2008 10th Intl. Conf. on Control, Automation, Robotics and Vision
Hanoi, Vietnam, 17–20 December 2008

A New Algorithm for Calibrating a Combined Camera and IMU Sensor Unit

Jeroen D. Hol
Xsens Technologies B.V.
Pantheon 6a, Postbus 559
7500 AN Enschede, The Netherlands
jeroen.hol@xsens.com

Thomas B. Schön, Fredrik Gustafsson
Division of Automatic Control
Linköping University
SE–581 83 Linköping, Sweden
{schon, fredrik}@isy.liu.se

***Abstract*—This paper is concerned with the problem of estimating the relative translation and orientation between an inertial measurement unit and a camera which are rigidly connected. The key is to realise 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. Furthermore, covariance expressions are provided for all involved estimates. The experimental results shows that the method works well in practice.**

***Index Terms*—Gray-box system identification, Kalman filter, Calibration, IMU, Camera.**

I. Introduction

This paper is concerned with the problem of estimating the translation and orientation between a camera and an inertial measurement unit (IMU) that are rigidly connected. Accurate knowledge of this translation and orientation is important for high quality sensor fusion using the measurements from both sensors. The sensor unit used in this work is shown in Fig. 1. For more information about this particular sensor unit, see [1, 2].

The IMU supports the vision algorithms by providing high-dynamic motion measurements that enable accurate predictions where features can be expected in the upcoming frame. This facilitates development of real-time pose estimation and feature detection/association algorithms, which are the cornerstones for a number of applications including augmented reality, vision based navigation and simultaneous localization and tracking (SLAM).

Fig. 1: The sensor unit, consisting of an IMU and a camera. The camera calibration (checkerboard) pattern is visible in the background.

The basic performance measure in all these applications is how accurately the features positions are predicted. Let this measure be a general cost function $V(c, \varphi, C, S)$ that measures the sum of all feature prediction errors (measured in pixels) over time and space, where

- c, φ denote the relative position and orientation between the IMU sensor and camera optical center.
- C denotes the intrinsic camera parameters.
- S denotes the sensor offsets in the IMU. These are partly factory calibrated but the time-variability makes their influence non-negligible and important to consider.

Camera calibration is a standard problem [3, 4], which can be solved using a camera calibration pattern printed using any standard office printer. That is, we assume that C is already calibrated and known.

We here propose to use a weighted quadratic cost function $V(c, \varphi, C, S)$ and treate the problem within the standard gray-box framework available from the system identification community [5–7]. This approach requires a prediction model, where the IMU sensor data is used to predict camera motion, and a Kalman filter is used to compute the sequence of innovations over the calibration batch of data. The cost function then consists of the normalized sum of squared innovations. Minimizing the cost function V over the parameters (c, φ, S) yields the nonlinear least squares (NLS) estimate. In case of Gaussian noise this estimate is also the maximum likelihood (ML) estimate.

It is well known that gray-box identification problems often requires good initial values to work, so initialization is an important issue. In particular, orientation has turned out be critical here. We here make use of a theorem by Horn [8] to find an initial orientation.

The resulting algorithm is fast and simple to apply in practice. Typically, waving the camera over a checkerboard for a couple of seconds gives enough excitation and information for accurately estimating the parameters. This is a significant improvement over previous work on this problem, see e.g., [9], where additional hardware is typically required.

978-1-4244-2287-6/08/$25.00

II. Problem Formulation

We will in this section give a more formal formulation of the problem we are trying to solve. The first thing to do is to introduce the three coordinate frames that are needed,

- **Earth (e):** The camera pose is estimated with respect to this coordinate system, which is fixed to the earth. It can be aligned in any way, however, preferably it should be vertically aligned.
- **Camera (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. The camera acquires its images in the image plane (i). This plane is perpendicular to the optical axis and is located at an offset (focal length) from the optical center of the camera.
- **Body (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.

