

# Camera-Inertial Sensor Modeling and Alignment for Visual Navigation\*

João Alves<sup>†</sup>, Jorge Lobo<sup>†</sup>, and Jorge Dias<sup>†</sup>

**Abstract:** This article presents a technique for modeling and calibrating a camera with integrated low-cost inertial sensors, three gyros and three accelerometers for full 3D sensing. Inertial sensors attached to a camera can provide valuable data about camera pose and movement. In biological vision systems, inertial cues provided by the vestibular system, are fused with vision at an early processing stage. Vision systems in autonomous vehicles can also benefit by taking inertial cues into account. Camera calibration has been extensively studied, and standard techniques established. Inertial navigation systems, relying on high-end sensors, also have established techniques. Nevertheless, in order to use off-the-shelf inertial sensors attached to a camera, appropriate modeling and calibration techniques are required. For inertial sensor alignment, a pendulum instrumented with an encoded shaft is used to estimate the bias and scale factor of inertial measurements. For camera calibration, a standard and reliable camera calibration technique is used, based on images of a planar grid. Having both the camera and the inertial sensors calibrated and observing the vertical direction at different poses, the rigid rotation between the two frames of reference is estimated, using a mathematical model based on unit quaternions. The technique for this alignment and consequent results with simulated and real data are presented at the end of this article.

**Keywords:** Camera, Inertial Sensors, Camera Calibration, Inertial Sensor Alignment, Sensor Model

## 1. Introduction

**I**NTERNAL sensing using inertial sensors is very useful in mobile robotic systems and autonomous vehicles since it is not dependent on any external references, except for the gravity field which does provide an external reference. Artificial vision systems can provide better perception of the vehicle's environment by using the inertial sensor measurement of camera pose (rotation and translation).

This paper presents a technique for modeling and calibrating a camera with integrated inertial sensors. Having both the camera and the inertial sensors observing the vertical direction at different poses, the rigid rotation between the two frames of reference  $\{C\}$  and  $\{I\}$  shown in Fig. 1, can be estimated.

This work is part of ongoing research into the fusion of inertial sensor data in artificial vision systems for applications on autonomous vehicles such as CyberCars [1]. A technique for ground plane segmentation, the inertial sensors, and the system prototype are described in [3] and [4]. In [5] a method is proposed for camera focal distance calibration using a vanishing point and the vertical reference. In [6] a vertical line segmentation method is described that performs the 3D reconstruction and mapping of the detected vertical line segments.

This paper is organized as follows: in Section 2 the data from the inertial sensors is considered. A calibration method using a pendulum with an encoded shaft is presented to estimate inertial sensor alignment, bias and scale factor. Section 3 introduces the camera model and the properties of vanishing points. The following section presents

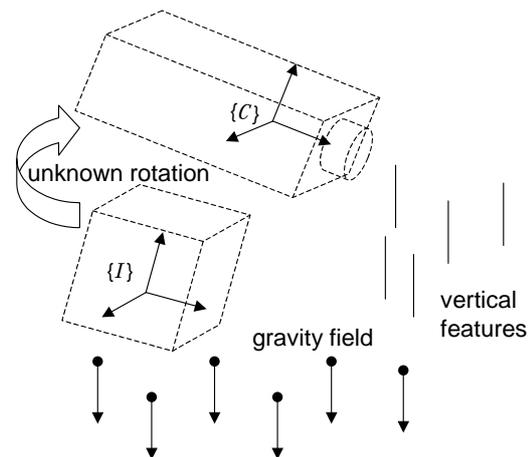


Fig. 1 Observing gravity with the camera and the inertial sensors, the unknown rotation can be determined

the estimation of the rigid rotation between the inertial sensors and camera frames of reference. In Section 5 results are presented for both inertial sensor calibration and frame rotation estimation.

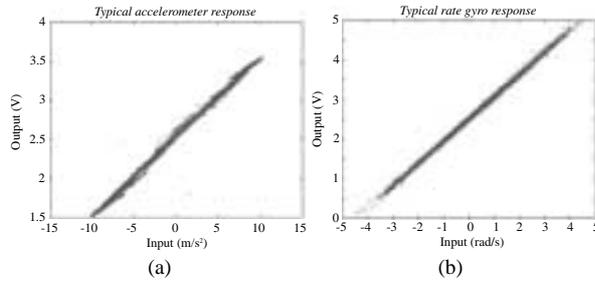
## 2. Data from Inertial Sensors

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 errors. 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 minimize the error buildup.

