Relative Pose Calibration of a Spherical Camera and an IMU

Jeroen D. Hol[*]
Xsens Technologies B.V.
Enschede, The Netherlands

Thomas B. Schön[†] Fredrik Gustafsson[‡]
Division of Automatic Control, Linköping University
Linköping, Sweden

Abstract

This paper is concerned with the problem of estimating the relative translation and orientation of an inertial measurement unit and a spherical 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 experimental results show that the method works well in practice.

1 Introduction

This paper is concerned with the problem of estimating the translation and orientation of 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 Figure 1. For more information about this particular sensor unit, see [4, 12].

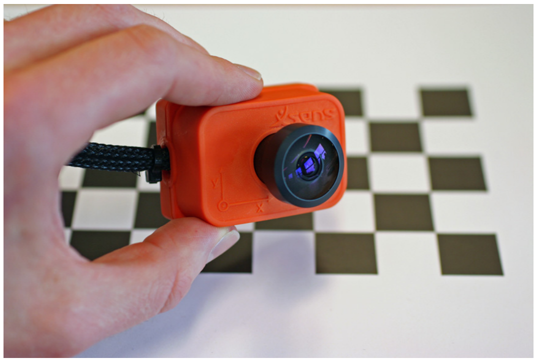


Figure 1: The sensor unit, consisting of an IMU and a spherical camera. The camera calibration pattern is visible in the background.

The combination of vision and inertial sensors is very suitable for augmented reality (AR), see e.g., [1]. An introduction to the technology is given by [2, 4]. 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. Combined with the large field of view of the spherical camera, this facilitates development of robust real-time pose estimation and feature detection/association algorithms, which are the cornerstones for many AR applications.

The basic performance measure in these applications is how accurate the feature positions are predicted. Let this measure be a general cost function $V(p, C, B)$ that measures the sum of all feature prediction errors (measured in pixels) over time, where

- p denotes the relative position and orientation (pose) of the IMU and the optical center of the camera.
- C denotes the intrinsic parameters of the camera. Camera calibration for spherical lenses is a standard problem [7, 11], which can be solved using a camera calibration pattern printed using a standard office printer. That is, we assume that camera is already calibrated and C is known.
- B denotes the parameters of the IMU. The IMU is already factory calibrated, but some non-negligible time-varying sensor offsets remain. Together with gravity, the sensor offsets are nuisance parameters which need to be considered in the calibration procedure.

In this paper we propose to use a weighted quadratic cost function $V(p, C, B)$ and treat the problem within the standard gray-box framework available from the system identification community [8]. This approach requires a prediction model. The key idea is to realize that the camera motion and the image formation process can be described using a nonlinear state-space model implying that an Extended Kalman Filter (EKF) can be used as a predictor. The cost function then consists of the sum of normalized squared innovations over a batch of data. Minimizing the cost function V over the parameters (p, B) 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 identifications often require good initial values to work, so initialization is an important issue. For the problem at hand, orientation turns out to be critical. We make use of a theorem by Horn [5] to align accelerometer readings with the camera verticals and to find an initial orientation estimate.

The proposed calibration algorithm is fast and simple to use 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 and manual effort is required.

2 Problem Formulation

In this section we will 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 environment. The 3D feature positions are assumed to be constant and known in this frame. It can be aligned in any direction, 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), which is perpendicular to the optical axis.

[*] e-mail: jeroen.hol@xsens.com

[†] e-mail: schon@isy.liu.se

[‡] e-mail: fredrik@isy.liu.se

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

In Figure 2 the relationship between the coordinate frames is illustrated. The 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 $q^{be}, \varphi^{be}, R^{be}$ are the unit quaternion, rotation vector and rotation matrix, respectively, describing the rotation from the earth frame to the body frame. These rotation parameterizations are interchangeable. The camera and the IMU are rigidly connected, i.e., c^b and φ^{cb} are constant.

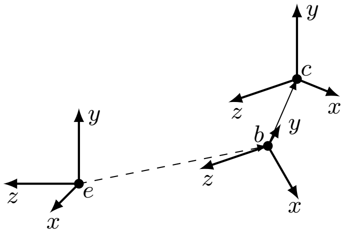


Figure 2: The sensor unit consists of an IMU (b frame) and a camera (c frame). These frames are rigidly connected, denoted by a solid line. The position of the sensor unit with respect to the earth (e frame) changes over time as the unit is moved, denoted with a dashed line.