These coordinate frames are used to denote geometric quantities of interest, for instance, b^{e} is the position of the body coordinate frame expressed in the earth frame and $q^{\mathrm{be}}, \varphi^{\mathrm{be}}, R^{\mathrm{be}}$ is the unit quaternion, rotation vector or rotation matrix, respectively, describing the rotation from the earth frame to the body frame. These rotation parameterizations are interchangeable. In Fig. 2 the relationship between the coordinate frames is illustrated. Note that the b frame is rigidly connected to the

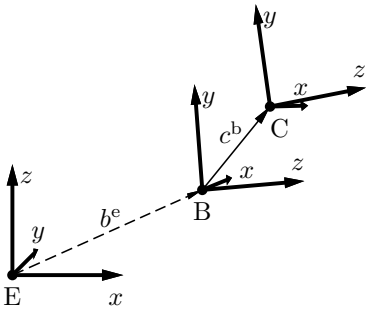


Fig. 2: The coordinate frames. The sensor unit consists of an IMU (b frame) and a camera (c frame) that are rigidly connected, i.e., c^{b} and φ^{cb} are constant. The rigid connection is illustrated using a solid line, whereas the dashed line indicates that this vector changes over time as the sensor is moved.

c frame. The aim in this paper is to device an algorithm that is capable of estimating the following parameters,

- The relative orientation between the body and the camera frames, parameterized using a rotation vector φ^{cb}.
- The relative position of these frames c^{b}, i.e., the position of the camera frame expressed in the body frame.

We will use θ to denote all the parameters to be estimated, which besides φ^{cb} and c^{b} will contain several parameters that we are not directly interested in, so called nuisance parameters, for example bias terms in the gyroscopes and the accelerometers. Even though we are not directly interested in these nuisance parameters, they have to be known to compute accurate estimates of φ^{cb} and c^{b}.

In order to compute estimates we need information about the system, provided by measurements. The measured data is denoted Z,

$$Z = \{u_1, \ldots, u_M, y_1, \ldots, y_N\}, \tag{1}$$

where u_t denote the input signals and y_t denote the measurements. In the present work the data from the inertial sensors is modelled as input signals and the information from the camera is modelled as measurements. Note that the inertial sensors are typically sampled at a higher frequency than the camera, motivating the use of M and N in (1). In this work the inertial data is sampled at 100 Hz and the camera has a frame rate of 25 Hz.

The problem of computing estimates of θ based on the information in Z is a standard gray-box system identification problem, see e.g., [5–7]. The parameters are typically estimated using the prediction error method, which has been extensively studied, see e.g., [7]. The idea used in the prediction error method is very simple, minimize the difference between the measurements and the predicted measurements obtained from a model of the system at hand. This prediction error is given by

$$\varepsilon_t(\theta) = y_t - \hat{y}_{t|t-1}(\theta), \tag{2}$$

where $\hat{y}_{t|t-1}(\theta)$ is used to denote the one-step ahead prediction from the model. The parameters are now found by minimizing a norm of the prediction errors. Here, the common choice of a quadratic cost function is used,

$$V_N(\theta, Z) = \frac{1}{N}\sum_{t=1}^{N} \frac{1}{2}\varepsilon_t^T(\theta)\Lambda_t^{-1}\varepsilon_t(\theta), \tag{3}$$

where Λ_t is a symmetric positive definite matrix that is chosen according to the relative importance of the corresponding component $\varepsilon_t(\theta)$. Finally, the parameter estimates are given by

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

which using (3) is a nonlinear least-squares problem and standard methods, such as Gauss-Newton and Levenberg-Marquardt [10, 11], apply.

It is worth noting that if Λ_t is chosen as the covariance of the innovations, the cost function (3) corresponds to the well known and statistically well behaved maximum likelihood method. In other words, the maximum likelihood method is a special case of the more general prediction error method.

III. Finding the Predictor

We need a predictor in order to solve (4). Finding such a predictor is the topic of the present section. The dynamic model that will be used in the predictor is described in the subsequent section and in Section III-B we briefly explain how the camera measurements are incorporated. Finally, the predictor is discussed in Section III-C.

A. Motion Model

The task of the motion model is to describe the motion of the sensor unit based on the inputs u_t. This boils down to purely kinematic relations describing motion in the absence of considerations of mass and force. In other words, it is a matter of assigning coordinate frames to rigid bodies and explaining how these move over time.

