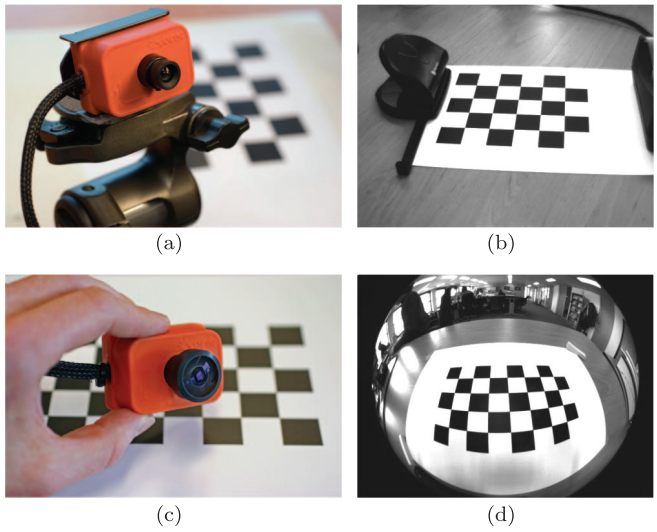


Fig. 5. Two configurations of the sensor unit. In (a) and (b) a 4 mm perspective lens is used and in (c) and (d) a 190° fisheye lens is used.

3. The sensor unit is subjected to 10–12 s of rotational motion around all three axes, while keeping the calibration pattern in view. The angular velocity during this rotational motion should be similar to the application being calibrated for. The inertial data is sampled at 100 Hz and the camera has a frame rate of 25 Hz. Due to the limited field of view, the sensor unit is mounted on a tripod and rotated in pan, tilt, and roll direction, when equipped with the perspective lens. For the fisheye configuration, hand-held sequences are used.

The measurements obtained in Step 2 are used in Algorithm 2 to determine an initial orientation. The measurements from Step 3 are used in Algorithm 1 to estimate the relative translation and orientation between the camera and the IMU. An example of a typical trajectory is given in Figure 6.

To facilitate cross-validation, the measurements are split into an estimation part and a validation part (Ljung 1999), both containing similar motion. The parameters are estimated from the estimation data and the quality of the estimates is assessed using the validation data. Sample calibration datasets are included in Extension 1.

5.1. Calibration Results

A number of different sensor units and/or different lens configurations have been calibrated using the above protocol. The resulting estimates of the relative position and orientation of the camera and the IMU, c^b and φ^{cb}, together with their standard deviation calculated using (31), are listed in Table 1.

Table 1. Relative pose estimates and 99% confidence intervals for five different sensor units and several different lens configurations.

Unit	Lens	$\hat{\varphi}^{cb}$ (°)	$\hat{c}^{b}$ (mm)
1	4 mm	$(-0.52,\ 0.43,\ 0.94) \pm 0.04$	$(-17.6, -4.8, 22.1) \pm 0.9$
2	6 mm	$(0.23, -0.34,\ 0.02) \pm 0.05$	$(-17.6, -6.2, 28.3) \pm 1.4$
3	6 mm	$(-0.53,\ 0.97,\ 0.29) \pm 0.02$	$(-14.9, -6.7, 29.8) \pm 0.5$
4	6 mm	$(-0.02,\ 0.21, -0.20) \pm 0.04$	$(-18.1, -8.7, 31.0) \pm 0.9$
5	6 mm	$(-0.27,\ 0.94,\ 0.09) \pm 0.13$	$(-14.0, -7.0, 30.3) \pm 1.3$
5	Fisheye	$(0.08,\ 0.17,\ 0.06) \pm 0.14$	$(-17.4, -4.9, 38.7) \pm 0.4$
Reference[a]		$(0,\ 0,\ 0)$	$(-14.5, -6.5,\ —)$

[a] Using the CCD position and orientation of the technical drawing.