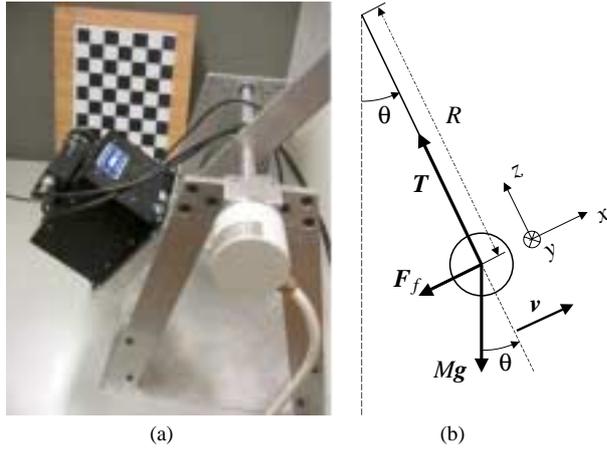
When using inertial sensors, scale factor, bias and axis-alignment need to be known. For low cost inertial sensors these parameters are not always provided by the manufac-

\* Received July 18, 2003; accepted November 12, 2003.

<sup>†</sup> Dept. of Electrical and Computer Engineering, Institute of Systems and Robotics, University of Coimbra - Polo II, 3030-290 Coimbra, Portugal. E-mail: {jalves, jlobo, jorge}@isr.uc.pt



**Fig. 2** Sensors' response: (a) accelerometer, (b) rate gyro



**Fig. 3** (a) Pendulum used to calibrate the inertial sensors, (b) forces acting on a moving pendulum

turer, and when using discrete components their alignment has to be measured.

## 2.1 Intrinsic calibration

Some of the inertial sensor parameters can be determined by performing simple operations and measuring the sensor outputs; but others cannot be so easily determined.

Observing the sensor response which is illustrated in **Fig. 2**, for a particular accelerometer and a particular rate gyro, it can be seen that this response is practically linear, and for that reason, in this work, a linear model will be used for the inertial sensors. This model is satisfactory to use with our autonomous mobile vehicles.

Equation (1) represents a simple model for each set of three non-coplanar accelerometers or rate gyros, which accounts for the three main errors in these sensors: bias, scale factors and cross-axis sensitivity.

$$\begin{aligned} z_o &= Mz_i + b \\ &= \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} \begin{bmatrix} z_{ix} \\ z_{iy} \\ z_{iz} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}. \end{aligned} \quad (1)$$

The quantities to be measured are represented by the vector  $z_i$ , while  $z_o$  represents the actual output from the sensors. Vector  $b$  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  $l$ .

## 2.2 Calibration with a pendulum

In this work, a pendulum is used in order to determine the inertial sensor parameters—see **Fig. 3**.

The pendulum was chosen since it is relatively straightforward to determine the real quantities the sensors are measuring. To get an indication of the quantities the inertial sensors should be measuring, 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.

In **Fig. 3(b)** the forces acting on the moving pendulum are represented. A friction force,  $F_f$ , is represented with its direction opposite to the direction of the instantaneous velocity of the pendulum, accounting for all kinds of friction inherent to the pendulum's motion.

The sum of all forces acting on the pendulum induces an acceleration which characterizes the motion equation of the pendulum. From this motion equation, the acceleration components along the  $x$  and  $z$  axis, as illustrated in **Fig. 3**, can be written as

$$a_x = -\|g\| \sin \theta - \frac{\|F_f\|}{M} \text{sgn}(v) \quad (2)$$

$$a_z = \frac{\|T\|}{M} - \|g\| \cos \theta = \frac{v^2}{R}. \quad (3)$$

In these equations,  $\text{sgn}(\cdot)$  is the sign function, given by

$$\text{sgn}(v) = \begin{cases} +1, & v \geq 0 \\ -1, & v < 0. \end{cases} \quad (4)$$

