Jorge Lobo*
Jorge Dias
Institute of Systems and Robotics – University of Coimbra, Portugal
{jlobo,jorge}@isr.uc.pt

Relative Pose Calibration Between Visual and Inertial Sensors

Abstract

This paper proposes an approach to calibrate off-the-shelf cameras and inertial sensors to have a useful integrated system to be used in static and dynamic situations. When both sensors are integrated in a system their relative pose needs to be determined. The rotation between the camera and the inertial sensor can be estimated, concurrently with camera calibration, by having both sensors observe the vertical direction in several poses. The camera relies on a vertical chequered planar target and the inertial sensor on gravity to obtain a vertical reference. Depending on the setup and system motion, the translation between the two sensors can also be important. Using a simple passive turntable and static images, the translation can be estimated. The system needs to be placed in several poses and adjusted to turn about the inertial sensor centre, so that the lever arm to the camera can be determined. Simulation and real data results are presented to show the validity and simple requirements of the proposed methods.

KEY WORDS—computer vision, inertial sensors, sensor fusion, calibration.

1. Introduction

Inertial sensors coupled to cameras can provide valuable data about camera ego-motion and how world features are expected to be oriented. Object recognition and tracking benefits from both static and inertial information. Several human vision tasks rely on the inertial data provided by the vestibular. Artificial system should also exploit this sensor fusion (Corke et al., 2007).

In our previous work we explored some of the benefits of combining the two sensing modalities, and how gravity can be used as a vertical reference (Lobo and Dias 2003, 2004). We now focus on how the two sensors can be cross-calibrated so that they can be used in static and dynamic situations.

The rotation between the camera and the inertial measurement unit (IMU) can be estimated by having both sensors observe the vertical direction, using a vertical visual target for the camera, and gravity for the inertial sensors. Standard camera calibration can be performed on the same set of images, both using the same visual target, such as a vertical chequered target, simplifying the whole calibration procedure.

The problem of estimating the rotation between the inertial sensor and the camera is a particular case of the well-known orthogonal Procrustes method for 3D attitude estimation (Dorst 2005). Instead of having two sets of points we have two sets of unit vectors corresponding to the observed vertical in each sensor at several poses. In our work we used the unit quaternion derivation of the method (Horn 1987).

The translation between the two will not be important in some applications, but if the inertial sensor is attached to the camera system with a significant lever arm, it will have to be taken into account for fast motions.

Using a simple passive turntable and static images the translation can be estimated. The system needs to be placed in several poses and adjusted to turn about the inertial sensor centre. The correct positioning is obtained when the accelerometers are not subject to rotation induced centripetal acceleration. The lever arm can than be estimated from static images of a suitably placed visual target before and after each rotation.

The translation estimation is a particular case of the standard hand–eye calibration (Tsai and Lenz 1989; Daniilidis 1999). However, since the target is being repositioned after each turn, the method is not applied to the full data set as in traditional hand–eye calibration.

In the following sections scalars are represented by simple italic font (a, b, α), vectors by boldface non-italic roman

The International Journal of Robotics Research
Vol. 26, No. 6, June 2007, pp. 561–575
DOI: 10.1177/0278364907079276

Figures 8, 11–17, 21, 23–27 appear in color online: http://ijr.sagepub.com

* Corresponding author

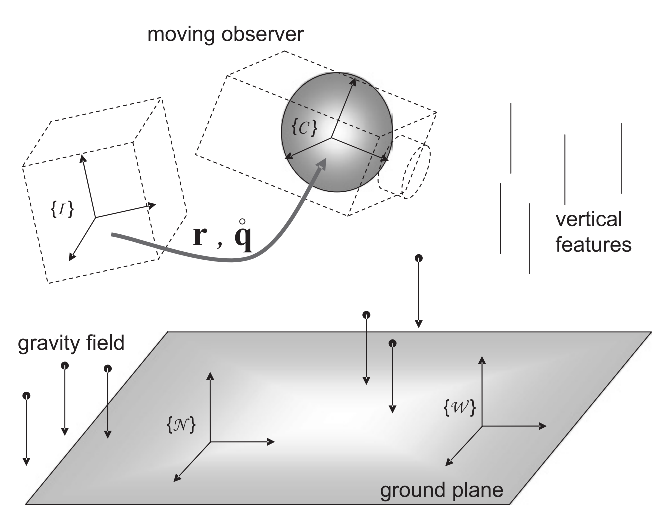


Fig. 1. Camera $\{\mathcal{C}\}$, IMU $\{\mathcal{I}\}$, world aligned mobile system $\{\mathcal{N}\}$, and world fixed $\{\mathcal{W}\}$ frames of reference

font ($\mathbf{v}$, $\mathbf{p}$, $\boldsymbol{\omega}$), quaternions are similar but with a small circle on top ($\mathring{\mathbf{q}}$, $\mathring{\mathbf{p}}$), and matrices by boldface roman capitals ($\mathbf{A}$, $\mathbf{M}$). Frames of reference use capital calligraphy font ($\{\mathcal{I}\}$, $\{\mathcal{C}\}$), and a superscript preceding a vector indicates the frame of reference (${}^{\mathcal{I}}\mathbf{v}$, ${}^{\mathcal{C}}\boldsymbol{\omega}$). Vector dot product is represented as $\mathbf{v} \cdot \mathbf{p}$, and cross product as $\mathbf{v} \times \mathbf{p}$. The quaternion conjugate is denoted by $\mathring{\mathbf{q}}^*$ and $\mathring{\mathbf{q}}\mathbf{v}\mathring{\mathbf{q}}^*$ is the rotation of vector $\mathbf{v}$ by $\mathring{\mathbf{q}}$.

2. Camera and IMU Data Relationship

2.1. Frames of Reference

When combining the two sensing modalities, the frame of reference in which sensor measurements are made need to be taken into account. The sensor observed features, visual or inertial, also have implicit or explicit frames of reference to be considered. Figure 1 shows the several frames of reference that can be defined. Considering a moving observer with a visual sensor and inertial sensors rigidly mounted, we have the camera $\{\mathcal{C}\}$, IMU $\{\mathcal{I}\}$, world aligned mobile system $\{\mathcal{N}\}$, and world fixed $\{\mathcal{W}\}$ frames of reference. The gravity field directly sensed by the inertial sensors, and indirectly from visual vertical features by the camera, provide some external references that help in obtaining a world aligned moving frame of reference, or navigation frame $\{\mathcal{N}\}$, and after motion compensation the world fixed $\{\mathcal{W}\}$ frame of reference.

2.2. Inertial Data in Camera Frame of Reference

The visual processing has to consider the motion parameters of the camera centre of projection. Since the inertial measurements performed by the inertial sensors are given in the IMU frame of reference $\{\mathcal{I}\}$ and not in the camera frame of reference $\{\mathcal{C}\}$, the rigid body transformation between the two has to be taken into account. This transformation can be expressed by the translation vector $\mathbf{r}$ from $\{\mathcal{I}\}$ to $\{\mathcal{C}\}$, and the unit quaternion $\mathring{\mathbf{q}}$ that rotates inertial measurements in the inertial sensor frame of reference $\{\mathcal{I}\}$ to the camera frame of reference $\{\mathcal{C}\}$. In the following sections the inertial sensed measurements are expressed in the camera frame of reference.

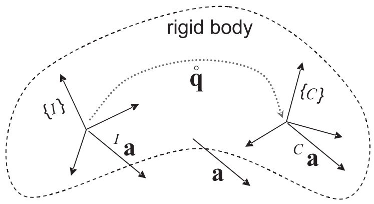


Fig. 2. Inertial sensed acceleration in non-rotating camera frame of reference.

2.2.1. Non-rotating Camera Linear Acceleration

If a rigid body has no angular velocity, any point within will have the same linear acceleration. As shown in Figure 2, to report the inertial sensed acceleration to the camera centre of projection, i.e. to have ${}^{\mathcal{C}}\mathbf{a}$, we just apply the known rotation between the two frames of reference, $\mathring{\mathbf{q}}$, i.e.

$$ {}^{\mathcal{C}}\mathbf{a} = \mathring{\mathbf{q}}\, {}^{\mathcal{I}}\mathbf{a}\, \mathring{\mathbf{q}}^* \tag{1} $$

where ${}^{\mathcal{I}}\mathbf{a}$ is the sensed acceleration in the IMU frame of reference.

2.2.2. Rotating Camera Angular Velocity

Any point of a rigid rotating body has the same angular velocity. As shown in Figure 3, to obtain the camera angular velocity in the camera frame of reference, ${}^{\mathcal{C}}\boldsymbol{\omega}$, we again just apply the known rotation between the two frames of reference:

$$ {}^{\mathcal{C}}\boldsymbol{\omega} = \mathring{\mathbf{q}}\, {}^{\mathcal{I}}\boldsymbol{\omega}\, \mathring{\mathbf{q}}^* \tag{2} $$

where ${}^{\mathcal{I}}\boldsymbol{\omega}$ is the sensed angular velocity in the IMU frame of reference.