The goal of this paper is to device 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 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, so called nuisance parameters, for example the sensor offsets of the gyroscopes and the accelerometers. 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 φ^{cb} and c^b.

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

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

where u_t denote the input signals and y_t denote the measurements. In the present work the data from the inertial sensors is 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 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., [3, 8]. The parameters are typically estimated using the prediction error method, which has been extensively studied, see e.g., [8]. 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

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

Using (3), the minimization is a nonlinear least-squares problem and standard methods, such as Gauss-Newton and Levenberg-Marquardt, see e.g., [10], apply.

It is worth noting that if Λ_t is chosen as the covariance of the prediction errors, the estimate (4) becomes the well known and statistically well behaved maximum likelihood estimate. In other words, the maximum likelihood method is a special case of the more general prediction error method.

3 Prediction Model

In order to solve (4) a prediction model $\hat{y}_{t|t-1}(\theta)$ is required. The key idea is to realize that the camera motion and the image formation can be modeled as a discrete-time state-space model.

The task of the motion model is to describe the motion of the sensor unit based on the inputs u_t. Following the derivation of [4], we have

$$b^e_{t+1} = b^e_t + T\dot{b}^e_t + \frac{T^2}{2}\ddot{b}^e_t, \tag{5a}$$

$$\dot{b}^e_{t+1} = \dot{b}^e_t + T\ddot{b}^e_t, \tag{5b}$$

$$q^{be}_{t+1} = e^{-\frac{T}{2}\omega^b_{eb,t}} \odot q^{be}_t, \tag{5c}$$

where b^e and $\dot{b}^e$ denote the position and velocity of the b frame resolved in the e frame, q^{be} is a unit quaternion describing the orientation of the b frame relative to the e 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,

$$e^{(0,v)} \triangleq \sum_{n=0}^{\infty} \frac{(0,v)^n}{n!} = \left(\cos\|v\|, \frac{v}{\|v\|}\sin\|v\|\right). \tag{6}$$

The acceleration $\ddot{b}^e_t$ and angular velocity $\omega^b_{eb,t}$ are modeled using the accelerometers signal u_a and the gyroscope signal u_ω

$$\ddot{b}^e_t = R^{eb}_t u_{a,t} + g^e - R^{eb}_t\delta^b_a - R^{eb}_t e^b_{a,t}, \tag{7a}$$

$$\omega^b_{eb,t} = u_{\omega,t} - \delta^b_\omega - e^b_{\omega,t}. \tag{7b}$$

Here, e^b_a and e^b_ω are i.i.d. Gaussian noises, δ^b_a and δ^b_ω are bias terms and g^e denotes the gravity vector. The bias terms 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 are used. Typically, a few seconds of data is sufficient for calibration.

The camera measurements consist of the $k = 1, \dots, n_y$ correspondences $p^i_{t,k} \leftrightarrow p^e_{t,k}$ between a 2D image feature $p^i_{t,k}$ and the corresponding 3D position in the real world $p^e_{t,k}$. 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., [11].

For a spherical camera, the relation between the normalized image point $p^i_n = (u, v)^T$ and the scene point $p^c = (X, Y, Z)$ is given, see [11], by

$$\lambda \begin{pmatrix} u \\ v \\ f(\rho) \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad f(\rho) \triangleq \sum_{i=0}^{n} \alpha_i \rho^i, \quad \rho \triangleq \sqrt{u^2 + v^2}, \tag{8}$$

for some scale factor $\lambda > 0$. Solving for p^i_n results in

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{P}(p^c) = \frac{\beta}{r} \begin{pmatrix} X \\ Y \end{pmatrix}, \qquad r \triangleq \sqrt{X^2 + Y^2}, \tag{9a}$$

where β is the positive real root of the equation

$$\sum_{i=0}^{n} \alpha_i \beta^i - \frac{Z}{r} \beta = 0. \tag{9b}$$

The complete measurement model is now obtained as

$$p^i_{t,k} = A\mathcal{P}\Big(R^{cb}(R^{be}(p^e_{t,k} - b^e_t) - c^b)\Big) + o^i + e^i_{t,k}. \tag{10}$$

Here $p^e_{t,k}$ is a position in 3D space with $p^i_{t,k}$ its coordinates in the camera image, R^{cb} is the rotation matrix which gives the orientation of the c frame w.r.t. the b frame, c^b is the position of the c frame w.r.t the b frame, and $e^i_{c,t,k}$ is zero mean i.i.d. Gaussian noise. Furthermore, $\mathcal{P}$ is the projection of (9a), A is a scaling matrix and o^i is the image center.