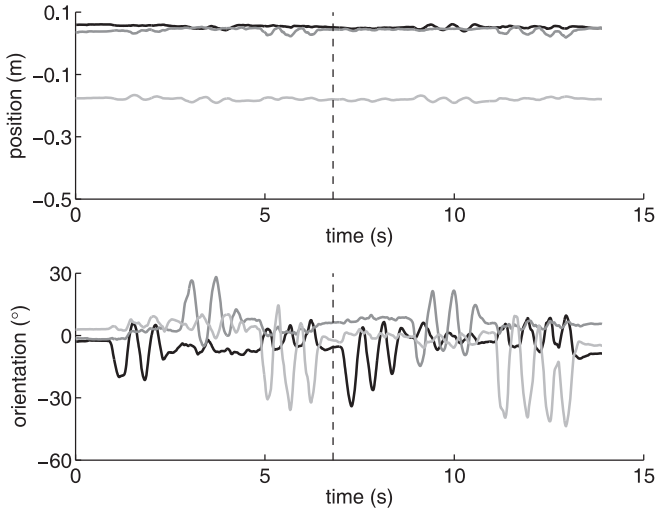


Fig. 6. Example trajectory of the sensor unit used for calibration. It contains both estimation data ($t < 6.8$ s) and validation data ($t \geq 6.8$ s), separated by the dashed line.

Table 1 also contains reference values obtained from the technical drawing. Note that the drawing defines the center of the CCD, not the optical center of the lens. Hence, no height reference is available and some shifts can occur in the tangential directions.

Table 1 shows that Algorithm 1 has been successfully applied to five different sensor units equipped with both perspective and fisheye lenses. In order to clearly show that the obtained estimates are indeed good, further validation is need, which will be provided below. Consistent results are obtained for multiple trials of the same configuration, which further reinforces the robustness and reliability of the proposed method, although the confidence measure for the fisheye lens is found to be slightly conservative. This could be caused by the unrealistic assumption that the correspondence noise is homogeneous over the entire image.

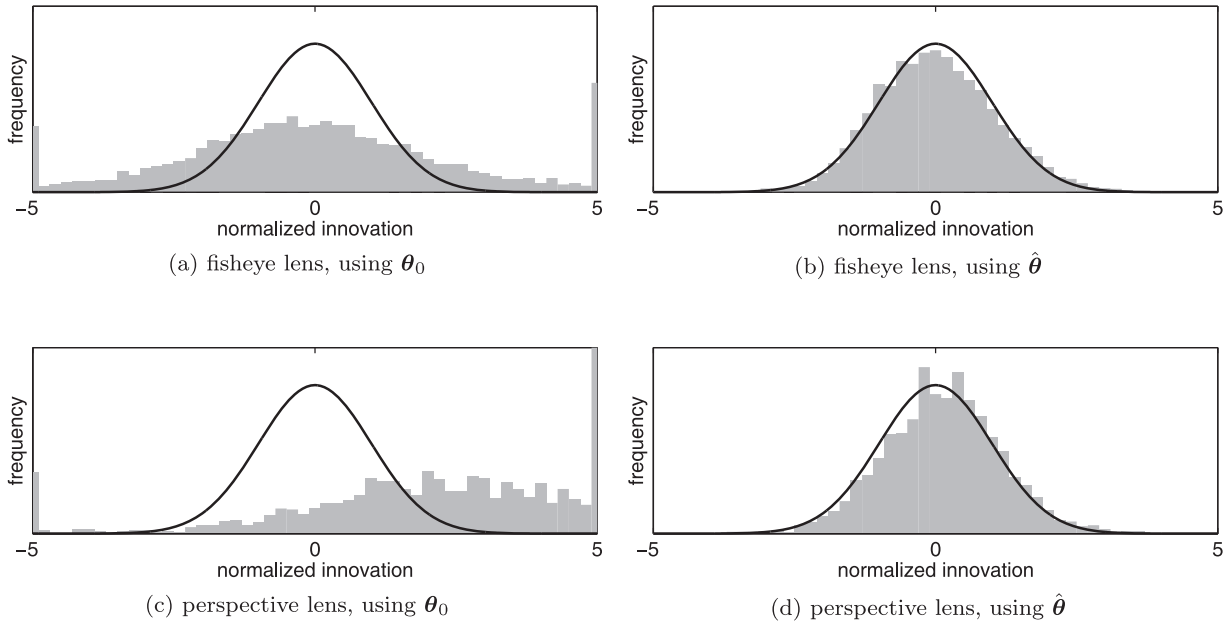
In order to further validate the estimates, the normalized innovations ϵ_t, computed according to (29), are studied. Histograms of the normalized innovations (for validation data) are given in Figure 7. Figures 7(a) and 7(c) show the effect of using wrong parameter vectors, in this case being the initial guess. After calibration, the normalized innovations are close to white noise, as shown in Figures 7(b) and 7(d). This implies that the model with the estimated parameters and its assumptions appear to be correct, which in turn is a very good indication that reliable estimates $\hat{\varphi}^{cb}$ and $\hat{c}^{b}$ have been obtained.

