Relative Pose Calibration Between Visual and Inertial Sensors

Jorge Lobo and Jorge Dias
Institute of Systems and Robotics
University of Coimbra
3030-290 Coimbra, Portugal
{jlobo,jorge}@isr.uc.pt

***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. The rotation between the camera and the inertial sensor can be estimated, when calibrating the camera, by having both sensors observe the vertical direction, using a vertical chessboard target and gravity. The translation between the two can be estimated using a simple passive turntable and static images, provided that the system can be adjusted to turn about the inertial sensor null point in several poses. Simulation and real data results are presented to show the validity and simple requirements of the proposed method.**

***Index Terms*— computer vision, inertial sensors, sensor fusion, calibration.**

I. 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.

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 [1][2]. 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 sensor 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 chessboard target, simplifying the whole calibration procedure.

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 positioning the integrated camera and inertial system centered on the inertial sensor, the lever arm can be estimated. Observing the inertial sensor outputs, the system can be adjusted to turn about their null point in several poses. The lever arm can than be estimated from static images of a suitably placed visual target before and after each rotation.

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 [3]. 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 [4].

Standard hand-eye calibration [5][6] can be applied to estimate translation, using the approach of rotating about the inertial sensor center. However, since the target is being repositioned after each turn, the method is not applied to the full data set like in traditional hand-eye calibration. We used an implementation of the full hand-eye calibration [5] to provide a comparison in the results using only a camera with fixed lever arm, by keeping a constant pivot point.

II. Stand Alone Sensor Calibration

A. Camera Calibration

Camera calibration has been extensively studied, and standard techniques established. For this work camera calibration was performed using the Camera Calibration Toolbox for Matlab [7]. The C implementation of this toolbox is included in the Intel Open Source Computer Vision Library [8].

The calibration uses images of a chessboard target in several positions and recovers the camera's intrinsic parameters, as well as the target positions relative to the camera. The calibration algorithm is based on Zhang's work in estimation of planar homographies for camera calibration [9], 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 was also used to recover camera extrinsic parameters in the subsequent relative pose calibration.

B. Inertial Sensor Calibration

Inertial navigation systems also have established calibration techniques, but rely on high-end sensors and actuators. Nevertheless, in order to use off-the-shelf inertial sensors attached to a camera, appropriate modelling and calibration techniques are required. Some of the inertial

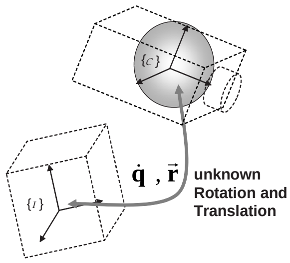

Fig. 1. Unknown rigid body transformation between IMU and camera frames of reference.

sensors parameters can be determined by performing simple operations and measuring the sensor outputs, others can not be so easily determined. We have 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 [10]. 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 inertial measuring unit (IMU). When considering a complete inertial navigation system, initial calibration and alignment are more elaborate [11].

III. Camera and IMU Data Relationship

A. IMU Data in Camera Frame of Reference

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 unit quaternion $\mathring{q}$ that rotates inertial measurements in the inertial sensor frame of reference $\{\mathcal{I}\}$ to the camera frame of reference $\{\mathcal{C}\}$, and translation vector $\vec{r}$. Quaternion algebra was developed by Hamilton in the nineteenth century as an extension of imaginary numbers to higher dimensions. Unit quaternions provide a convenient rotation representation [4].

1) Angular Velocity of Camera Center of Projection: Any point of a rigid body has the same angular velocity. To obtain the camera angular velocity in the camera frame of reference, we just rotate apply the known rotation between the two frames of reference:

$$^{\mathcal{C}}\vec{\omega} = \mathring{q}\,{}^{\mathcal{I}}\vec{\omega}\,\mathring{q}^* \tag{1}$$

2) Linear Acceleration of Camera Center of Projection: 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, i.e.

$$^{\mathcal{C}}\vec{a} = \mathring{q}\,({}^{\mathcal{I}}\vec{a} - {}^{\mathcal{I}}\vec{a}_c)\mathring{q}^* + {}^{\mathcal{C}}\vec{a}_c \tag{2}$$