However, the above formulation does not take into account the centre of rotation. In Figure 3 the centre of rotation is shown to be within the rigid body, but it could be anywhere. We can always model the rigid body motion as rotating about its centre of mass and experiencing centripetal acceleration with respect to the true centre of rotation. As we will see below, this adds some complexity when reporting inertial measurements from one frame of reference to another.

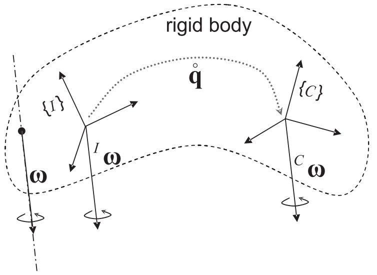


Fig. 3. Inertial sensed angular velocity in camera frame of reference.

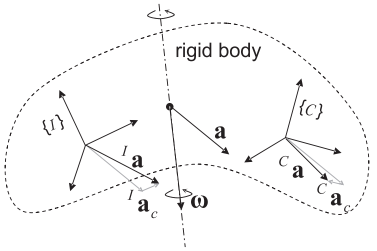


Fig. 4. Inertial sensed acceleration in rotating camera frame of reference.

2.2.3. Rotating Camera Linear Acceleration

If a rigid body has no angular velocity, any point within will have the same linear acceleration. But if the rigid body is rotating about some axis, a centripetal acceleration, proportional to the perpendicular distance to the rotation axis, will be added. As shown in Figure 4, the linear acceleration of both camera and IMU will have a component due to the rotation about some axis, so when reporting inertial sensor observations to the camera frame of reference they must be taken into account, i.e.

$$^{\mathcal{C}}\mathbf{a} = \mathring{\mathbf{q}}(^{\mathcal{I}}\mathbf{a} - {}^{\mathcal{I}}\mathbf{a}_c)\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\mathbf{a}_c \tag{3}$$

where $^{\mathcal{I}}\mathbf{a}_c$ is the IMU centripetal acceleration, and $^{\mathcal{C}}\mathbf{a}_c$ the camera centripetal acceleration, both relative to some rotation axis.

The rotation axis must be fixed relative to an inertial frame of reference, i.e. a non-accelerating non-rotating frame of reference. In other words, the inertial sensor measures the centripetal acceleration to the true rotation axis, and not relative to say the system centre of mass, which would not be fixed relative to an inertial frame of reference.

In general, centripetal acceleration $\mathbf{a}_c$ at a point $\mathbf{r}$ with the origin on the rotation axis is given by

$$\mathbf{a}_c = \boldsymbol{\omega} \times \mathbf{v}_t = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \tag{4}$$

where $\boldsymbol{\omega}$ is the angular velocity and $\mathbf{v}_t$ is the tangential velocity.

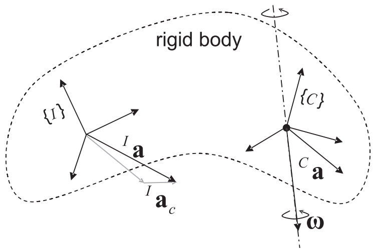


Fig. 5. Inertial sensed acceleration in camera frame of reference, with rotation about the camera.

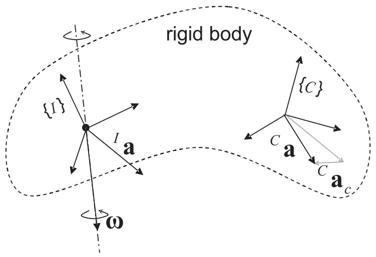


Fig. 6. Inertial sensed acceleration in camera frame of reference, with rotation about the IMU.

If we assume that the rotation axis goes through the camera centre of projection, as shown in Figure 5, then it will not have centripetal acceleration and its linear acceleration is given by

$$\begin{aligned} ^{\mathcal{C}}\mathbf{a} &= \mathring{\mathbf{q}}(^{\mathcal{I}}\mathbf{a} - {}^{\mathcal{I}}\mathbf{a}_c)\mathring{\mathbf{q}}^* \\ &= \mathring{\mathbf{q}}(^{\mathcal{I}}\mathbf{a} - {}^{\mathcal{I}}\boldsymbol{\omega} \times (^{\mathcal{I}}\boldsymbol{\omega} \times {}^{\mathcal{I}}\mathbf{r}))\mathring{\mathbf{q}}^* \\ &= \mathring{\mathbf{q}}\;^{\mathcal{I}}\mathbf{a}\;\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\boldsymbol{\omega} \times (^{\mathcal{C}}\boldsymbol{\omega} \times {}^{\mathcal{C}}\mathbf{r}) \end{aligned} \tag{5}$$

where $^{\mathcal{I}}\mathbf{r}$ is the translation from the IMU to the camera in the IMU frame of reference, $^{\mathcal{C}}\mathbf{r}$ is the translation from the camera to the IMU in the camera frame of reference, and $\mathring{\mathbf{q}}\;^{\mathcal{I}}\mathbf{r}\;\mathring{\mathbf{q}}^* = -{}^{\mathcal{C}}\mathbf{r}$.

If we assume that the rotation axis goes though the IMU centre, as shown in Figure 6, than no centripetal acceleration will be sensed, and the camera linear acceleration is given by

$$\begin{aligned} ^{\mathcal{C}}\mathbf{a} &= \mathring{\mathbf{q}}(^{\mathcal{I}}\mathbf{a},)\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\mathbf{a}_c \\ &= \mathring{\mathbf{q}}\;^{\mathcal{I}}\mathbf{a}\;\mathring{\mathbf{q}}^* + {}^{\mathcal{C}}\boldsymbol{\omega} \times (^{\mathcal{C}}\boldsymbol{\omega} \times (-{}^{\mathcal{C}}\mathbf{r})) \\ &= \mathring{\mathbf{q}}\;^{\mathcal{I}}\mathbf{a}\;\mathring{\mathbf{q}}^* - {}^{\mathcal{C}}\boldsymbol{\omega} \times (^{\mathcal{C}}\boldsymbol{\omega} \times {}^{\mathcal{C}}\mathbf{r}). \end{aligned} \tag{6}$$

From the above derivation, the knowledge of the rotation axis is crucial to describe the absolute motion of all the points within a rigid body, i.e. the motion relative to an inertial frame.

Inertial navigation systems rely on the path taken from a known initial position to report current attitude and position

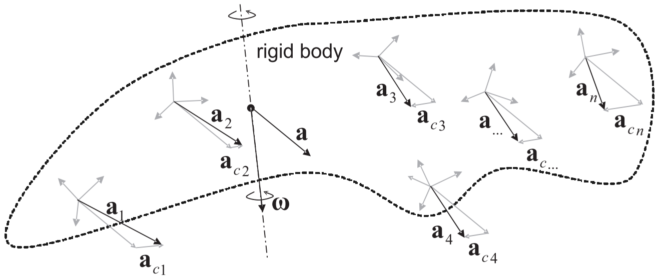


Fig. 7. Rotating rigid body with distributed tri-axial accelerometers to find rotation axis.

in the frame of reference of the initial known point. In other words we might know how to describe the motion of a point, and hence the whole rigid body, by integrating the measured acceleration at a given point, linear plus centripetal, with the appropriate rotation update from the gyros. But if the initial position is not known, we are not able to determine the rotation axis, and correctly report centripetal acceleration to the camera.

Consider a rotating rigid body with distributed tri-axial accelerometers as shown in Figure 7. Each sensor will measure a different resultant acceleration determined by its relative position to the axis of rotation.

If the rigid body has a pure rotation, i.e. $\mathbf{a} = 0$, than each sensor will only measure the centripetal acceleration $\mathbf{a}_i = \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_i)$.

As we will see in the following sections, gravity can be used as a common reference to calibrate relative rotation between sensors. In this case taking sets of measurements over several static poses, the relative rotation between the several tri-axial accelerometers can be determined. A single triad of gyros can be used to measure angular velocity, and the estimated frame relative rotations used to obtain each $\boldsymbol{\omega}_i$.

3. Stand-alone Sensor Calibration

Before we consider the cross-calibration of cameras and inertial sensors, we will take a look at how each sensor can be calibrated individually.

3.1. Camera Calibration

Camera calibration has been extensively studied, and standard techniques established. For this work camera calibration was performed using Bouguet's camera calibration toolbox (Bouguet 2006).

The calibration uses images of a chequered target in several positions and recovers the camera's intrinsic parameters, as well as the target positions relative to the camera, as shown further ahead in Figure 15.