The calibration results shown in Table 1 are close to the reference values, but show individual differences between the different sensor units and lens configurations. These differences are significant, which is further illustrated in Figure 8. This figure illustrates the behavior when applying the calibration values of one sensor unit to a second sensor unit having the same type of lens.

Notice the characteristic saw-tooth behavior present in the position plot. It is present in all three position channels and explains the big difference between the obtained normalized innovations and the theoretic distribution. When the correct calibration parameters are used this saw-tooth behavior is absent, which is illustrated in Figure 6. To summarize, the significant individual differences once more illustrate the need for an easy-to-use calibration method, since each sensor unit has to be individually calibrated for optimal performance.

5.2. Sensitivity Analysis

The proposed calibration algorithm has been subjected to a sensitivity analysis to determine its behavior for varying signal-to-noise conditions and for different geometric conditions. The fisheye dataset (see Section 5 and Extension 1), has been used as a starting point for Monte-Carlo simulations.



(a) fisheye lens, using $\boldsymbol{\theta}_0$

(b) fisheye lens, using $\hat{\boldsymbol{\theta}}$

(c) perspective lens, using $\boldsymbol{\theta}_0$

(d) perspective lens, using $\hat{\boldsymbol{\theta}}$

Fig. 7. Histograms of the normalized innovations, for validation data. Both the empirical distribution (gray bar) as well as the theoretical distribution (black line) are shown.

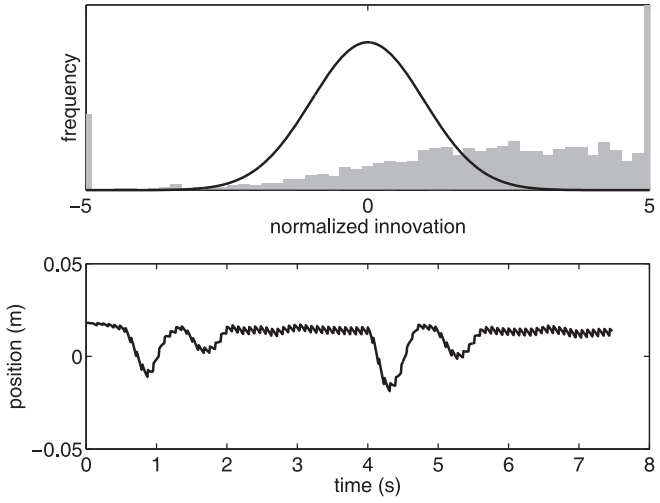


Fig. 8. Typical behavior obtained when using sensor unit A with calibration values of sensor unit B. The figure shows the empirical distribution (grey bar) and the theoretic distribution (black line) (top) as well as the x-position trajectory (bottom).