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

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

$$\vec{a}_c = \vec{\omega} \times \vec{v}_t = \vec{\omega} \times (\vec{\omega} \times \vec{r}) \tag{3}$$

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

If we assume that the rotation axis goes through the camera center of projection, than it will not have centripetal acceleration and its linear acceleration is given by

$$\begin{aligned} {}^{\mathcal{C}}\vec{a} &= \mathring{q}\,({}^{\mathcal{I}}\vec{a} - {}^{\mathcal{I}}\vec{a}_c)\mathring{q}^* \\ &= \mathring{q}\,({}^{\mathcal{I}}\vec{a} - {}^{\mathcal{I}}\vec{\omega} \times ({}^{\mathcal{I}}\vec{\omega} \times {}^{\mathcal{I}}\vec{r}))\mathring{q}^* \\ &= \mathring{q}\,{}^{\mathcal{I}}\vec{a}\,\mathring{q}^* + {}^{\mathcal{C}}\vec{\omega} \times ({}^{\mathcal{C}}\vec{\omega} \times {}^{\mathcal{C}}\vec{r}) \end{aligned} \tag{4}$$

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

If we assume that the rotation axis goes though the IMU center, than no centripetal acceleration will be sensed, and the camera linear acceleration is given by

$$^{\mathcal{C}}\vec{a} = \mathring{q}\,{}^{\mathcal{I}}\vec{a}\,\mathring{q}^* - {}^{\mathcal{C}}\vec{\omega} \times ({}^{\mathcal{C}}\vec{\omega} \times {}^{\mathcal{C}}\vec{r}) \tag{5}$$

The rigid body transformation between the IMU and the camera has to be calibrated when using both sensors. Direct physical measurements are difficult to perform, since the camera center of projection and inertial sensor sensing point and axis are not obvious. But rotation $\mathring{q}$ and translation $\vec{r}$ can be derived from (1) and (5) provided something is known about the motion.

B. 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 fig. 2. 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 chessboard target placed vertically, the vertical direction can be taken from the corresponding vanishing point.

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

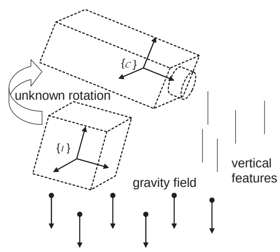

Fig. 2. IMU and camera observing gravity.

If n observations are made for distinct camera positions, recording the vertical reference provided by the inertial sensors 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 [4], 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 ${}^{\mathcal{I}}\vec{\boldsymbol{v}}_i$ be a measurement of the vertical by the inertial sensors, and ${}^{\mathcal{C}}\vec{\boldsymbol{v}}_i$ the corresponding measurement made by the camera derived from some scene vanishing point. We want to determine the unit quaternion $\mathring{\boldsymbol{q}}$ 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 $\mathring{\boldsymbol{q}}$ that maximises

$$\sum_{i=1}^{n} (\mathring{\boldsymbol{q}}\, {}^{\mathcal{I}}\vec{\boldsymbol{v}}_i\, \mathring{\boldsymbol{q}}^*) \cdot {}^{\mathcal{C}}\vec{\boldsymbol{v}}_i \tag{6}$$

which after some manipulation can be expresses as finding $\mathring{\boldsymbol{q}}$ such that

$$\max \mathring{\boldsymbol{q}}^T\, \vec{\boldsymbol{N}}\, \mathring{\boldsymbol{q}} \tag{7}$$

where the elements of matrix $\vec{\boldsymbol{N}}$ can be expressed using sums of all 9 product pairings of the components of the two vector sets. The sums contain all the information that is required to find the solution. Since $\vec{\boldsymbol{N}}$ is a symmetric matrix, the solution to this problem is the four-vector $\vec{\boldsymbol{q}}_{max}$ corresponding to the largest eigenvalue λ_{max} of $\vec{\boldsymbol{N}}$ - see [4] for details. A more detailed derivation and some results of this calibration method are presented in [12] and [10].