The calibration algorithm is based on Zhang's work in estimation of planar homographies for camera calibration (Zhang 1999), but the closed-form estimation of the internal parameters from the homographies is slightly different, since the orthogonality of vanishing points is explicitly used and the distortion coefficients are not estimated at the initialization phase.

The calibration toolbox will also be used to recover camera extrinsic parameters, from the reconstructed target positions, in the subsequent relative pose calibration.

3.2. Inertial Sensor Calibration

Inertial navigation systems also have established calibration techniques, but rely on high-end sensors and actuators. When considering a complete inertial navigation system, initial calibration and alignment are more elaborate (Nebot and Durrant-Whyte 1997). Nevertheless, in order to use off-the-shelf inertial sensors attached to a camera, appropriate modelling and calibration techniques are required.

Inertial sensors measure linear acceleration and angular velocity. An inertial measurement unit (IMU) has three orthogonal accelerometers and three orthogonal rate gyros.

To estimate velocity and position, integration over time has to be performed, leading to unbounded error. The gyros keep track of rotations, so that linear velocity and position are computed in the correct frame of reference. Appropriate calibration has to be performed to minimise the error buildup.

Assuming a simple linear model for the inertial sensors, scale factor, bias and axis-alignment need to be known to correctly use the inertial sensor measurements. Equation (7), represents this simple model for each set of three non-coplanar accelerometers or rate gyros.

$$\begin{aligned} \mathbf{z}_o &= \mathbf{M} \cdot \mathbf{z}_i + \mathbf{b} \\ &= \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} \cdot \begin{bmatrix} z_{ix} \\ z_{iy} \\ z_{iz} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}. \end{aligned} \tag{7}$$

The true values of quantities to be measured are represented by $\mathbf{z}_i$, while $\mathbf{z}_o$ represents the actual output from the sensors. Vector $\mathbf{b}$ represents the bias for each individual sensor, while s_{kk} is the sensitivity (or scale factor) for the sensor oriented along axis k, and s_{kl} the cross sensitivity, resulting from axis misalignments, relating axis k and l. For low cost inertial sensors these parameters are not always provided by the manufacturer, and when using discrete components their alignment has to be estimated.

Some of the inertial sensors parameters can be determined by performing simple operations and measuring the sensor outputs.

In Vieville and Faugeras (1989), where the use of an inertial system in a robotic system is analyzed, a set of calibration procedures is presented for accelerometers and gyros. In this seminal work, that sets as a future objective the study of the cooperation between vision and inertial sensing, the data provided by the inertial sensors in studied within the context of mobile robotic applications.

Using gravity as a reference, horizontal aligned accelerometers should have zero output, and vertical ones a full $\pm 1g$. Placing the IMU in particular directions with respect to gravity, sufficient data can be collected to calibrate, without any special hardware. This static calibration only requires the ability to orient the accelerometers in particular directions with respect to gravity, and to maintain the system without any movement during the static measurements.

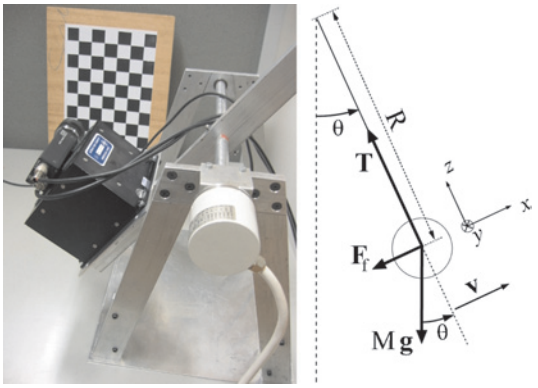
For gyros no such reference is available, there is the earth rotation induced Coriolis force, but it is too small to be measurable by the low-cost rate gyros considered. However, the sensor bias or offset can be determined by measuring the output of a static gyro sensor.

To estimate all the parameters, a dynamic calibration is required. However, if a controlled turn rate device is not available, performing a rotation in the vertical plane enables the use gravity as a reference. Using a mechanical axis of rotation, that can be oriented in any direction, the vertical reference will provide the calibration. See Vieville and Faugeras (1989) for the mathematical derivation of this calibration procedure for inertial sensors.

Performing a rotation in the vertical plane also solves the problem of determining the alignment between accelerometers and gyros, by relating gyros sensing axis with accelerometer alignment. Having a fixed horizontal rotating axis, continuous rotation provides the gyros sensing axis, and stops along the way provide the relative pose of this sensing axis with the accelerometers. The relative pose between the accelerometers and gyros obtained in this way is important so that the methods presented in this work are not limited to the relative pose between the camera and the accelerometers.

In Alves et al. (2003) we assumed a linear sensor model and used a pendulum instrumented with an encoded shaft to estimate the alignment, bias and scale factor of inertial measurements. Observing the sensor outputs the response was practically linear, indicating the suitability of a linear model. Due to inertial sensor temperature sensitivity, temperature variations might introduce non-linear changes in the linear model parameters. Building a lookup table for several working temperatures, and interpolating the parameters for a given temperature can overcome the problem.

The pendulum was chosen since it is relatively straightforward to determine it's motion and acting forces. It is instrumented with a high-resolution absolute encoder attached to its axis, so that the angular position of the pendulum is known and consequently, the pose of the IMU. Figure 8 shows the pendulum setup and a diagram with the forces acting on the moving pendulum.



(a) (b)

Fig. 8. (a) Pendulum used to calibrate the inertial sensors. (b) Forces acting on a moving pendulum.

By attaching the IMU to the pendulum in three different orthogonal orientations, sufficient data can be collected to calibrate the three accelerometers and the three rate gyros. The procedure consists in determining the parameters in **M** and **b** of the sensor model described in (7). In Alves et al. (2003) further details and results are presented.

4. Camera and IMU Relative Pose Calibration

In the following sections we will present our method for calibration of rotation and translation between the camera and the inertial sensors. Using the gravity reference a static boresight (visual alignment) approach is proposed, requiring a simple setup and avoiding fast blurred images and controlled active rate generators to perform known motion. This only relates the accelerometers' with the cameras, the relative pose between the accelerometers and gyros can be done independently, using horizontal and vertical rotation axis as suggested above.

4.1. Calibration of Rotation between IMU and Camera

In order to determine the rigid rotation between the INS frame of reference $\{\mathcal{I}\}$ and the camera frame of reference $\{\mathcal{C}\}$, both sensors are used to measure the vertical direction, as shown in Figure 9. When the IMU sensed acceleration is equal in magnitude to gravity, the sensed direction is the vertical. For the camera, using a specific calibration target such as a chequered rectangular planar target placed vertically, the vertical direction can be taken from the corresponding vanishing point of the vertical lines joining the corners of the chequered squares.

This boresight static approach can be easily performed, not requiring any additional equipment, apart from the chequered target, obtained using a standard printer, already used for camera calibration.

If n observations are made for distinct camera positions, recording the vertical reference provided by the inertial sensors

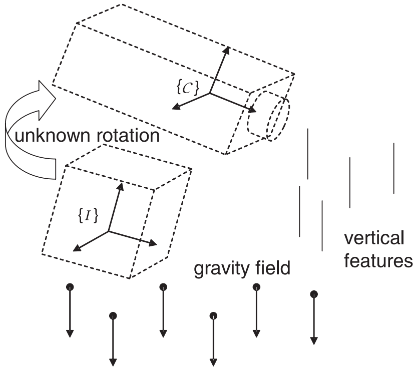


Fig. 9. IMU and camera observing gravity.

and the vanishing point of scene vertical features, the absolute orientation can be determined using the orthogonal Procrustes method for 3D attitude estimation. We will use Horn's closed-form solution for absolute orientation using unit quaternions (Horn 1987), applied here only to unit vectors. Since we are only observing a 3D direction in space, we can only determine the rotation between the two frames of reference.