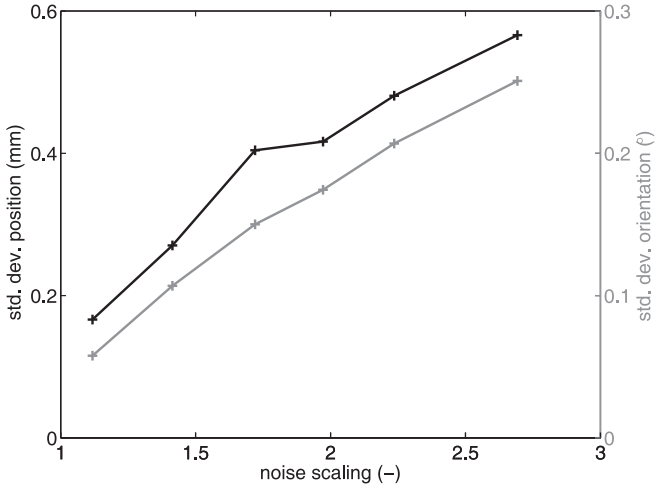


Fig. 9. Standard deviation of the relative position estimate $\hat{\boldsymbol{c}}^b$ (black) and the relative orientation estimate $\hat{\boldsymbol{\varphi}}^{cb}$ (gray) at various noise levels.

The signal-to-noise ratio has been modified by adding noise proportional to its original magnitude to all measurements, i.e., the accelerometer signals, the gyroscope signals and the feature locations. For a number of noise scalings the calibration parameters for the modified dataset have been determined for $N = 100$ noise realizations. Standard deviations, $\sigma(\boldsymbol{x}) = (\mathrm{tr}\,\mathrm{Cov}\,\boldsymbol{x})^{1/2}$, of the position estimate $\hat{\boldsymbol{c}}^b$ and the orientation estimate $\hat{\boldsymbol{\varphi}}^{cb}$ are shown in Figure 9. As expected, the calibration accuracy degrades with increased noise levels.

The geometric conditions have been modified by scaling of the checkerboard pattern. To this end, the feature locations in the dataset are replaced by simulated ones. These have been

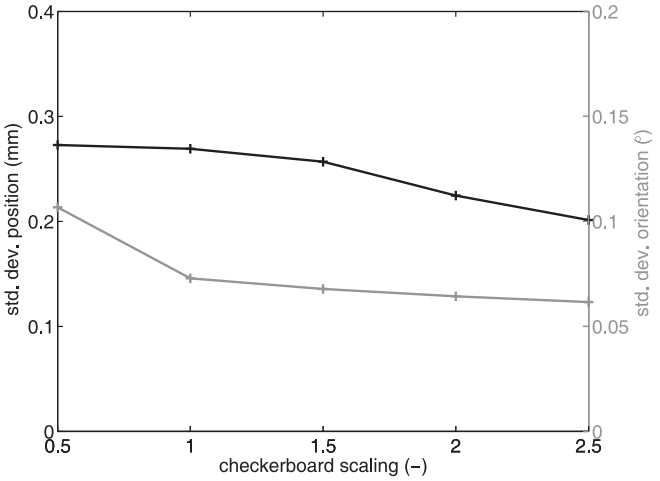


Fig. 10. Standard deviation of the relative position estimate $\hat{\boldsymbol{c}}^b$ (black) and the relative orientation estimate $\hat{\boldsymbol{\varphi}}^{cb}$ (gray) at various sizes of the checkerboard.

simulated using the camera trajectory estimated as part of the calibration procedure on the original dataset and realistic noise has been added. For a number of scale factors the calibration parameters for the modified dataset have been determined for $N = 100$ noise realizations. Standard deviations of the position estimate $\hat{\boldsymbol{c}}^b$ and the orientation estimate $\hat{\boldsymbol{\varphi}}^{cb}$ are shown in Figure 10. It can be concluded that a larger checkerboard pattern results in reduced variance of the calibration parameters.

6. Conclusion

In this paper we propose a new calibration method to determine the relative position and orientation of an IMU and a camera that are rigidly connected. The method is based on a physical model of the sensor unit which can also be used in solving, for example, sensor fusion problems. Both perspective and wide angle lenses are handled by the approach. This solves the important calibration problem, enabling successful integration of vision and inertial sensors in many applications. The experiments indicate that the proposed algorithm is easy to use. Even small displacements and misalignments can be accurately calibrated from short measurement sequences made using the standard camera calibration setup.