The input signals $u(t_k)$ arrive at discrete time instants, explaining the notation $u(t_k)$, where subindex k is used to denote discrete time. The input signals are given by

$$u(t_k) = \begin{pmatrix} u_{\mathrm{a}}^{\mathrm{b}}(t_k) \\ u_{\omega}^{\mathrm{b}}(t_k) \end{pmatrix}, \tag{5}$$

where $u_{\mathrm{a}}^{\mathrm{b}}(t_k)$ and $u_{\omega}^{\mathrm{b}}(t_k)$ denote the specific force and the angular velocity reported by the inertial sensors, both resolved in the b frame. The accelerometer signal $u_{\mathrm{a}}^{\mathrm{b}}(t_k)$ from the IMU is modelled according to

$$u_{\mathrm{a}}^{\mathrm{b}}(t_k) = R^{\mathrm{be}}(t_k)(\ddot{b}^{\mathrm{e}}(t_k) - g^{\mathrm{e}}) + \delta_{\mathrm{a}}^{\mathrm{b}} + e_{\mathrm{a}}^{\mathrm{b}}(t_k), \tag{6}$$

where R^{be} is the rotation matrix from the e to the b frame, $\ddot{b}^{\mathrm{e}}$ is the acceleration of the b frame resolved in the e frame, g^{e} is the gravity vector resolved in the e frame, $\delta_{\mathrm{a}}^{\mathrm{b}}$ denotes the bias term present in the accelerometers and $e_{\mathrm{a}}^{\mathrm{b}}$ denotes zero mean i.i.d. Gaussian noise. The gyroscope signal $u_{\omega}^{\mathrm{b}}(t_k)$ is modelled as

$$u_{\omega}^{\mathrm{b}}(t_k) = \omega_{\mathrm{eb}}^{\mathrm{b}}(t_k) + \delta_{\omega}^{\mathrm{b}} + e_{\omega}^{\mathrm{b}}(t_k), \tag{7}$$

where $\omega_{\mathrm{eb}}^{\mathrm{b}}$ denotes the angular velocity of the b frame relative to the e frame resolved in the b frame, $\delta_{\omega}^{\mathrm{b}}$ is a bias term, and e_{ω}^{b} denotes zero mean i.i.d. Gaussian noise. Note that the bias terms $\delta_{\mathrm{a}}^{\mathrm{b}}$ and $\delta_{\omega}^{\mathrm{b}}$ are in fact slowly time-varying. However, for the purpose of this work it is sufficient to model them as constants since short data sequences, typically a few seconds, are used.

The acceleration $\ddot{b}^{\mathrm{e}}$ and the angular velocity $\omega_{\mathrm{eb}}^{\mathrm{b}}$ are solved from (6) and (7) and are used in combination with the following state vector,

$$x(t) = \begin{pmatrix} b^{\mathrm{e}}(t)^T & \dot{b}^{\mathrm{e}}(t)^T & q^{\mathrm{be}}(t)^T \end{pmatrix}^T, \tag{8}$$

where $b^{\mathrm{e}}(t)$ denotes the position of the b frame resolved in the e frame. Furthermore, $\dot{b}^{\mathrm{e}}(t)$ denotes the velocity of the b frame resolved in the e frame and $q^{\mathrm{be}}(t)$ is a unit quaternion describing the orientation of the b frame relative to the e frame (that is $q^{\mathrm{be}}(t)$ describes the rotation from the e frame to the b frame). There are several alternative possibilities when it comes to state vectors, see [1] for a detailed discussion on this topic. Using the state vector (8), the continuous-time state-space model is given by

$$\frac{\partial b^{\mathrm{e}}(t)}{\partial t} = \dot{b}^{\mathrm{e}}(t), \tag{9a}$$

$$\frac{\partial \dot{b}^{\mathrm{e}}(t)}{\partial t} = \ddot{b}^{\mathrm{e}}(t), \tag{9b}$$

$$\frac{\partial q^{\mathrm{be}}(t)}{\partial t} = -\frac{1}{2}\omega_{\mathrm{eb}}^{\mathrm{b}}(t) \odot q^{\mathrm{be}}(t), \tag{9c}$$

where $\odot$ is the quaternion product. The derivation of (9c) together with the necessary quaternion algebra can be found in [1].