C. Calibration of Translation between IMU and Camera

From (5) we can see that only dynamic motion will have non zero acceleration from which translation $\vec{\boldsymbol{r}}$ can be inferred.

A static boresight approach like the one used for rotation is easier to perform. If the IMU can be set to rotate about its sensing point and axis, than the camera motion will have the same rotation and a translation depending on the lever arm $\vec{\boldsymbol{r}}$ joining the two.

With a turntable and suitable positioning rig the IMU can be set to rotate about a null point. 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 chessboard target fixed and placed in camera view for each pair of measurements, lever arm $\vec{\boldsymbol{r}}$ can be estimated.

Standard hand-eye calibration [6] can than formulated using homogeneous transformation matrices as solving

$$\boldsymbol{AX} = \boldsymbol{XB} \tag{8}$$

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. This

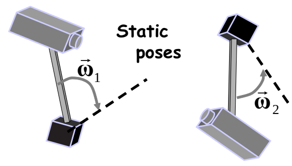

Fig. 3. n turns, $2n$ static poses with rotation about IMU null point.

equation is a particular case of the Sylvester equation $\boldsymbol{AX} - \boldsymbol{XB} = \boldsymbol{C}$. Decomposing the homogeneous transformations in (8) into rotation and translation components (R, t) we get one matrix and one vector equation

$$\boldsymbol{R}_A \boldsymbol{R}_X = \boldsymbol{R}_X \boldsymbol{R}_B, \tag{9}$$

$$(\boldsymbol{R}_A - \boldsymbol{I})\, \vec{\boldsymbol{t}}_X = \boldsymbol{R}_X \vec{\boldsymbol{t}}_B - \vec{\boldsymbol{t}}_A. \tag{10}$$

The majority of the approaches solve first for rotation (9) and than for translation (10). At least two motions with rotations about non parallel axis 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, a set n poses can generate $(\frac{n!}{2!(n-2)!})$ relative motions for which the above equations can be solved.

For our particular case we want to estimate the lever arm $\vec{\boldsymbol{r}}$ in the camera frame of reference, and perform simple turns about the lever arm end point, adjusted to coincide with the inertial sensor center. Our *hand* does not translate, and only rotates in exactly the same way as the camera, i.e. $\vec{\boldsymbol{t}}_B = \vec{\boldsymbol{0}}$, $\boldsymbol{R}_A = \boldsymbol{R}_B$ and $\boldsymbol{R}_X = \boldsymbol{I}$. Rewriting (10) for this case we have

$$(\boldsymbol{R}_A - \boldsymbol{I})\, \vec{\boldsymbol{r}} = -\vec{\boldsymbol{t}}_A. \tag{11}$$

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 for the estimation of $\vec{\boldsymbol{r}}$. Each pair contributes with the projection of $\vec{\boldsymbol{r}}$ 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

$$(\boldsymbol{R}_{\triangle_i} - \boldsymbol{I})\, \vec{\boldsymbol{r}} = -\vec{\boldsymbol{t}}_{\triangle_i}. \tag{12}$$

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

Solving (12) for n turns using the standard hand-eye method [5] we obtain the 3D lever arm $\vec{\boldsymbol{r}}$ in the camera frame of reference.

IV. Simulation Results

In order to validate the proposed methods and perform noise sensitivity tests, both where tested in simulation under varying conditions.

A. Rotation Estimation

For each simulation run a random rotation $\mathring{\boldsymbol{q}}$ is applied to a random set of simulated inertial observed verticals, ${}^{\mathcal{I}}\overrightarrow{\boldsymbol{v}}_i$, to obtain a corresponding set of camera observed verticals, ${}^{\mathcal{C}}\overrightarrow{\boldsymbol{v}}_i$. These simulated camera observations are corrupted by applying a rotation magnitude with zero mean and absolute standard deviation about a random axis, i.e. a uniformly distributed 3D axis. The rotation quaternion that relates the two sets is estimated as $\hat{\mathring{\boldsymbol{q}}}$ 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, $\mathring{\boldsymbol{q}} = \mathring{\boldsymbol{e}} * \hat{\mathring{\boldsymbol{q}}}$. With $\theta_e = 2\cos^{-1}(e_s)$, where $e_s$is the scalar component of $\mathring{\boldsymbol{e}}$, we take $\delta_\theta = |\theta_e|$ as the error measure. Fig. 4 shows simulation results of several takes with different noise levels and number of observations used, with 1000 runs in each take. As expected the method performs well, even with few observations.