Acknowledgment

This work was partly supported by the strategic research center MOVIII, funded by the Swedish Foundation for Strategic Research, SSF.

Appendix A: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extension

Extension	Type	Description
1	Data	Sample calibration datasets

Appendix B: Proof for Theorem 1

Analogous to the original proof by Horn (1987), the squared residuals can be written as

$$\|\boldsymbol{e}_t\|^2 = \|\boldsymbol{v}_t^a\|^2 - 2\boldsymbol{v}_t^a \cdot (\mathring{\boldsymbol{q}}^{ab} \odot \boldsymbol{v}_t^b \odot \mathring{\boldsymbol{q}}^{ba}) + \|\boldsymbol{v}_t^b\|^2.$$

Minimization only affects the middle term, which can be simplified to

$$\begin{aligned}\boldsymbol{v}_t^a \cdot (\mathring{\boldsymbol{q}}^{ab} \odot \boldsymbol{v}_t^b \odot \mathring{\boldsymbol{q}}^{ba}) &= -(\boldsymbol{v}_t^a \odot (\mathring{\boldsymbol{q}}^{ab} \odot \boldsymbol{v}_t^b \odot \mathring{\boldsymbol{q}}^{ba}))_0 \\ &= -(\boldsymbol{v}_t^a \odot \mathring{\boldsymbol{q}}^{ab})^T (\boldsymbol{v}_t^b \odot \mathring{\boldsymbol{q}}^{ba})^c \\ &= -(\mathring{\boldsymbol{q}}^{ab})^T (\boldsymbol{v}_t^a)_L (\boldsymbol{v}_t^b)_R \mathring{\boldsymbol{q}}^{ab},\end{aligned}$$

using the relation $(\mathring{\boldsymbol{a}} \odot \mathring{\boldsymbol{b}})_0 = \mathring{\boldsymbol{a}}^T \mathring{\boldsymbol{b}}^c$ for the scalar part of quaternion multiplication. The minimization problem can now be restated as

$$\arg\min_{\|\mathring{\boldsymbol{q}}^{ab}\|=1} \sum_{t=1}^{N} \|\boldsymbol{e}_t\|^2 = \arg\max_{\|\mathring{\boldsymbol{q}}^{ab}\|=1} (\mathring{\boldsymbol{q}}^{ab})^T \boldsymbol{A} \mathring{\boldsymbol{q}}^{ab},$$

where $\boldsymbol{A}$ is defined in (33). Note that the matrices $\cdot_L$ and $\cdot_R$ commute, i.e., $\mathring{\boldsymbol{a}}_L \mathring{\boldsymbol{b}}_R = \mathring{\boldsymbol{b}}_R \mathring{\boldsymbol{a}}_L$, since $\mathring{\boldsymbol{a}}_L \mathring{\boldsymbol{b}}_R x = \mathring{\boldsymbol{a}} \odot \mathring{\boldsymbol{x}} \odot \mathring{\boldsymbol{b}} = \mathring{\boldsymbol{b}}_R \mathring{\boldsymbol{a}}_L \mathring{\boldsymbol{x}}$ for all $\mathring{\boldsymbol{x}}$. Additionally, $\cdot_L$ and $\cdot_R$ are skew symmetric for vectors. This implies that

$$\begin{aligned}(\boldsymbol{v}_t^a)_L (\boldsymbol{v}_t^b)_R &= [-(\boldsymbol{v}_t^a)_L^T][-(\boldsymbol{v}_t^b)_R^T] = [(\boldsymbol{v}_t^b)_R (\boldsymbol{v}_t^a)_L]^T \\ &= [(\boldsymbol{v}_t^a)_L (\boldsymbol{v}_t^b)_R]^T,\end{aligned}$$

from which can be concluded that $\boldsymbol{A}$ is a real symmetric matrix.