Equations (5) and (10) form a discrete-time nonlinear state-space model parameterized by

$$\theta = \big((\varphi^{cb})^T \quad (c^b)^T \quad (\delta^b_\omega)^T \quad (\delta^b_a)^T \quad (g^e)^T\big)^T \tag{11}$$

Hence, for a given θ it is straightforward to make use of the EKF [6] to compute the one-step ahead predictor $\hat{y}_{t|t-1}(\theta)$ and its covariance S_t. The EKF is run at the high data rate of the IMU and vision updates are only performed when an image is taken. By choosing the weights in (3) as $\Lambda_t = S_t$ the predictor is fully specified.

4 Algorithms

All parts that are needed to assemble the calibration algorithm are now in place, resulting in Algorithm 1. This is a flexible algorithm for estimating the relative pose of the IMU and the spherical camera. It does not require any additional hardware, except 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 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.

An initial estimate for the relative orientation can be obtained simply by performing a standard camera calibration. 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 [5].

Theorem 1 (Relative Orientation) *Suppose $\{v^a_t\}^N_{t=1}$ and $\{v^b_t\}^N_{t=1}$ are measurements satisfying $v^a_t = q^{ab} \odot v^b_t \odot q^{ba}$. Then the sum of the squared residuals,*

$$V(q^{ab}) = \sum_{t=1}^{N} \|e_t\|^2 = \sum_{t=1}^{N} \|v^a_t - q^{ab} \odot v^b_t \odot q^{ba}\|^2, \tag{12}$$

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

$$A = -\sum_{t=1}^{N} (v^a_t)_L (v^b_t)_R. \tag{13}$$

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

$$q_L \triangleq \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{14a}$$

$$q_R \triangleq \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \tag{14b}$$

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 $\{u_{a,t}\}^M_{t=1}$, $\{u_{\omega,t}\}^M_{t=1}$ as well as images $\{I_t\}^N_{t=1}$.
   - Rotate around all 3 axes, with sufficiently exiting angular velocities.
   - Always keep the calibration pattern in view.
3. Obtain the point correspondences between the 2D feature locations $p^i_{t,k}$ and the corresponding 3D grid coordinates $p^e_{t,k}$ of the calibration pattern for all images $\{I_t\}^N_{t=1}$.
4. Solve the gray-box identification problem (4), starting the optimization from $\theta_0 = ((\hat{\varphi}^{\mathrm{cb}}_0)^T, 0, 0, 0, (g^e_0)^T)^T$. Here, $g^e_0 = (0, 0, -g)^T$ since the calibration pattern is placed horizontally and $\hat{\varphi}^{cb}_0$ can be obtained using Algorithm 2.

This theorem is used in Algorithm 2 to obtain an initial orientation estimate. Note that $g^e = \begin{pmatrix} 0 & 0 & -g \end{pmatrix}^T$, 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 $\{I_t\}^N_{t=1}$ of the pattern while holding the sensor unit static in various poses, simultaneously acquiring accelerometer readings $\{u_{\mathrm{a},t}\}^N_{t=1}$.
3. Perform a camera calibration using the images $\{I_t\}^N_{t=1}$ to obtain the orientations $\{q^{\mathrm{ce}}_t\}^N_{t=1}$.
4. Compute an estimate $\hat{q}^{\mathrm{cb}}$ from the vectors $g^{\mathrm{c}}_t = R^{\mathrm{ce}}_t g^e$ and $g^{\mathrm{b}}_t = -u_{\mathrm{a},t}$ using Theorem 1.

5 Experiments

Algorithm 1 has been used to calibrate the sensor unit introduced in Section 1. This algorithm computes estimates of the relative position and orientation of the IMU and the camera, i.e., c^b and φ^{cb}, based on the motion of the sensor unit. The setup employed is identical to that of a typical camera calibration setup. A number

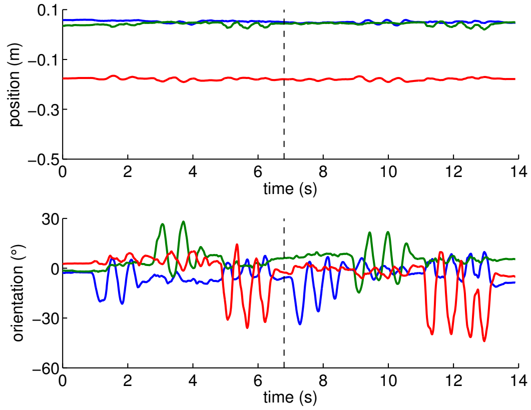


Figure 3: A trajectory of the sensor unit used for calibration. It contains both estimation data ($t < 6.8$ s) and validation data ($t \geq 6.8$ s), as indicated by the dashed line.

of experiments have been performed. During such an experiment the sensor unit has been rotated around its three axis, see Figure 3 for an illustration. The data is split into two parts, one estimation part and one validation part, see Figure 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 [8].