The accelerometers measure the acceleration sensed by a proof mass internal to the measuring unit which in turn is attached to the pendulum. This means that the measured accelerations are caused by forces acting on the measuring unit's case, but not on the proof mass. In this particular scenario, since the gravity force acts both on the proof mass and on the case, the accelerometers only measure the accelerations caused by the other forces: the tension,  $T$ , and the friction force,  $F_f$ . The measured accelerations along the  $x$  and  $z$  axis,  $\tilde{a}_x$  and  $\tilde{a}_z$ , are given by

$$\begin{aligned} \tilde{a}_x &= -\frac{\|F_f\|}{M} \text{sgn}(v) = a_x + \|g\| \sin \theta \\ &= \alpha R + \|g\| \sin \theta \end{aligned} \quad (5)$$

$$\begin{aligned} \tilde{a}_z &= \frac{\|T\|}{M} = \frac{v^2}{R} + \|g\| \cos \theta \\ &= \omega^2 R + \|g\| \cos \theta \end{aligned} \quad (6)$$

where  $\omega$  and  $\alpha$  represent the angular velocity and angular acceleration of the pendulum.

The values for  $\theta$ ,  $\omega$  and  $\alpha$  are measured by the encoder readings, and its derivatives. The measurements of the rate gyros are the components of the angular velocity of the pendulum. This means that the only rate gyro with a non-zero measurement should be the one oriented perpendicular to the plane of motion. Using **Fig. 3** as a reference, only the rate gyro along the  $y$  axis should measure a non-zero quantity, i.e.

$$\tilde{\omega} = \begin{bmatrix} 0 \\ \tilde{\omega}_y \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{d\theta}{dt} \\ 0 \end{bmatrix}. \quad (7)$$

By attaching the measuring unit to the pendulum in three different orthogonal orientations, sufficient data can be collected to calibrate the three accelerometers and the three rate gyros of the inertial measuring unit. The procedure consists in determining the nine scale factors,  $s_{kl}$ , and the three biases,  $b_k$ , of the sensor model described in Eq. (1). Rewriting the system of Eq. (1) as a function of the unknowns  $s_{kl}$  and  $b_k$ . The resulting system of equations is given by

$$z_o = \mathbf{A}M' \quad (8)$$

$$= \begin{bmatrix} z_{i,x} & 0 & 0 \\ z_{i,y} & 0 & 0 \\ z_{i,z} & 0 & 0 \\ 0 & z_{i,x} & 0 \\ 0 & z_{i,y} & 0 \\ 0 & z_{i,z} & 0 \\ 0 & 0 & z_{i,x} \\ 0 & 0 & z_{i,y} \\ 0 & 0 & z_{i,z} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} s_{xx} \\ s_{xy} \\ s_{xz} \\ s_{yx} \\ s_{yy} \\ s_{yz} \\ s_{zx} \\ s_{zy} \\ s_{zz} \\ b_x \\ b_y \\ b_z \end{bmatrix}$$

where  $M'$  is the vector with the twelve parameters to be determined.

Each measurement provides three equations as can be seen in Eq. (8). The sensor inputs,  $z_i$ , are known by feeding the encoder readings, and its derivatives, into Eqs. (5), (6) and (7); the sensor outputs,  $z_o$ , are directly measured. Only the twelve parameters in vector  $M'$  are unknown. To obtain a solution for  $M'$ , at least four measurements have to be known, but since the measurements are disturbed by random noise, a much bigger set of measurements should be used.

A least squares solution can be obtained for the parameters, by using Eq. (9):

$$M' = \mathbf{A}^\dagger z_o \quad (9)$$

where  $\mathbf{A}^\dagger$  denotes the pseudo-inverse of matrix  $\mathbf{A}$  obtained through the use of the singular value decomposition [13].

It should be noted that two systems of equations have to be solved: one to determine the parameters of the accelerometers, and another to determine the parameters of the rate gyros.

### 2.3 Temperature dependence

A well-known fact is that inertial sensor parameters are temperature dependent. This model does not account for that, and usually there is a non-linear relation between the parameters and the temperature, which can be different for each of the individual sensors. The proposed solution, being able to cope with different working temperatures, is to build a lookup table containing the parameters for several working temperatures, and then to determine the appropriate parameters for the current temperature by interpolating the contents of the table.