Let ${}^{\mathcal{I}}\mathbf{v}_i$ be a measurement of the vertical by the inertial sensors, and ${}^{\mathcal{C}}\mathbf{v}_i$ the corresponding measurement made by the camera derived from some scene vanishing point. We want to determine the unit quaternion $\mathring{\mathbf{q}}$ that rotates inertial measurements in the inertial sensor frame of reference $\{\mathcal{I}\}$ to the camera frame of reference $\{\mathcal{C}\}$. We want to find the unit quaternion $\mathring{\mathbf{q}}$ that maximises

$$\sum_{i=1}^{n} (\mathring{\mathbf{q}}\, {}^{\mathcal{I}}\mathbf{v}_i\, \mathring{\mathbf{q}}^*) \cdot {}^{\mathcal{C}}\mathbf{v}_i \tag{8}$$

which can be rewritten as

$$\sum_{i=1}^{n} (\mathring{\mathbf{q}}\, {}^{\mathcal{I}}\mathbf{v}_i) \cdot ({}^{\mathcal{C}}\mathbf{v}_i\, \mathring{\mathbf{q}}). \tag{9}$$

The quaternion product can be expressed as a matrix. Using ${}^{\mathcal{I}}\mathbf{v}_i = ({}^{\mathcal{I}}x_i, {}^{\mathcal{I}}y_i, {}^{\mathcal{I}}z_i)^T$ and ${}^{\mathcal{C}}\mathbf{v}_i = ({}^{\mathcal{C}}x_i, {}^{\mathcal{C}}y_i, {}^{\mathcal{C}}z_i)^T$ we define

$$\mathring{\mathbf{q}}\, {}^{\mathcal{I}}\mathbf{v}_i = \begin{bmatrix} 0 & -{}^{\mathcal{I}}x_i & -{}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}z_i \\ {}^{\mathcal{I}}x_i & 0 & {}^{\mathcal{I}}z_i & -{}^{\mathcal{I}}y_i \\ {}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}z_i & 0 & {}^{\mathcal{I}}x_i \\ {}^{\mathcal{I}}z_i & {}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}x_i & 0 \end{bmatrix} \mathring{\mathbf{q}} = {}^{\mathcal{I}}\mathbf{V}_i \mathring{\mathbf{q}} \tag{10}$$

and

$${}^{\mathcal{C}}\mathbf{v}_i \mathring{\mathbf{q}} = \begin{bmatrix} 0 & -{}^{\mathcal{C}}x_i & -{}^{\mathcal{C}}y_i & -{}^{\mathcal{C}}z_i \\ {}^{\mathcal{C}}x_i & 0 & -{}^{\mathcal{C}}z_i & {}^{\mathcal{C}}y_i \\ {}^{\mathcal{C}}y_i & {}^{\mathcal{C}}z_i & 0 & -{}^{\mathcal{C}}x_i \\ {}^{\mathcal{C}}z_i & -{}^{\mathcal{C}}y_i & {}^{\mathcal{C}}x_i & 0 \end{bmatrix} \mathring{\mathbf{q}} = {}^{\mathcal{C}}\mathbf{V}_i \mathring{\mathbf{q}}. \tag{11}$$

Substituting in (9)

$$\sum_{i=1}^{n} ({}^{\mathcal{I}}\mathbf{V}_i\, \mathring{\mathbf{q}}) \cdot ({}^{\mathcal{C}}\mathbf{V}_i\, \mathring{\mathbf{q}}) \tag{12}$$

or

$$\sum_{i=1}^{n} \mathring{\mathbf{q}}^T\, {}^{\mathcal{I}}\mathbf{V}_i^T\, {}^{\mathcal{C}}\mathbf{V}_i\, \mathring{\mathbf{q}} \tag{13}$$

factoring out $\mathring{\mathbf{q}}$ we get

$$\mathring{\mathbf{q}}^T \left(\sum_{i=1}^{n} {}^{\mathcal{I}}\mathbf{V}_i^T\, {}^{\mathcal{C}}\mathbf{V}_i \right) \mathring{\mathbf{q}}. \tag{14}$$

So we want to find $\mathring{\mathbf{q}}$ such that

$$\max \mathring{\mathbf{q}}^T\, \mathbf{N}\, \mathring{\mathbf{q}} \tag{15}$$

where

$$\mathbf{N} = \sum_{i=1}^{n} {}^{\mathcal{I}}\mathbf{V}_i^T\, {}^{\mathcal{C}}\mathbf{V}_i. \tag{16}$$

Having

$$S_{xx} = \sum_{i=1}^{n} {}^{\mathcal{I}}x_i\, {}^{\mathcal{C}}x_i, \quad S_{xy} = \sum_{i=1}^{n} {}^{\mathcal{I}}x_i\, {}^{\mathcal{C}}y_i \tag{17}$$

and analogously for all nine pairings of the components of the two vectors, matrix $\mathbf{N}$ can be expressed using these sums as

$$\mathbf{N} = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix}. \tag{18}$$

The sums contain all the information that is required to find the solution. Since $\mathbf{N}$ is a symmetric matrix, the solution to this problem is the four-vector $\mathbf{q}_{max}$ corresponding to the largest eigenvalue λ_{max} of $\mathbf{N}$ - see Horn (1987) for details.

4.1.1. Measurement Span for Rotation Estimation

The above method finds the rotation that maximises the alignment of the rotated inertial frame verticals with the camera observed verticals expressed by (8).

The inertial frame verticals, ${}^{\mathcal{I}}\mathbf{v}_i$, are easily obtained from the IMU accelerometers. The only restriction is that the system has to be motionless, or subject to constant speed, so that gravity can be used as a vertical reference.

The camera frame verticals, ${}^{\mathcal{C}}\mathbf{v}_i$, are not so easily obtained. Some scene element must be known to have vertical features, so that the vertical vanishing point can be determined. In our experimental work we relied on the same chequered target used for calibrating the camera, but now placing it vertically. For the n observations, the target does not have to remain in the same position, but must be vertical.

The camera calibration toolbox used (Bouguet 2006) provides the recovered extrinsic parameters, that result from the minimisation of the reprojection error (through gradient descent). The columns of the camera to target rotation matrix provide the x,y and z (out of plane) axis of the planar target in the camera frame of reference. Depending on the order of selection of target points during calibration, ${}^{\mathcal{C}}\mathbf{v}_i$ will either be x or y, and will correspond to the vanishing point of the vertical lines joining the corners of the chequered squares in the planar target.

A single pair of measurements, i.e. $n = 1$, provides a valid rotation for the given observation, but prone to degenerate cases, depending on the system pose and rotation between frames. Using more observations at distinct system poses avoids this, and improves the estimate by reducing estimation error, assuming that the measurements have zero mean Gaussian noise. The camera poses used need not span the entire 3D attitude space, a few poses with the system at different rotations relative to the inertial vertical are sufficient to avoid ill conditioned cases.

4.1.2. Rotation Calibration Summary

Figure 10 provides a summary of the required steps to perform calibration of rotation between camera and IMU using the proposed algorithm.

Rotation Calibration Summary

- N static observations at distinct positions:
  - vertical chequered target;
  - save image and corresponding inertial data.
- Perform standard camera calibration for image set.
- Compute rotation quaternion:
  - Use target vertical vanishing points ${}^{\mathcal{C}}\mathbf{v}_i$ detected for camera calibration;
  - Inertial data from accelerometers provide ${}^{\mathcal{I}}\mathbf{v}_i$;
  - Rotation quaternion $\mathring{\mathbf{q}}$ given by (15).

Fig. 10. Summary of required steps to perform calibration of rotation between camera and IMU using the proposed algorithm.

4.1.3. Error Sensitivity and Simulation Results

In order to validate the proposed method and perform noise sensitivity tests, simulations where performed under varying conditions.

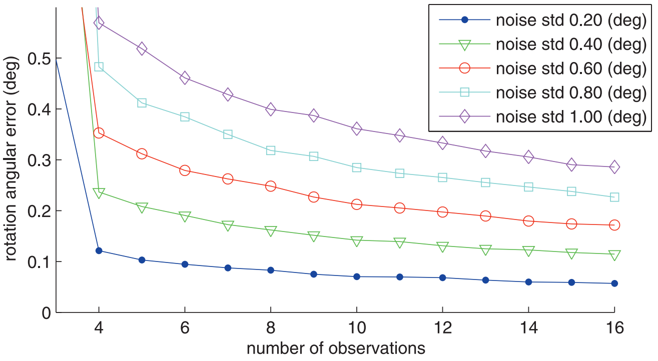


Fig. 11. Simulation rotation estimation mean error for increasing number of observations.

For each simulation run a random rotation $\mathring{\mathbf{q}}$ is applied to a random set of simulated inertial observed verticals, ${}^{\mathcal{I}}\mathbf{v}_i$, to obtain a corresponding set of camera observed verticals, ${}^{\mathcal{C}}\mathbf{v}_i$. These simulated camera observations are corrupted by applying a random rotation with a normal distributed magnitude (with zero mean and set standard deviation) about a random axis, i.e. a uniformly distributed 3D axis. The rotation quaternion that relates the two sets is estimated as $\hat{\mathring{\mathbf{q}}}$ by the above method. The error in the estimation can be measured by considering the rotation required to correct the estimate to the true value, $\mathring{\mathbf{q}} = \mathring{\mathbf{e}} * \hat{\mathring{\mathbf{q}}}$. With $\theta_e = 2\cos^{-1}(e_s)$, where e_s is the scalar component of $\mathring{\mathbf{e}}$, we take $\delta_\theta = |\theta_e|$ as the error measure.