We will approximate the continuous-time model (9) with a discrete-time model. In order to obtain a discrete-time state-space model we invoke the commonly used assumption that the input signals are piecewise constant between the sampling instants, i.e.,

$$u(t) = u(t_k), \qquad kT \leq t < (k+1)T, \tag{10}$$

where T denotes the sampling interval. The resulting discrete-time state-space model is obtained by making use of the assumption (10) in (9) and carrying out the necessary integrations. This is a standard procedure discussed for instance in [12]. With a slight abuse of notation we have

$$b_{t+1}^{\mathrm{e}} = b_t^{\mathrm{e}} + T\dot{b}_t^{\mathrm{e}} + \frac{T^2}{2}\ddot{b}_t^{\mathrm{e}}, \tag{11a}$$

$$\dot{b}_{t+1}^{\mathrm{e}} = \dot{b}_t^{\mathrm{e}} + T\ddot{b}_t^{\mathrm{e}}, \tag{11b}$$

$$q_{t+1}^{\mathrm{be}} = e^{-\frac{T}{2}\omega_{\mathrm{eb},t}^{\mathrm{b}}} \odot q_t^{\mathrm{be}}, \tag{11c}$$

where $\ddot{b}_t^{\mathrm{e}}$ and $\omega_{\mathrm{eb},t}^{\mathrm{b}}$ are given by

$$\ddot{b}_t^{\mathrm{e}} = R_t^{\mathrm{eb}} u_{\mathrm{a},t}^{\mathrm{b}} + g^{\mathrm{e}} - R_t^{\mathrm{eb}}\delta_a^b - R_t^{\mathrm{eb}} e_{\mathrm{a},t}^b, \tag{12a}$$

$$\omega_{\mathrm{eb},t}^{\mathrm{b}} = u_{\omega,t}^{\mathrm{b}} - \delta_{\omega}^{\mathrm{b}} - e_{\omega,t}^{\mathrm{b}}. \tag{12b}$$

The quaternion exponential used in (11c) is defined as a power series, similar to the matrix exponential,

$$e^{(0,v)} \triangleq \sum_{n=0}^{\infty} \frac{(0,v)^n}{n!} = \left(\cos\|v\|, \frac{v}{\|v\|}\sin\|v\|\right). \tag{13}$$

B. Camera Measurements

The camera measurements y_t are constructed from the $k = 1, \ldots, N$ correspondences $p_{t,k}^{\mathrm{i}} \leftrightarrow p_{t,k}^{\mathrm{e}}$ between a 2D image feature $p_{t,k}^{\mathrm{i}}$ and the corresponding 3D position in the real world $p_{t,k}^{\mathrm{e}}$. In general, finding these correspondences is a difficult problem. However, for the special case of the checkerboard patterns used in camera calibration it is relatively easy to obtain the correspondences and off-the-shelf software is available, e.g., [4].

For a calibrated perspective camera, the camera measurements can be modeled using a particularly simple form,

$$\begin{aligned} y_{t,k} &= h(x_t, \theta) + e_{\mathrm{c},t} \\ &= \begin{bmatrix} -I_2 & p_{t,k}^{\mathrm{i},n} \end{bmatrix} R^{\mathrm{cb}}(R^{\mathrm{be}}(p_{t,k}^{\mathrm{e}} - b_t^{\mathrm{e}}) - c^{\mathrm{b}}) + e_{\mathrm{c},t}. \end{aligned} \tag{14}$$

Here $p_{t,k}^{\mathrm{e}}$ is a position in 3D space with $p_{t,k}^{\mathrm{i},n}$ its coordinates in a normalized image, R_t^{cb} is the rotation matrix which gives the orientation of the c frame w.r.t. the b frame, and $e_{\mathrm{c},t}$ is zero mean i.i.d. Gaussian noise.