Let $\mathring{\boldsymbol{q}}^{ab} = \boldsymbol{X}\boldsymbol{\alpha}$ with $\|\boldsymbol{\alpha}\| = 1$, where $\boldsymbol{X}$ is an orthonormal basis obtained from the symmetric eigenvalue decomposition of $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{X}^T$. Then,

$$(\mathring{\boldsymbol{q}}^{ab})^T \boldsymbol{A} \mathring{\boldsymbol{q}}^{ab} = \boldsymbol{\alpha}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{X}^T \boldsymbol{X}\boldsymbol{\alpha} = \sum_{i=1}^{4} \alpha_i^2 \lambda_i \leq \lambda_1,$$

where λ_1 is the largest eigenvalue. Equality is obtained for $\boldsymbol{\alpha} = (1, 0, 0, 0)^T$, that is, $\widehat{\mathring{\boldsymbol{q}}}^{ab} = \boldsymbol{x}_1$.

Extending Horn (1987), the sensitivity of the solution can be determined based on an analysis of the real symmetric

eigenvalue equation, $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. The Jacobian of the eigenvector $\boldsymbol{x}(\boldsymbol{A})$ is given by

$$\mathrm{D}_{\boldsymbol{A}}\boldsymbol{x} = \boldsymbol{x}^T \otimes (\lambda_1 \boldsymbol{I}_4 - \boldsymbol{A})^{\dagger}$$

as derived by Magnus and Neudecker (1999). Furthermore, writing $\boldsymbol{A}_t = -\boldsymbol{R}_t\boldsymbol{L}_t = -\boldsymbol{L}_t\boldsymbol{R}_t$ one can show that

$$\mathrm{d}\boldsymbol{A}_t(\boldsymbol{L}_t) = -\boldsymbol{R}_t(\mathrm{d}\boldsymbol{L}_t) \quad \Leftrightarrow \quad \mathrm{D}_{\boldsymbol{L}_t}\boldsymbol{A}_t = -\boldsymbol{I}_4 \otimes \boldsymbol{R}_t,$$

$$\mathrm{d}\boldsymbol{A}_t(\boldsymbol{R}_t) = -\boldsymbol{L}_t(\mathrm{d}\boldsymbol{R}_t) \quad \Leftrightarrow \quad \mathrm{D}_{\boldsymbol{R}_t}\boldsymbol{A}_t = -\boldsymbol{I}_4 \otimes \boldsymbol{L}_t.$$

Straightforward application of the chain rule results in

$$\mathrm{D}_{v_t^a}\widehat{\boldsymbol{q}}^{ab} = [\mathrm{D}_{\boldsymbol{A}}\boldsymbol{x}][\mathrm{D}_{\boldsymbol{L}_t}\boldsymbol{A}][\mathrm{D}_{v_t^a}\boldsymbol{L}_t],$$

$$\mathrm{D}_{v_t^b}\widehat{\boldsymbol{q}}^{ab} = [\mathrm{D}_{\boldsymbol{A}}\boldsymbol{x}][\mathrm{D}_{\boldsymbol{R}_t}\boldsymbol{A}][\mathrm{D}_{v_t^b}\boldsymbol{R}_t].$$

Evaluating this expression gives (35).

References

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, **13**(3): 108–117.

Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. (2008). Speeded-up robust features (SURF). *Journal of Computer Vision and Image Understanding*, June, 346–359.

Bleser, G. and Stricker, D. (2008). Advanced tracking through efficient image processing and visual-inertial sensor fusion. *Proceedings of IEEE Virtual Reality Conference*, Reno, NE, USA, March, pp. 137–144.

Bouguet, J.-Y. (2003). Camera calibration toolbox for matlab, 2003. http://www.vision.caltech.edu/bouguetj/calib_doc/. Accessed 2 April 2008.

Chandaria, J., Thomas, G. A. and Stricker, D. (2007). The MATRIS project: real-time markerless camera tracking for augmented reality and broadcast applications. *Journal of Real-Time Image Processing*, **2**(2): 69–79.