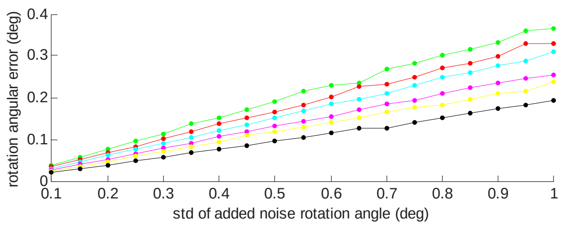

Fig. 4. Simulation rotation estimation mean error for increasing standard deviation of zero mean gaussian noise added as a rotation of the observed camera verticals. The decreasing error lines correspond to simulating 4,5,6,8,10 and 15 observations. For each noise setting the method runs 1000 times and the mean error is evaluated.

B. Translation Estimation

The above described method takes a set of measured camera translations $\overrightarrow{\boldsymbol{t}}_{\Delta_i}$ and rotations θ_{Δ_i}, induced by the unknown lever arm $\overrightarrow{\boldsymbol{r}}$.

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

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

$$\widetilde{\overrightarrow{\boldsymbol{t}}}_{\Delta_i} = \overrightarrow{\boldsymbol{t}}_{\Delta_i} + \nu \left\| \overrightarrow{\boldsymbol{t}}_{\Delta_i} \right\| randn_{3\times1} \tag{13}$$

where $randn_{n\times1}$ is a n vector of random numbers that follow a uniform distribution, simulating white gaussian noise with zero mean and $\sigma = 1$.

The estimated lever arm $\hat{\overrightarrow{\boldsymbol{r}}}$ is compared with the true simulation value $\overrightarrow{\boldsymbol{r}}$, in length and alignment, to get the error measure. Fig. 5 shows a set of simulation results of several takes with different noise levels and number of turns used, with 1000 runs in each take.

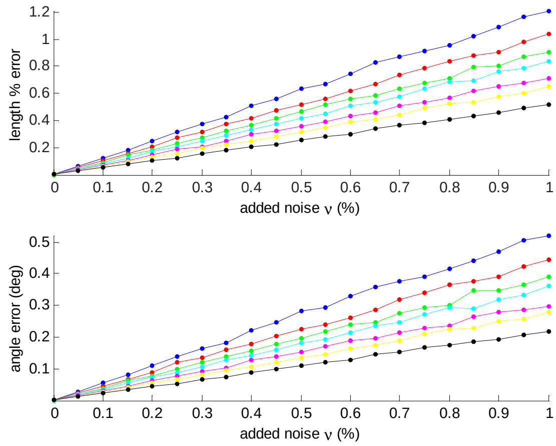

Fig. 5. Simulation translation estimation mean error for increasing noise ν. Mean length error is given as a percentage of real value and angular error by its absolute mean value. The decreasing error lines correspond to simulating 3,4,5,6,8,10 and 15 turns.

To better understand noise sensitivity issues, we have to take into account how the rotation induced translation is measured. By observing the chessboard target and performing the camera calibration with the Matlab Camera Calibration Toolbox [7], we obtain the camera extrinsic parameters for each image relative to the target, as shown in fig. 6.

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

$$\boldsymbol{R}_{\Delta_i} = \boldsymbol{R}c_1 \boldsymbol{R}c_2^{-1} \tag{14}$$

$$\overrightarrow{\boldsymbol{t}}_{\Delta_i} = \boldsymbol{R}c_1 \left(\boldsymbol{R}c_2^{-1}\left(-\overrightarrow{\boldsymbol{t}}c_2\right)\right) + \overrightarrow{\boldsymbol{t}}c_1 \tag{15}$$

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 white gaussian noise to $\overrightarrow{\boldsymbol{t}}c_n$ and $\boldsymbol{R}c_n$. The behavior of the method with added noise and number of turns has already been evaluated. The critical factor when considering the geometry presented in fig. 6 is the dilution of precision that results when estimating the translation with (15). To study this effect, the simulation runs where performed for different target distances, relative to the lever arm length.