C. The Predictor

The state-space model describing the motion of the sensor unit is given in (11), its input signals from the IMU are given in (12) and the measurement equation for the correspondences

is given in (14). Altogether this is a standard discrete-time nonlinear state-space model parameterized by

$$\theta = \begin{pmatrix} (\varphi^{\text{cb}})^T & (c^{\text{b}})^T & (\delta^{\text{b}}_{\omega})^T & (\delta^{\text{b}}_{\text{a}})^T & (g^{\text{e}})^T \end{pmatrix}^T \tag{15}$$

Hence, for a given θ it is straightforward to make use of the extended Kalman filter (EKF) [13, 14] to compute the one-step ahead predictor $\hat{y}_{t|t-1}(\theta)$. It is straightforward to see that the covariance S_t for the prediction error (2) is given by

$$S_t = C_t P_{t|t-1} C_t^T + R_t, \tag{16}$$

where the state covariance $P_{t|t-1}$, the measurement Jacobian C_t and the measurement covariance R_t are provided by the EKF. The weights in (4) are chosen as $\Lambda_t = S_t$.

IV. Algorithms

All the parts that are needed to assemble the calibration algorithm are now in place. In Section IV-A the algorithm is stated and in Section IV-B we explain how to obtain good initial values for the algorithm.

A. Gray-Box Algorithm

In order to solve (4) we have to compute the gradients

$$\frac{\partial V_N}{\partial \theta} \tag{17}$$

which is done using the structure in the following way,

$$\frac{\partial V_N}{\partial \theta} = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_{t|t-1}(\theta))^T \Lambda_t^{-1} \left(-\frac{\partial \hat{y}_{t|t-1}(\theta)}{\partial \theta} \right) \tag{18}$$

The one-step ahead prediction of the measurement is directly available from the EKF according to

$$\hat{y}_{t|t-1}(\theta) = h(\hat{x}_{t|t-1}, \theta), \tag{19}$$

where $\hat{x}_{t|t-1}$ is the one-step ahead prediction from the EKF. The gradient $\frac{\partial \hat{y}_{t|t-1}(\theta)}{\partial \theta}$ can now straightforwardly be approximated using the central difference approximation. This is the approach taken in this paper. Alternatively, the structure can be further exploited, by writing the gradient of $\hat{y}_{t|t-1}(\theta)$ with respect to θ according to

$$\begin{aligned} \frac{\partial \hat{y}_{t|t-1}(\theta)}{\partial \theta} &= \frac{\partial h(\hat{x}_{t|t-1}, \theta)}{\partial \theta} \\ &= \frac{\partial h(\hat{x}_{t|t-1}, \theta)}{\partial \theta} + \frac{\partial \hat{x}_{t|t-1}^T(\theta)}{\partial \theta} \frac{\partial h(\hat{x}_{t|t-1}, \theta)}{\partial \hat{x}_{t|t-1}(\theta)} \end{aligned} \tag{20}$$

The problem has now been reduced to finding $\frac{\partial \hat{x}_{t|t-1}(\theta)}{\partial \theta}$, which can be handled either by straightforward numerical approximation or by setting up an additional filter.

In order to compute the covariance of the estimate the Nn_y-dimensional vector $\epsilon = (\epsilon_1^T, \ldots, \epsilon_N^T)^T$ is formed by stacking the normalized innovations

$$\epsilon_t = S_t^{-1/2} \left(y_t - \hat{y}_{t|t-1}(\theta) \right) \tag{21}$$

on top of each other. Recall that S_t denotes the covariance for the innovations which is directly available from the EKF, according to (16). Finally, the covariance of the estimate can be computed according to [7]

$$\operatorname{Cov} \hat{\theta} = \frac{\epsilon^T \epsilon}{N n_y} \left([D_\theta \epsilon][D_\theta \epsilon] \right)^{-1}, \tag{22}$$

where the residuals ϵ and the Jacobian's $[D_\theta \epsilon]$ are evaluated at the current estimate $\hat{\theta}$. Now everything is in place for Algorithm 1, which is a flexible algorithm for estimating the relative pose between the IMU and the camera. This algorithm does not require any additional hardware, save for a standard camera calibration pattern that can be produced with a standard printer. Besides relative position and orientation, nuisance parameters like sensor biases and gravity are also determined. The algorithm is very flexible, 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. However, hand-held sequences can be used equally well.