Chatfield, A. (1997). *Fundamentals of High Accuracy Inertial Navigation*, vol. 174, 3rd edn. American Institute of Aeronautics and Astronautics, USA.

Corke, P., Lobo, J. and Dias, J. (2007). An introduction to inertial and visual sensing. *International Journal of Robotics Research*, **26**(6): 519–535.

Daniilidis, K. (1999). Hand-eye calibration using dual quaternions. *International Journal of Robotics Research*, **18**(3): 286–298.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics & Automation Magazine*, **13**(2): 99–110.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, **24**(6): 381–395.

Foxlin, E. and Naimark, L. (2003). Miniaturization, calibration & accuracy evaluation of a hybrid self-tracker. *Proceedings of 2nd International Symposium on Mixed and Augmented Reality*, Tokyo, Japan, October, pp. 151–160.

Graebe, S. (1990). Theory and implementation of gray box identification. *Ph.D. Thesis*, Royal Institute of Technology, Stockholm, Sweden.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, Manchester, UK, pp. 147–151.

Hol, J. D. (2008). Pose estimation and calibration algorithms for vision and inertial sensors. *Licentiate Thesis no 1379*, Department of Electrical Engineering, Linköping University, Sweden.

Hol, J. D., Schön, T. B. and Gustafsson, F. (2008a). A new algorithm for calibrating a combined camera and IMU sensor unit. *Proceedings of 10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, December, pp. 1857–1862.

Hol, J. D., Schön, T. B. and Gustafsson, F. (2008b). Relative pose calibration of a spherical camera and an IMU. *Proceedings of 8th International Symposium on Mixed and Augmented Reality*, Cambridge, UK, September, pp. 21–24.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, **4**(4): 629–642.

Kailath, T., Sayed, A. H. and Hassibi, B. (2000). *Linear Estimation*. Upper Saddle River, NJ, Prentice Hall.

Kuipers, J. B. (1999). *Quaternions and Rotation Sequences*. Princeton, NJ, Princeton University Press.

Ljung, L. (1999). *System Identification: Theory for the User*, 2nd edn. Upper Saddle River, NJ, Prentice Hall.

Ljung, L. (2008). Perspectives on system identification. *Proceedings of 17th International Federation of Automatic Control World Congress*, Seoul, South Korea, July, pp. 7172–7184.

Ljung, L. and Söderström, T. (1983). *Theory and Practice of Recursive Identification*. The MIT Press series in Signal Processing, Optimization, and Control. Cambridge, MA, MIT Press.

Lobo, J. and Dias, J. (2007). Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research*, **26**(6): 561–575.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**(2): 91–110.

Magnus, J. R. and Neudecker, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 2nd edn. Chichester, UK, Wiley.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, **65**(1): 43–72.

Mirzaei, F. M. and Roumeliotis, S. I. (2008). A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, **24**(5): 1143–1156.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. New York, Springer.

Ozuysal, M., Fua, P. and Lepetit, V. (2007). Fast keypoint recognition in ten lines of code. *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, MI, June, pp. 1–8.

Scaramuzza, D., Martinelli, A. and Siegwart, R. (2006). A toolbox for easily calibrating omnidirectional cameras. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October, pp. 5695–5701.

Shi, J. and Tomasi, C. (1994). Good features to track. *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Seattle, WA, June, pp. 593–600.

Shuster, M. D. (1993). A survey of attitude representations. *The Journal of the Astronautical Sciences*, **41**(4): 439–517.

Titterton, D. H. and Weston, J. L. (1997). Strapdown inertial navigation technology. *IEE Radar, Sonar, Navigation and Avionics Series.* Stevenage, UK, Peter Peregrinus Ltd.

Tsai, R. Y. and Lenz, R. K. (1989). A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, **5**(3): 345–358.

Woodman, O. J. (2007). An introduction to inertial navigation. *Technical Report UCAM-CL-TR-696*, University of Cambridge, Computer Laboratory.

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