In Table 1 the estimates produced by Algorithm 1 are given together with confidence intervals (99%). Reference values are also given, these are taken as the result of Algorithm 2 on a separate data set (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 available and some shifts can occur in the tangential directions. Table 1 indicates that the estimates are indeed rather good, but that their accuracy is a little underestimated.

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

Orientation	$\hat{\varphi}^{\text{cb}}_x$ (°)	$\hat{\varphi}^{\text{cb}}_y$ (°)	$\hat{\varphi}^{\text{cb}}_z$ (°)
Trial 1	-0.16 [-0.30, -0.03]	0.50 [0.39, 0.61]	-0.30 [-0.51, -0.09]
Trial 2	-0.89 [-1.00, -0.77]	0.01 [-0.07, 0.09]	-0.90 [-1.04, -0.77]
Reference[a]	-0.55	0.01	-0.63
Position	$\hat{c}^{\text{b}}_x$ (mm)	$\hat{c}^{\text{b}}_y$ (mm)	$\hat{c}^{\text{b}}_z$ (mm)
Trial 1	-16.0 [-16.3, -15.7]	-4.7 [-5.0, -4.4]	38.2 [37.8, 38.6]
Trial 2	-18.1 [-18.3, -17.9]	-6.2 [-6.3, -6.1]	37.6 [37.3, 37.9]
Reference[b]	-14.5	-6.5	-

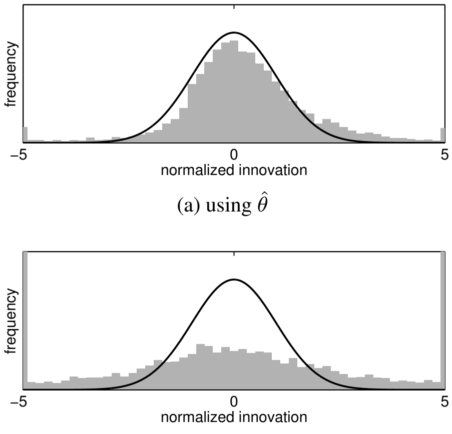
[a] using Algorithm 2 on a large data set.

[b] using the CCD position of the technical drawing.

In order to further validate the estimates the normalized innovations are studied. Histograms of the normalized innovations are given in Figure 4. This figure is generated using the validation data. In Figure 4b the effect of using the wrong relative translation and orientation and sensor biases is shown. From Figure 4a it is clear that the normalized innovations are close to white noise, but have heavy tails. This implies that the model with the estimated parameters and its assumptions appears to be reasonable, 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.

6 Conclusion

The experiments indicate that the proposed algorithm is an easy-to-use calibration method to determine the relative position and orientation of an IMU and a spherical camera, that are rigidly connected. This solves an important issue preventing successful integration of vision and inertial sensors in AR applications. Even small displacements and misalignments can be accurately calibrated from short measurement sequences made using the standard camera calibration setup.



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

(b) using θ_0

Figure 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.

References

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

[2] P. Corke, J. Lobo, and J. Dias. An introduction to inertial and visual sensing. *Int. J. Rob. Res.*, 26(6):519–535, 2007.

[3] S. Graebe. *Theory and Implementation of Gray Box Identification*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, June 1990.

[4] J. D. Hol. *Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors*. Lic. thesis no 1379, Dept. Electr. Eng, Linköpings universitet, Sweden, May 2008.

[5] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, Apr. 1987.

[6] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Inc, 2000.

[7] J. Kannala and S. S. Brandt. Generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(8): 1335–1340, aug 2006.

[8] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Inc, Upper Saddle River, NJ, USA, 2nd edition, 1999.

[9] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *Int. J. Rob. Res.*, 26(6):561–575, 2007.

[10] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer-Verlag, New York, 2006.

[11] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Systems*, pages 5695–5701, Beijing, China, Oct. 2006.

[12] Xsens Motion Technologies, 2008. URL `http://www.xsens.com/`. Accessed April 2nd, 2008.