Algorithm 1 Calibration

1) Place a camera calibration pattern on a horizontal, level surface, e.g., a desk or the floor.
2) Acquire inertial data $\{u_{\text{a},t}\}_{t=1}^M, \{u_{\omega,t}\}_{t=1}^M$ and images $\{I_t\}_{t=1}^N$.
   - Rotate around all 3 axes, with sufficiently exiting angular velocities.
   - Always keep the calibration pattern in view.
3) Obtain the point correspondences between the undistorted and normalized 2D feature locations $p_{t,k}^{\text{i}}$ and the corresponding 3D grid coordinates $p_{t,k}^{\text{e}}$ of the calibration pattern for all images $\{I_t\}_{t=1}^N$.
4) Solve the gray-box problem (4), using $\theta_0 = ((\hat{\varphi}_0^{\text{cb}})^T, 0, 0, 0, (g_0^{\text{e}})^T)^T$ as a starting point for the optimization. Here, $g_0^{\text{e}} = (0, 0, -g)^T$ since the calibration pattern is placed horizontally and $\hat{\varphi}_0^{\text{cb}}$ can be obtained using Algorithm 2.
5) Determine the covariance of $\hat{\theta}$ using (22).

B. Finding Initial Values

An initial estimate for the relative orientation can be obtained simply by performing a standard camera calibration. Placing the calibration pattern level, 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 [8]. A simplified and straightforward proof is included for completeness.

Theorem 1 (Relative Orientation): Suppose $\{v_t^{\text{a}}\}_{t=1}^N$ and $\{v_t^{\text{b}}\}_{t=1}^N$ are measurements satisfying $v_t^{\text{a}} = q^{\text{ab}} \odot v_t^{\text{b}} \odot q^{\text{ba}}$. Then the sum of the squared residuals,

$$V(q^{\text{ab}}) = \sum_{t=1}^{N} \|e_t\|^2 = \sum_{t=1}^{N} \|v_t^{\text{a}} - q^{\text{ab}} \odot v_t^{\text{b}} \odot q^{\text{ba}}\|^2, \tag{23}$$

is minimized by $\hat{q}^{\text{ab}} = x_1$, where x_1 is the eigenvector corresponding to the largest eigenvalue λ_1 of the system

$Ax = \lambda x$ with

$$A = -\sum_{t=1}^{N} (v_t^{\mathrm{a}})_L (v_t^{\mathrm{b}})_R. \tag{24}$$

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

$$q_L \triangleq \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{25a}$$

$$q_R \triangleq \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \tag{25b}$$

Proof: The squared residuals in (23) can be written as

$$\|e_t\|^2 = \|v_t^{\mathrm{a}}\|^2 - 2v_t^{\mathrm{a}} \cdot (q^{\mathrm{ab}} \odot v_t^{\mathrm{b}} \odot q^{\mathrm{ba}}) + \|v_t^{\mathrm{b}}\|^2.$$

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

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

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

$$\underset{\|q^{\mathrm{ab}}\|=1}{\arg\min} \sum_{t=1}^{N} \|e_t\|^2 = \underset{\|q^{\mathrm{ab}}\|=1}{\arg\max} (q^{\mathrm{ab}})^T A q^{\mathrm{ab}},$$

where A is defined in (24). Note that the matrices $\cdot_L$ and $\cdot_R$ commute, i.e., $a_L b_R = b_R a_L$, since $a_L b_R x = a \odot x \odot b = b_R a_L x$ for all x. Additionally, $\cdot_L$ and $\cdot_R$ are skew symmetric for vectors. This implies that

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

from which it can be concluded that A is a real symmetric matrix.

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

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

where λ_1 is the largest eigenvalue. Equality is obtained for $\alpha = (1, 0, 0, 0)^T$, that is, $\hat{q}^{\mathrm{ab}} = x_1$. ∎

The exact procedure to obtain an initial orientation estimate is summarized in Algorithm 2. Note that $g^{\mathrm{e}} = \begin{pmatrix} 0 & 0 & -g \end{pmatrix}^T$, 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 $\{u_{\mathrm{a},t}\}_{t=1}^N$.
3) Perform a camera calibration using the images $\{I_t\}_{t=1}^N$ to obtain the orientations $\{q_t^{\mathrm{ce}}\}_{t=1}^N$.
4) Compute an estimate $\hat{q}^{\mathrm{cb}}$ from $g_t^{\mathrm{c}} = R_t^{\mathrm{ce}} g^{\mathrm{e}}$ and $g_t^{\mathrm{b}} = -u_{\mathrm{a},t}$ using Theorem 1.