Figure 11 shows simulation results of several takes with different numbers of observations used. The increasing error lines correspond to increasing rotation error added to the simulated observed camera verticals. For each setting the method runs 1000 times and the mean error is evaluated.

The above simulation was performed with a random set of simulated camera observed verticals, uniformly distributed on the unit sphere. To better evaluate the method, simulations were performed with restricted sets of simulated observations.

Figure 12 shows simulation results with simulated camera observed verticals, restricted to a 20° patch of the unit sphere. The geometric dilution of precision from such a narrow field of observation leads to poorer results, but since the added noise has a normal distribution with a maximum standard deviation of 1° a good estimate of the rotation is still obtained.

To approach a degenerate case of having all observation in the same plane, another simulation was performed with cam-

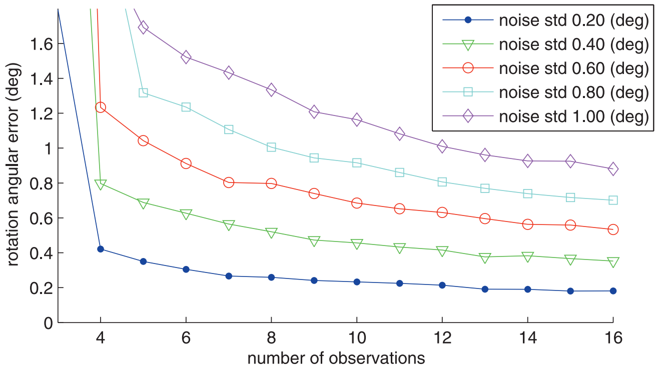


Fig. 12. Simulation rotation estimation mean error for increasing number of observations, with simulated camera observed verticals, restricted to a 20° patch of the unit sphere.

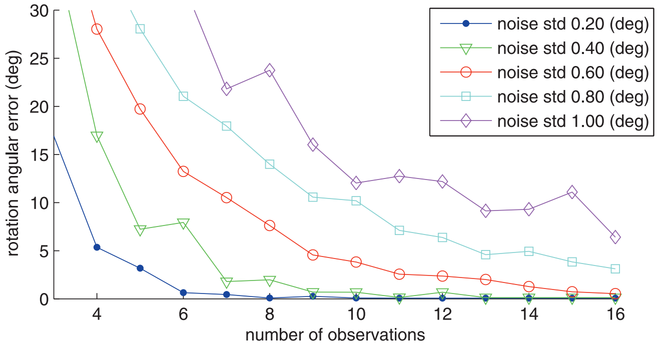


Fig. 13. Simulation rotation estimation mean error for increasing number of observations, with simulated camera observed verticals, restricted to a patch of the unit sphere corresponding to a full great circle band with 1° width.

era observed verticals restricted to a patch of the unit sphere corresponding to a full great circle band with 1° width. Figure 13 shows the results. Since the noise level (±std) is above the out of plane distribution of the observation, the degenerate single plane observations dominate and lead to the high error in the estimated rotation.

In all the simulations there is a fast decay of the error as the number of observations grow, until the error is at the same level of the input Gaussian noise. Beyond this point the decay is must smaller and seams to indicate a residual bias, but at the limit, as the number of observations grows to infinity, the error should converge to zero. Performing the calibration with real data, the residual bias will be determined by the input error characteristics.

4.1.4. Real Data Results

The rotation estimation can be performed together with the camera calibration with the simple setup shown in Figure 14. The code used is available from the implemented InerVis toolbox (Lobo, 2006), that adds on to Bouguet's camera calibration toolbox (Bouguet 2006).

After rigidly fixing an inertial sensor to a camera rig, the calibration was performed with the proposed method.

The camera was calibrated with images of a vertical chequered target from several camera positions. Figure 15 shows some of the images used in this calibration and corresponding extrinsic parameters.

Since the chequered rectangular planar target is placed vertically, the camera verticals ${}^{\mathcal{C}}\boldsymbol{v}_i$ are given by the corresponding vanishing point of the vertical lines joining the corners of the chequered squares.

A total set of 16 images and accelerometer data was taken, and the estimated rotation was $\mathring{\mathbf{q}} = -0.7149 < 0.010013, 0.023479, 0.69876 >$, indicating a $-88.73°$ rotation about the axis (0.0143, 0.0336, 0.9993), i.e. a near right angle about the camera z-axis consistent with the layout shown in Figure 16.

Using the estimated rotation, the inertial sensed verticals where rotated to match with the vertical vanishing point of the chequered target, and the observed misalignment had a root mean square error of 0.69°, as shown in Figure 17.

The results show that the method performs well, and is easy to implement. From our experimental tests, of which the above is just an example, the key factors are the quality of the vanishing points obtained from the camera target images, that also determine the quality of the camera calibration.

4.2. Calibration of Translation between IMU and Camera

From (6) we can see that only dynamic motion will have nonzero acceleration from which translation $\mathbf{r}$ can be inferred.

A static boresight approach like the one used for rotation is easier to perform, since it only requires static poses with the target in view. If the IMU can be set to rotate about its sensing centre, then the camera motion will have the same rotation and a translation depending on the lever arm $\mathbf{r}$ joining the two.

With a turntable and suitable positioning rig the IMU can be set to rotate about its sensing centre, so that the accelerometers are not subject to rotation induced centripetal acceleration. This requires a mechanical rig, but not a controlled dynamic motion requiring expensive equipment. The output has to be monitored and adjustments made, starting from the expected sensing axis. After adjusting the IMU, if $2n$ observations are made for distinct camera positions, with the chequered target fixed and placed in camera view for each pair of measurements, lever arm $\mathbf{r}$ can be estimated.

The proposed setup will be a particular case of the hand–eye calibration used in robotics. Standard hand–eye calibration (Daniilidis 1999) can than formulated using homogeneous transformation matrices as solving

$$\boldsymbol{AX} = \boldsymbol{XB} \tag{19}$$

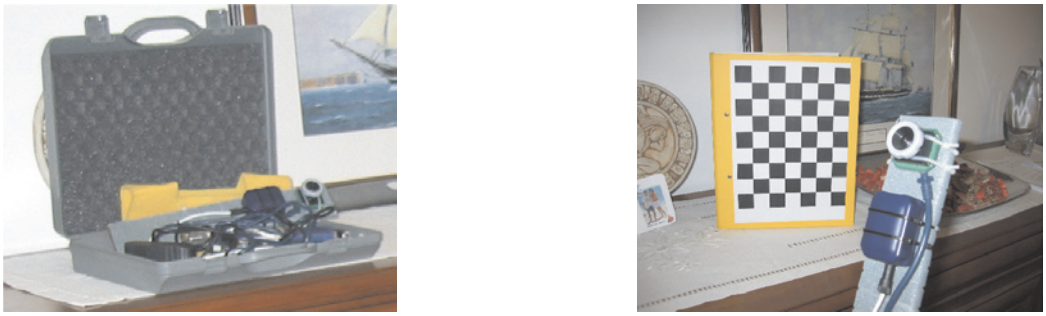


Fig. 14. Required setup for *out of the box* camera and inertial to camera rotation calibration.

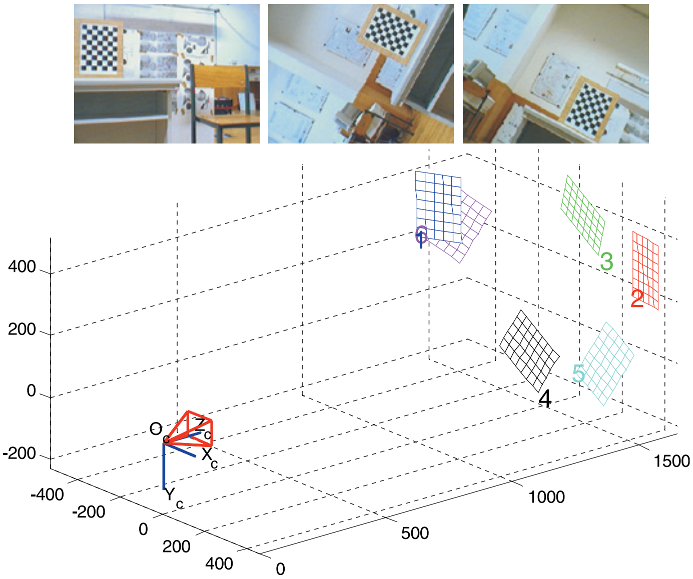


Fig. 15. Some of the images with vertical chequered target used for calibration and reconstructed target positions relative to the camera.