## 3. Data from Camera Sensor

The camera used in the vision system is modeled as pinhole model. The pinhole camera model derives from the

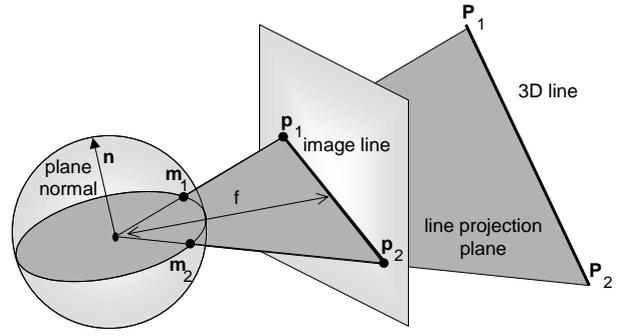


Fig. 4 Line projection onto the unit sphere

geometry of the camera, and considers the projection of world points onto a plane, but the projection need not be onto a plane. Consider a unit sphere around the optical center, with the images being formed on its surface. The image plane can be seen as a plane tangent to a *virtual sphere* of radius  $f$ , concentric with the unit sphere, as shown in Fig. 4. The distance  $f$  is the focal distance of the camera. The image plane touches the *virtual sphere* at the equator, and at this point the origin of the image plane coordinate system, the image center, is defined.

Using the unit sphere gives a more general model for central perspective and provides an intuitive visualization of projective geometry [8]. It also has numerical advantages when considering points at infinity, such as vanishing points.

A world point  $P_i$  will project on the image plane as  $p_i$  and can be represented by the unit vector  $m_i$  placed at the center of the sphere, the optical center of the camera. With image centered coordinates  $p_i = (u_i, v_i)$  we have

$$P_i \rightarrow m_i = \frac{P_i}{\|P_i\|} = \frac{1}{\sqrt{u_i^2 + v_i^2 + f^2}} \begin{bmatrix} u_i \\ v_i \\ f \end{bmatrix}. \quad (10)$$

To avoid ambiguity,  $m_i$  is forced to be positive, so that only the points on the image side hemisphere are considered.

Image lines can also be represented in a similar way. Any image line defines a plane with the center of projection, as shown in Fig. 4. A vector  $n$  normal to this plane uniquely defines the image line and can be used to represent the line.

For a given image line  $au + bv + c = 0$ , the unit vector is given by

$$n = \frac{1}{\sqrt{a^2 + b^2 + (c/f)^2}} \begin{bmatrix} a \\ b \\ c/f \end{bmatrix}. \quad (11)$$

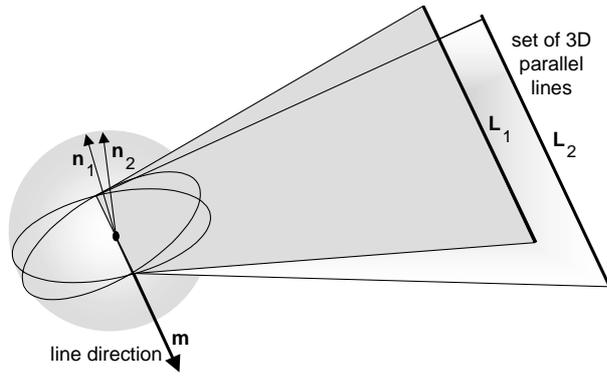
As seen in Fig. 4, we can write the unit vector of an image line with points  $m_1$  and  $m_2$  as

$$n = m_1 \times m_2. \quad (12)$$

### 3.1 Vanishing points

Parallel lines only meet at infinity, but in the image plane, the point where they meet can be quite visible and is called the *vanishing point* of that set of parallel lines.

A space line with the orientation of a unit vector  $m$  has, when projected, a *vanishing point* with unit sphere vector



**Fig. 5** Vanishing point of a set of 3D parallel lines

$\pm \mathbf{m}$ , as shown in **Fig. 5**. Since the vanishing point is only determined by the 3D orientation of the space line, projections of parallel space lines intersect at a common vanishing point.