V. Experiments

Algorithm 1 has been used to calibrate the sensor unit introduced in Section I. This algorithm computes estimates of the relative position and orientation between the IMU and the camera, i.e., c^{b} and φ^{cb}, based on the motion of the sensor unit. This motion can be arbitrary, as long as it is sufficiently exiting in angular velocity and the calibration pattern stays in view. The setup employed is identical to that of a typical camera calibration setup. A number of experiments have been performed. During such an experiment the sensor unit has been rotated around its three axis, see Fig. 3 for an illustration. The measurements contains relatively small

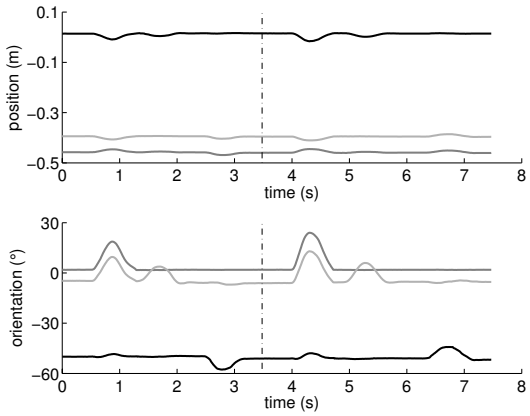


Fig. 3: A trajectory of the sensor unit used for calibration. It contains both estimation data ($t < 3.5$ s) and validation data ($t \geq 3.5$ s), separated by the dashed line.

rotations, since the calibration pattern has to stay in view. However, modest angular velocities are present, which turn out to provide sufficient excitation. The data is split into two parts, one estimation part and one validation part, see Fig. 3. This facilitates cross-validation, where the parameters are estimated using the estimation data and the quality of the estimates can then be assessed using the validation data [7].

In Table I the estimates produced by Algorithm 1 are given together with confidence intervals (99%). Note that the estimates are contained within the 99% confidence intervals. Reference values are also given, these are taken as the result of Algorithm 2 (orientation) and from the technical drawing (position). Note that the drawing defines the position of the CCD, not the optical center. Hence, no height reference is

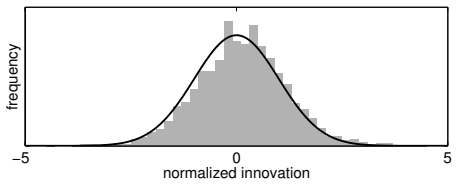
TABLE I: Estimates from Algorithm 1 together with 99% confidence intervals and reference values.

Orientation	$\hat{\varphi}_x^{cb}$ (°)	$\hat{\varphi}_y^{cb}$ (°)	$\hat{\varphi}_z^{cb}$ (°)
Trial 1	-0.06 [-0.28, 0.17]	0.84 [0.67, 1.01]	0.19 [-0.06, 0.44]
Trial 2	-0.19 [-0.36, -0.02]	0.75 [0.62, 0.88]	0.45 [0.23, 0.67]
Trial 3	-0.29 [-0.48, -0.10]	0.91 [0.76, 1.05]	0.08 [-0.11, 0.27]
Reference[a]	-0.23 [-0.29, -0.17]	0.80 [0.73, 0.87]	0.33 [0.22, 0.44]
Position	$\hat{c}_x^b$ (mm)	$\hat{c}_y^b$ (mm)	$\hat{c}_z^b$ (mm)
Trial 1	-13.5 [-15.2, -11.9]	-6.7 [-8.1, -5.2]	34.5 [31.0, 38.0]
Trial 2	-15.7 [-17.3, -14.2]	-8.8 [-10.1, -7.5]	33.2 [28.7, 37.7]
Trial 3	-13.5 [-14.9, -12.0]	-7.3 [-8.6, -6.0]	29.7 [26.8, 32.7]
Reference[b]	-14.5	-6.5	-

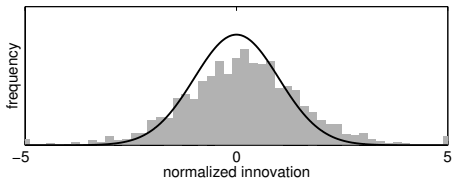
[a] using Algorithm 2 on a large dataset.
[b] using the CCD position of the technical drawing.

available and some shifts can occur in the tangential directions. Table I indicates that the estimates are indeed rather good.