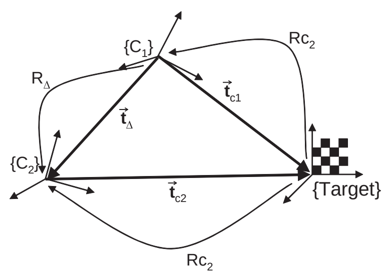

Fig. 6. Parameters obtained from camera calibration and derived translation induced by lever arm rotation.

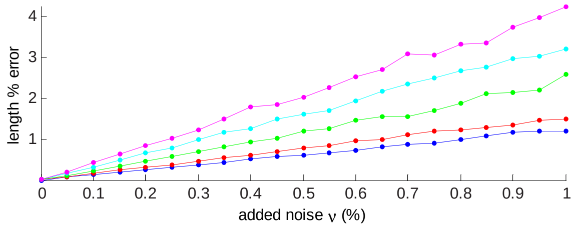

Fig. 7. Translation estimation from simulated camera extrinsic parameters for increasing noise ν. Mean length error is given as a percentage of real value. The increasing error lines correspond to simulating 1,2,4,6 and 8 target distance scale relative to lever arm length.

Fig. 8. Required setup for rotation calibration, and turntable used for translation calibration

Fig. 7 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. 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.

V. Real Data Results

A. Rotation Estimation

The rotation estimation can be performed together with the camera calibration with the simple setup shown in fig. 8. The code used is available from the implemented InerVis Matlab Toolbox [13], that adds on to the Camera Calibration Toolbox [7].

To present some results, a data set was obtained using MT9-B IMU sensor from Xsens [14] and a low cost firewire camera from Unibrain shown in fig. 9. A set of 16 images and accelerometer data was taken, and the estimated rotation was $\mathring{\boldsymbol{q}} = -0.7149 < 0.010013, 0.023479, 0.69876 >$, 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 fig. 9. Using the estimated rotation, the inertial sensed verticals where rotated to match with the vertical vanishing point of the chessboard target, and the observed misalignment had a root mean square error of $0.69°$.

B. Translation Estimation

To better assess the calibration performance a rotating u-joint was initially used so that a fixed pivot could be used over several turns, enabling the use of standard hand-eye calibration methods for comparison, as seen in fig. 10. With

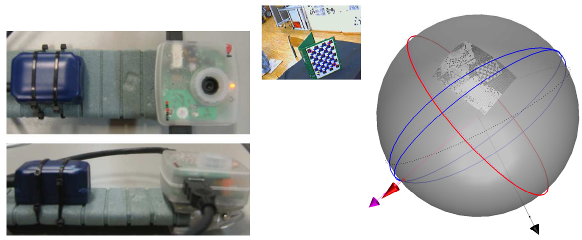

Fig. 9. Setup and results of rotation estimation

this setup a set of 30 images was taken, corresponding to 15 distinct turns about a single pivot point with the chessboard target always in view, placed in 2 different places during image acquisition.

Our method is compared with a standard implementation of the Tsai and Lenz [5] 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.

Table Ia 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, $\overrightarrow{\boldsymbol{r}} = (r_x, r_y, r_z)$ with $r = \|\overrightarrow{\boldsymbol{r}}\|$, 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.

Tsai and Lenz clearly has a better performance, since it performs a global optimization using all the images by considering the pivot point and the target are always fixed. 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 therefor 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 differ-

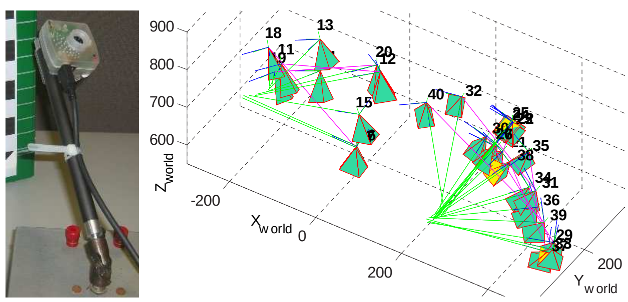

Fig. 10. Camera reconstructed pose relative to calibration target, with the pivot point at two different positions, showing frame number, camera orientation and the estimated lever arm in green.