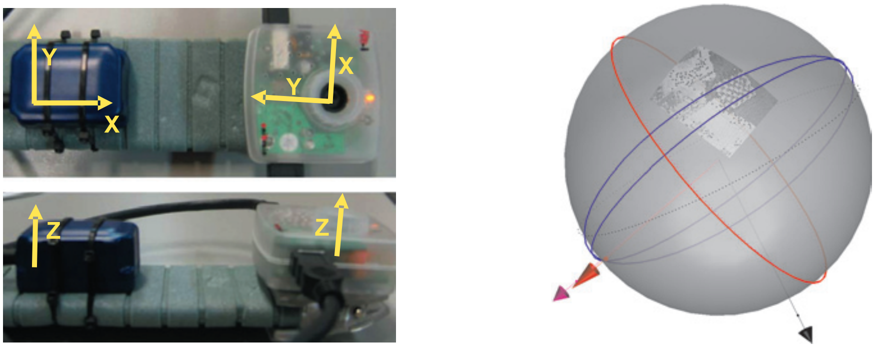


Fig. 16. Sensor layout, and rendered unit sphere showing vanishing point construction for one of the images, and observed and reprojected verticals from rotation calibration.

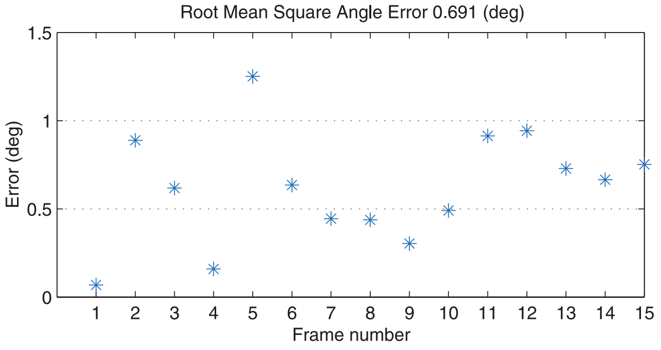


Fig. 17. Reprojection alignment errors for verticals in each frame used in rotation estimation

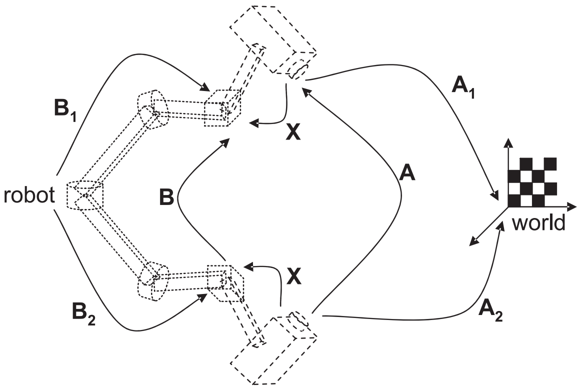


Fig. 18. Transformations between frames in robot with camera, where $\boldsymbol{X}$ is the unknown hand-to-eye transformation.

for an unknown hand-to-eye transformation $\boldsymbol{X}$, where A is the camera (eye) relative motion transformation, and B the gripper (hand) relative motion transformation, as shown in Figure 18.

This equation is a particular case of the Sylvester equation $\boldsymbol{AX} - \boldsymbol{XB} = \boldsymbol{C}$. Decomposing the homogeneous transformations in (19) into rotation and translation components (R, t) we get one matrix and one vector equation

$$\boldsymbol{R}_A\boldsymbol{R}_X = \boldsymbol{R}_X\boldsymbol{R}_B, \tag{20}$$

$$(\boldsymbol{R}_A - \boldsymbol{I})\,\mathbf{t}_X = \boldsymbol{R}_X\mathbf{t}_B - \mathbf{t}_A. \tag{21}$$

The majority of the approaches solve first for rotation (20) and than for translation (21). At least two motions with rotations about non-parallel axes are required.

When performing the hand–eye calibration for a robotic manipulator the relative camera transformation $\boldsymbol{A}$ can be obtained using a fixed world target and computing the camera-to-world transformation before and after the motion, $\boldsymbol{A}_1\ \boldsymbol{A}_2$, and making $\boldsymbol{A} = \boldsymbol{A}_2\boldsymbol{A}_1^{-1}$. Similarly, having the transformation matrices from the fixed robot base to the gripper, $\boldsymbol{B}_1\ \boldsymbol{B}_2$, we have $\boldsymbol{B} = \boldsymbol{B}_2^{-1}\boldsymbol{B}_1$. Keeping the robot base and target fixed, as shown in Figure 18, a set n poses can generate $(\frac{n!}{2!(n-2)!})$ relative motions for which the above equations can be solved.

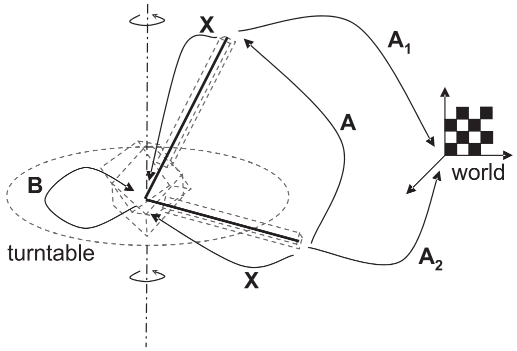


Fig. 19. Turntable used for unknown lever arm calibration as a hand-to-eye transformation for one turn. Complete calibration requires n turns, with $2n$ static poses with rotation about IMU null point.

The proposed translation calibration based on turning the system about the IMU centre falls within the above formulation with some simplifications. For our particular case we want to estimate the lever arm $\mathbf{r}$ in the camera frame of reference, and perform simple turns about the lever arm end point, adjusted to coincide with the inertial sensor centre. Each turn contributes with a set of two end point measurements before and after a turn about a pivot point centred on the inertial sensor. Our *hand* does not translate, and only rotates in exactly the same way as the camera, i.e. $\mathbf{t}_B = \mathbf{0}$, $\boldsymbol{R}_A = \boldsymbol{R}_B$ and $\boldsymbol{R}_X = \boldsymbol{I}$, and the transformations considered in Figure 18 are simplified as shown in Figure 19.

Rewriting (21) for this case we have

$$(\boldsymbol{R}_A - \boldsymbol{I})\,\mathbf{r} = -\mathbf{t}_A. \tag{22}$$

where the relative motion parameters can be obtained from the camera-to-target visual calibration. However, since the target is being repositioned after each turn, $2n$ poses only contribute n relative motions (turns) for the estimation of $\mathbf{r}$. Each pair contributes with the projection of $\mathbf{r}$ on the rotation plane, and at least two rotations about non parallel axis are required. The above equation can be rewritten for the n relative motions $\triangle_i$ as

$$\left(\boldsymbol{R}_{\triangle_i} - \boldsymbol{I}\right)\mathbf{r} = -\mathbf{t}_{\triangle_i}. \tag{23}$$

The camera translation $\mathbf{t}_{\triangle_i}$ induced by the lever arm $\mathbf{r}$ can be estimated by observing a fixed chequered target with the camera and recovering the extrinsic parameters. The final camera position relative to its initial position gives translation $\mathbf{t}_{\triangle_i}$ and rotation $\boldsymbol{R}_{\triangle_i}$.

Solving (23) for n turns using the standard hand–eye method (Tsai and Lenz 1989) we obtain the 3D lever arm $\mathbf{r}$ in the camera frame of reference. The n turns are performed as depicted in Figure 19. For each turn the system is repositioned

Translation Calibration Summary

- Perform standard camera calibration and rotation calibration with same image data set.
- 2N static observations with N turns about IMU at distinct poses:
  - position system on turntable;
  - force rapid motion and observe accelerometer output;
  - reposition until accelerometer output is null;
  - place the chequered target in camera view for the maximum turn amplitude;
  - save image and corresponding inertial data before and after the turn;
  - repeat for N turns with distinct axis about the IMU center.
- Compute translation (lever arm):
  - Use target vertical vanishing points ${}^{\mathcal{C}}\boldsymbol{v}_i$ detected for camera calibration;
  - Inertial data from accelerometers provide ${}^{\mathcal{I}}\boldsymbol{v}_i$;
  - Solve (23) for N turns using the standard hand-eye method to obtain the 3D lever arm $\mathbf{r}$

Fig. 20. Summary of required steps to perform calibration of translation between camera and IMU using the proposed algorithm.

with a distinct pose on the turntable and adjusted to rotate about a null point. The chequered target is also repositioned and placed in camera view for each pair of measurements, so that it is seen at the start and end of each turn.

4.2.1. Translation Calibration Summary

Figure 20 provides a summary of required steps to perform calibration of translation between camera and IMU using the proposed algorithm.

4.2.2. Error Sensitivity and Simulation Results

In order to validate the proposed method and perform noise sensitivity tests, simulations where performed under varying conditions.

The above described method takes a set of measured camera translations $\mathbf{t}_{\triangle_i}$ and rotations $\theta_{\triangle_i}$, induced by the unknown lever arm $\mathbf{r}$.