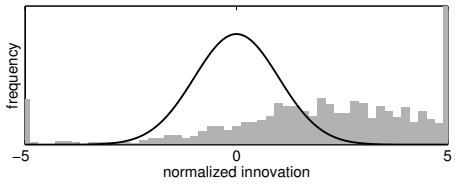
In order to further validate the estimates the normalized innovations (21) are studied. Histograms of the normalized innovations and their autocorrelations are given in Fig. 4 and Fig. 5, respectively. Both figures are generated using the validation data. In Fig. 4b and 4c the effect of using the wrong



(a) using $\hat{\theta}$



(b) using $c^b = 0$



(c) using $\varphi^{cb} = 0$

Fig. 4: Histogram of the normalized innovations, for validation data. Both the empirical distribution (gray bar) as well as the theoretical distribution (black line) are shown.

relative translation and orientation is shown. From Fig. 4a and Fig. 5 it is clear that the normalized innovations are close to white noise. This implies that the model with the estimated parameters and its assumptions appears to be correct, which in turn is a good indication that reliable estimates $\hat{\varphi}^{cb}, \hat{c}^b$ have been obtained. The reliability and repeatability of the estimates has also been confirmed by additional experiments.

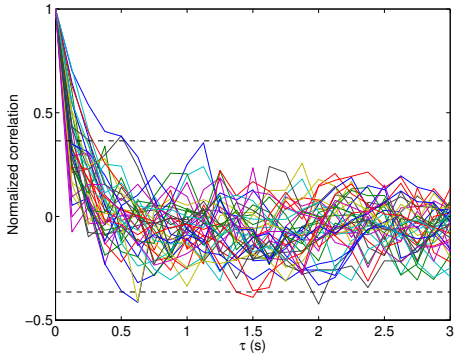


Fig. 5: Autocorrelation of the normalized innovations, for validation data. The horizontal dashed lines indicate the 99% confidence interval.

VI. Conclusion

The experiments indicates that the proposed algorithm is an easy-to-use calibration method to determine the relative position and orientation between an IMU and a camera, that are rigidly connected. 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.

References

[1] J. D. Hol, "Vision and inertial measurements: sensor fusion and calibration," Licentiate Thesis, Department of Electrical Engineering, Linköping University, Sweden, May 2008.

[2] "Xsens technologies," www.xsens.com, 2008, last accessed on March 31, 2008.

[3] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the International Conference on Computer Vision (ICCV)*, Corfu, Greece, Sep. 1999, pp. 666–673.

[4] "Camera calibration toolbox," www.vision.caltech.edu/bouguetj/calib_doc, 2008, last accessed on March 31, 2008.

[5] S. Graebe, "Theory and implementation of gray box identification," Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden, Jun. 1990.

[6] T. Bohlin, *Interactive System Identification: Prospects and pitfalls*. Berlin: Springer, 1991.

[7] L. Ljung, *System identification, Theory for the user*, 2nd ed., ser. System sciences series. Upper Saddle River, NJ, USA: Prentice Hall, 1999.

[8] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, Apr. 1987.

[9] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, Jun. 2007.

[10] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, 1983.

[11] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research. New York, USA: Springer, 2006.

[12] W. J. Rugh, *Linear System Theory*, 2nd ed., ser. Information and system sciences series. Upper Saddle River, NJ, USA: Prentice Hall, 1996.

[13] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, ser. Information and System Sciences Series. Upper Saddle River, NJ, USA: Prentice Hall, 2000.

[14] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, ser. Information and system science series. Englewood Cliffs, NJ, USA: Prentice Hall, 1979.