TABLE I

a) 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	**252.54**	**252.77**	**253.05**	**252.91**	0.14	81.70	**251.16**	**251.79**	**251.48**	0.32	**249±5**
r_y	**26.27**	29.85	**22.28**	**26.07**	3.79	-75.00	28.72	**21.67**	**25.20**	3.52	**25±3**
r_z	**-31.57**	-34.47	**-29.64**	**-32.05**	2.42	793.01	**-28.71**	**-27.78**	**-28.24**	0.46	**-31±3**
r	**255.86**	**256.85**	**255.75**	**256.26**	0.55	800.72	**254.42**	**254.25**	**254.31**	0.09	**252±5**

b) 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	-87.4	-86.7	-92.9	-86.6	-83.0	-83.2	-86.6	3.6
r_y	91.7	91.6	92.0	91.5	93.1	92.1	92.0	0.6
r_z	2.6	1.7	1.8	1.7	6.2	2.8	2.8	1.7
r	126.7	126.1	130.8	126.0	124.9	124.1	126.4	2.3

ent poses as shown in fig. 8 and 11, and fine adjusting the position to zero the force sensed by the accelerometers, besides gravity, placing them at the rotation center.

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 center of rotation. The chessboard target was conveniently placed, and the reconstruction result for the complete set is shown in fig. 11.

In table Ib results are presented for several groupings of sets of measurements, to better evaluate the estimation performance. Direct measurement of the lever arm indicated a length about $125 \pm 10mm$, 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 Matlab Toolbox [13], that currently only performs rotation estimation.

VI. 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, 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.

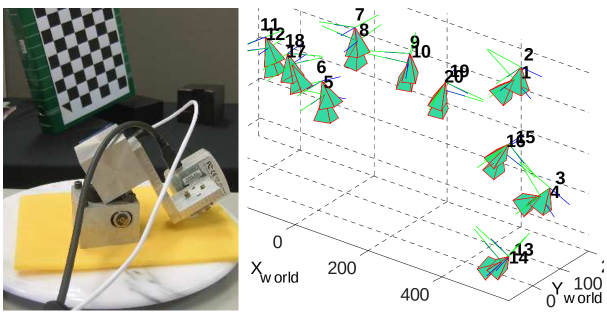

Fig. 11. Camera reconstructed pose relative to calibration target showing estimated lever arm in green.

Lever arm calibration can also be accomplished using standard Hand/Eye calibration [6], like the Tsai and Lenz implementation used above for comparison [5]. These methods, applied here in a simplified case where the camera rotation is used as the base-to-hand transformation, are clearly more stable. 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 accommodate 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.

References

[1] Jorge Lobo and Jorge Dias. Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1597–1608, 2003.

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

[3] Leo Dorst. First order error propagation of the procrustes method for 3d attitude estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2):221–229, 2005.

[4] B.K.P Horn. Closed-Form Solution of Absolute Orientation Using Unit Quaternions. *Journal of the Optical Society of America*, 4(4):629–462, April 1987.

[5] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE Trans. Robot. Automat.*, 5:345358, 1989.

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

[7] Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[8] Intel. Intel Open Source Computer Vision Library. http://www.intel.com/research/mrl/research/opencv/index.htm.

[9] Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 666–673, Kerkyra, Greece, September 1999.

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

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

[12] João Alves, Jorge Lobo, and Jorge Dias. Camera-Inertial Sensor modelling and alignment for Visual Navigation. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 1693–1698, Coimbra, Portugal, July 2003.

[13] Jorge Lobo. InerVis Toolbox for Matlab. http://www.deec.uc.pt/ jlobo/InerVis_WebIndex/.

[14] Xsens Technologies. http://www.xsens.com/.