For each simulation run a random lever arm $\mathbf{r}$ is chosen and set of random rotations $\boldsymbol{R}_{\triangle_i}$ are applied to produce a set of simulated camera translations $\mathbf{t}_{\triangle_i}$.

With $\nu = \mathrm{SNR}^{-1} \in (0, 1)$ being the inverse of the signal to noise ratio, we disturb the simulated translation values $\mathbf{t}_{\triangle_i}$, by

$$\widetilde{\mathbf{t}_{\triangle_i}} = \mathbf{t}_{\triangle_i} + \nu \left\|\mathbf{t}_{\triangle_i}\right\| randn_{3\times 1} \qquad (24)$$

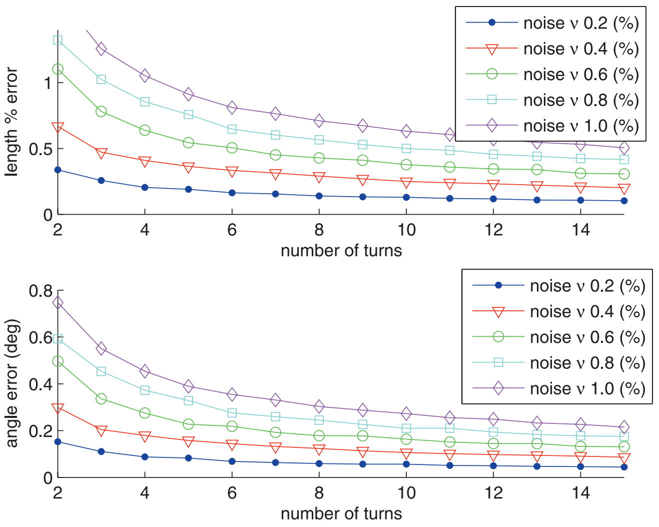


Fig. 21. Simulation translation estimation mean error for increasing number of turns.

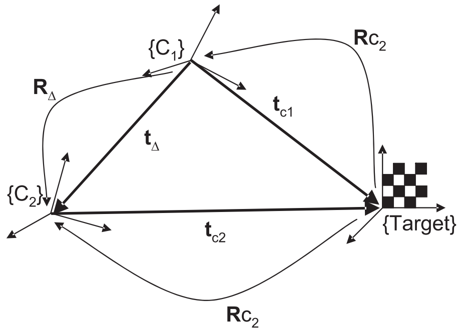


Fig. 22. Parameters obtained from camera calibration and derived translation induced by lever arm rotation.

where $randn_{n\times 1}$ is a n vector of random numbers that follow a uniform distribution, simulating Gaussian noise with zero mean and $\sigma = 1$.

The estimated lever arm $\hat{\mathbf{r}}$ is compared with the true simulation value $\mathbf{r}$, in length and alignment, to get the error measure. Figure 21 shows a set of simulation results of several takes with different noise levels and number of turns used, with 1000 runs in each take. Mean length error is given as a percentage of real value and angular error by its absolute mean value.

To better understand noise sensitivity issues, we have to take into account how the rotation induced translation is measured. By observing the chequered target and performing the camera calibration with Bouguet's camera calibration toolbox (Bouguet 2006), we obtain the camera extrinsic parameters for each image relative to the target, as shown in Figure 22.

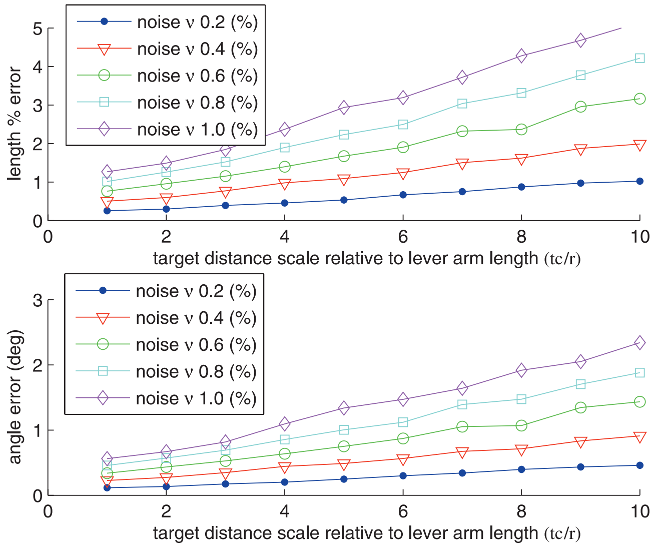


Fig. 23. Translation estimation from simulated camera extrinsic parameters for increasing target distance scale relative to lever arm length.

The above described camera translations $\mathbf{t}_{\triangle_i}$ and rotations $\boldsymbol{R}_{\triangle_i}$, induced by the unknown lever arm $\mathbf{r}$, can be derived from the camera extrinsic parameters as follows

$$\boldsymbol{R}_{\triangle_i} = \boldsymbol{R}c_1 \boldsymbol{R}c_2^{-1} \tag{25}$$

$$\mathbf{t}_{\triangle_i} = \boldsymbol{R}c_1 \left(\boldsymbol{R}c_2^{-1}\left(-\mathbf{t}c_2\right)\right) + \mathbf{t}c_1 \tag{26}$$

where index 1 and 2 indicate the initial and final extrinsic camera parameters for turn i, both relative to the camera position before the turn.

Since the real data will be derived in this way, a second simulation trial was made, but now adding Gaussian noise to $\mathbf{t}c_n$ and $\boldsymbol{R}c_n$. The behaviour of the method with added noise and number of turns has already been evaluated. The critical factor when considering the geometry presented in Figure 22 is the dilution of precision that results when estimating the translation with (26). To study this effect, the simulation runs where performed for different target distances, relative to the lever arm length.

Figure 23 shows simulation results of several takes with different noise levels and target distance to camera scale relative to lever arm length, using 10 turns per run, with 1000 runs in each take. Mean length error is given as a percentage of real value. The results clearly shows the limitations of the method, and that care has to be taken in positioning the target, so that the error is not amplified in the lever arm computation.

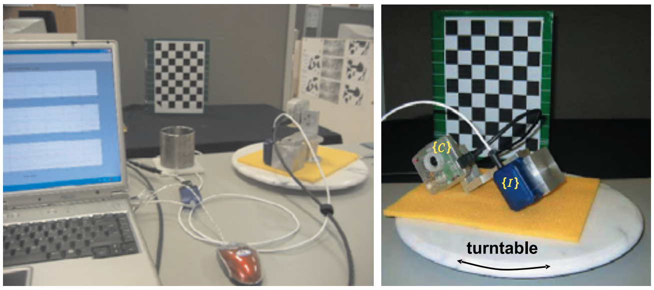


Fig. 24. Required setup for translation calibration, passive turntable and chequered calibration target.

4.2.3. Real Data Results

Figure 24 shows the setup required to perform the translation calibration. The system is placed on the passive turntable in several distinct poses. Each pose is adjusted to the null point by observing the inertial sensor outputs under forced motion, so that the rotation axis coincides with the inertial sensor centre. Two static camera images are than taken, before and after rotation, to obtain the lever arm induced translation used in the above described method to determine the lever arm or translation between the IMU and the camera.

Our method was initially compared with a standard implementation of the Tsai and Lenz (1989) hand–eye calibration. Assuming the fixed pivot point and fixed target, the gripper to camera transformation will be the lever arm translation, if the camera rotation is used as the world to gripper transformation. A rotating universal joint (Cardan) was used so that a fixed pivot could be used over several turns, as seen in Figure 25, so that both methods could be applied. With this setup sets of images were taken with the universal joint pivot placed in two different places, with several distinct turns about a single pivot point at each position with the chequered target always in view.

Table 1 presents the results. A total of 40 images where taken, the first 10 were used only to improve the camera calibration set, data set A has 5 turns (10 images) with a single pivot point and set B has 10 turns (20 images) with a distinct fixed pivot point. Results of lever arm estimation, $\mathbf{r} = (r_x, r_y, r_z)$ with $r = \|\mathbf{r}\|$, are shown for our method and for Tsai and Lenz applied to sets A&B, A and B, and $\bar{r}$ is the mean of the distinct estimates from set A and set B. The values shown in bold fall within the uncertainty of the direct ruler measurement r_m.

By considering that the pivot point and the target are always fixed, the Tsai and Lenz hand–eye calibration method performs a global optimization using all the images, showing a better performance for sets A and B. When the method is applied to the complete data set A&B it fails completely since its not applicable. Our method just requires sets of turns between which both the target and pivot point can be repositioned. It is based only on the relative camera motion in each turn, and is

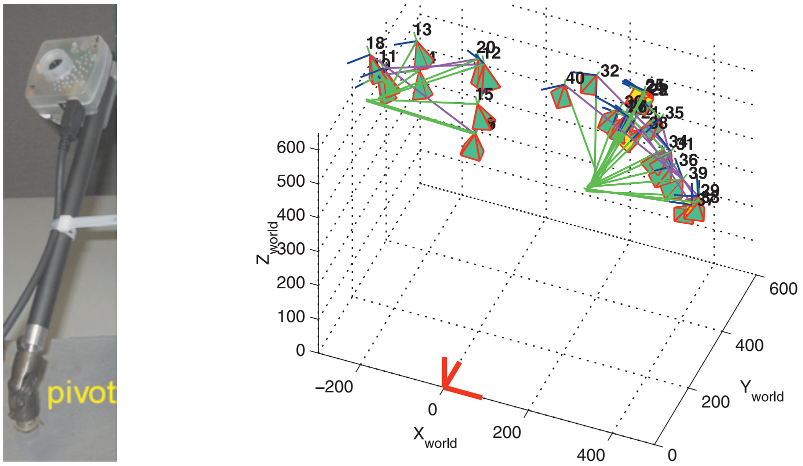


Fig. 25. Setup used and camera reconstructed pose relative to calibration target, with the pivot point at two different positions, showing frame number, camera orientation and the estimated lever arm.

Table 1. Translation estimation using two data sets with fixed pivot point

	Our method					Tsai and Lenz					
	A&B	A	B	$\bar{r}$	σ	A&B	A	B	$\bar{r}$	σ	r_m
r_x (mm)	**252.54**	**252.77**	**253.05**	**252.91**	0.14	81.70	**251.16**	**251.79**	**251.48**	0.32	**249±5**
r_y (mm)	**26.27**	29.85	**22.28**	**26.07**	3.79	–75.00	28.72	**21.67**	**25.20**	3.52	**25±3**
r_z (mm)	**–31.57**	–34.47	**–29.64**	**–32.05**	2.42	793.01	**–28.71**	**–27.78**	**–28.24**	0.46	**–31±3**
r (mm)	**255.86**	**256.85**	**255.75**	**256.26**	0.55	800.72	**254.42**	**254.25**	**254.31**	0.09	**252±5**

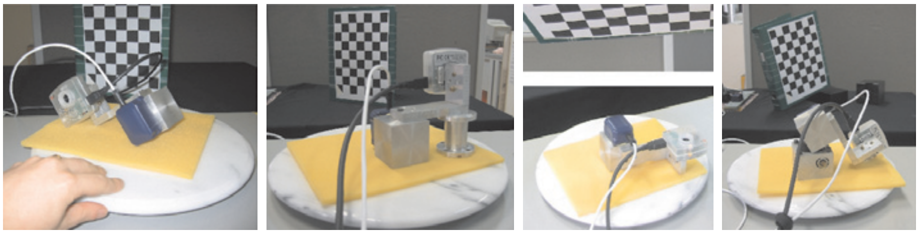


Fig. 26. System placed on turntable in different poses for translation calibration.

therefore more sensitive and prone to errors. But, as we will see in the second example, requires a much simpler setup and can provide a good estimate of the lever arm under controlled conditions.

A second calibration was done with a passive turntable, placing the camera with attached inertial sensors in different poses as shown in Figures 24 and 26, and fine adjusting the position to zero the force sensed by the accelerometers, besides gravity, placing them at the rotation centre.

With the passive turntable setup a set of 30 images was taken, corresponding to 15 distinct turns. The accelerometer output was observed while manually forcing rapid turns to adjust their position to the centre of rotation. The chequered target was conveniently placed, and the reconstruction result for the complete set is shown in Figure 27.

In table 2 results are presented for several groupings of sets of measurements (sets are labelled as start:step:end), to better evaluate the estimation performance. Direct measurement of the lever arm indicated a length about 125 ± $10mm$, since the exact position of the accelerometers within the packaged sensor is not known, confirming the estimated value.

The implemented code for translation estimation will be made available in the InerVis toolbox (Lobo 2006).

Table 2. Translation estimation using turntable

n	1:1:15	1:2:15	1:1:10	1:2:11	5:1:15	5:2:15	mean	σ
r_x (mm)	–87.4	–86.7	–92.9	–86.6	–83.0	–83.2	–86.6	3.6
r_y (mm)	91.7	91.6	92.0	91.5	93.1	92.1	92.0	0.6
r_z (mm)	2.6	1.7	1.8	1.7	6.2	2.8	2.8	1.7
r (mm)	126.7	126.1	130.8	126.0	124.9	124.1	126.4	2.3

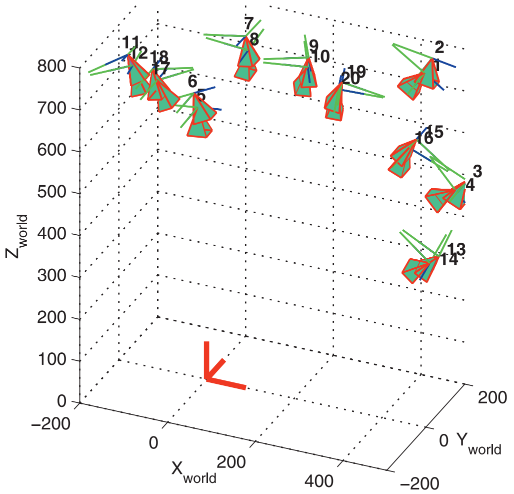


Fig. 27. Camera reconstructed pose relative to calibration target.

5. Conclusions

We have seen how a simple calibration can be made with off-the-shelf cameras and inertial sensors to have a useful integrated system.

With a set of static poses observing a vertical target, full camera calibration can be performed using standard techniques, and inertial sensor to camera rotation can estimated as well by registering the inertial sensed gravity. With a simple passive turntable and with $2n$ static poses of n rotations about the inertial sensor centre, the translation between the two sensors can also be estimated.

The method works well in estimating rotation, but the translation estimation is sensitive to the chosen target position, and care has to be taken so that the geometric configuration does not magnify the error in the visual target pose onto the final lever arm estimation.

Lever arm calibration can also be accomplished using standard Hand/Eye calibration (Daniilidis 1999), like the Tsai and Lenz (1989) implementation used above for comparison. These methods are clearly more stable but require a single fixed pivot point. Our method only uses the relative camera motion in each turn, but Hand/Eye methods use the full camera and hand pose data over the complete data set. But they are also more restrictive on the setup. A simple turntable is no longer sufficient, since a fixed pivot point has to be maintained. A passive double gimbal might prove useful, but would have to accomodate for proper centering of the system, and using an active controlled manipulator might be better. Our aim however is to have a simple procedure to estimate the lever arm, that can be performed without complicated equipment, and complement the simple procedure used for camera and rotation calibration.

If some assumptions can be made about the observed scene, such as the predominance of vertical features in an indoor environment, a self-calibrating version of the proposed rotation calibration could be implemented, as long as the identification of vertical features could be assured. Self-calibration algorithms that only assume the rigidity of a static background are based on the geometry constraints of how scene points should appear in the image seen from different perspectives. By moving a camera in a static scene the rigidity of the scene generates new constraints for each view, and correspondences between three images are sufficient to recover both the internal and external parameters, up to an unknown scale factor. The obvious advantage of this method is that no special calibration target is required, but point correspondences over many images are required. The inertial sensors can also be used to estimate the scale factor, by providing, through double integration of acceleration, an absolute measure on translation between frames.

References

Alves, J., Lobo, J., and Dias, J. (2003). Camera-inertial sensor modeling and alignment for visual navigation. *Machine Intelligence and Robotic Control*, **5**(3): 103–112.

Bouguet, J.-Y. (2006). Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

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

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

Dorst, L. (2005). First order error propagation of the procrustes method for 3D attitude estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(2): 221–229.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, **4**(4): 629–462.

Lobo, J. and Dias, J. (2003). Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(12): 1597–1608.

Lobo, J. and Dias, J. (2004). Inertial sensed ego-motion for 3d vision. *Journal of Robotic Systems*, **21**(1): 3–12.

Lobo, J. (2006). InerVis Toolbox for Matlab. http://www.deec.uc.pt/~jlobo/InerVis_WebIndex/

Nebot, E. and Durrant-Whyte, H. (1997). Initial calibration and alignment of an inertial navigation system. *Proceedings of the 4th Annual Conference on Mechatronics and Machine Vision in Practice*, IEEE Computer Society, p.175.

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

Viéville, T. and Faugeras, O. D. (1989). Computation of inertial information on a robot. In *Fifth International Symposium on Robotics Research* (eds H. Miura and S. Arimoto), pp.57–65, MIT Press.

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, Kerkyra, Greece, September, Vol. 1, pp.666–673.