As seen in Fig. 5, the normals to the line projection planes will all lie in the same plane, orthogonal to the vanishing point  $\mathbf{m}$ . The vanishing point of a set of 3D parallel lines with image lines  $\mathbf{n}_1$  and  $\mathbf{n}_2$  is given by

$$\mathbf{m} = \mathbf{n}_1 \times \mathbf{n}_2. \quad (13)$$

#### 4. Rotation Between Camera and IMU

In order to determine the rigid transformation 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. 1. When the IMU sensed acceleration is equal in magnitude to the gravity, the sensed direction is the vertical. For the camera, either using a specific calibration target, such as a chessboard placed vertically, or assuming the scene has enough predominant vertical edges, the vertical direction can be taken from the corresponding vanishing point.

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 Horn's method [9]. 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  $\hat{\mathbf{q}}$  that rotates inertial measurements in the inertial sensor frame of reference  $\{\mathcal{I}\}$  to the camera frame of reference  $\{\mathcal{C}\}$ . In the following equations, when multiplying vectors with quaternions, the corresponding imaginary quaternions are implied. We want to find the unit quaternion  $\hat{\mathbf{q}}$  that maximizes

$$\sum_{i=1}^n (\hat{\mathbf{q}} {}^{\mathcal{I}}\mathbf{v}_i \hat{\mathbf{q}}^*) \cdot {}^{\mathcal{C}}\mathbf{v}_i \quad (14)$$

which can be rewritten as

$$\sum_{i=1}^n (\hat{\mathbf{q}} {}^{\mathcal{I}}\mathbf{v}_i) \cdot ({}^{\mathcal{C}}\mathbf{v}_i \hat{\mathbf{q}}). \quad (15)$$

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

$$\hat{\mathbf{q}} {}^{\mathcal{I}}\mathbf{v}_i = \begin{bmatrix} 0 & -{}^{\mathcal{I}}x_i & -{}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}z_i \\ {}^{\mathcal{I}}x_i & 0 & {}^{\mathcal{I}}z_i & -{}^{\mathcal{I}}y_i \\ {}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}z_i & 0 & {}^{\mathcal{I}}x_i \\ {}^{\mathcal{I}}z_i & {}^{\mathcal{I}}y_i & -{}^{\mathcal{I}}x_i & 0 \end{bmatrix} \hat{\mathbf{q}} = {}^{\mathcal{I}}\mathbf{V}_i \hat{\mathbf{q}} \quad (16)$$

and

$${}^{\mathcal{C}}\mathbf{v}_i \hat{\mathbf{q}} = \begin{bmatrix} 0 & -{}^{\mathcal{C}}x_i & -{}^{\mathcal{C}}y_i & -{}^{\mathcal{C}}z_i \\ {}^{\mathcal{C}}x_i & 0 & -{}^{\mathcal{C}}z_i & {}^{\mathcal{C}}y_i \\ {}^{\mathcal{C}}y_i & {}^{\mathcal{C}}z_i & 0 & -{}^{\mathcal{C}}x_i \\ {}^{\mathcal{C}}z_i & -{}^{\mathcal{C}}y_i & {}^{\mathcal{C}}x_i & 0 \end{bmatrix} \hat{\mathbf{q}} = {}^{\mathcal{C}}\mathbf{V}_i \hat{\mathbf{q}}. \quad (17)$$

Substituting in Eq. (15)

$$\sum_{i=1}^n ({}^{\mathcal{I}}\mathbf{V}_i \hat{\mathbf{q}}) \cdot ({}^{\mathcal{C}}\mathbf{V}_i \hat{\mathbf{q}}) \quad (18)$$

or

$$\sum_{i=1}^n \hat{\mathbf{q}}^T {}^{\mathcal{I}}\mathbf{V}_i^T {}^{\mathcal{C}}\mathbf{V}_i \hat{\mathbf{q}} \quad (19)$$

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

$$\hat{\mathbf{q}}^T \left( \sum_{i=1}^n {}^{\mathcal{I}}\mathbf{V}_i^T {}^{\mathcal{C}}\mathbf{V}_i \right) \hat{\mathbf{q}}. \quad (20)$$

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

$$\max \hat{\mathbf{q}}^T \mathbf{N} \hat{\mathbf{q}} \quad (21)$$

where  $\mathbf{N} = \sum_{i=1}^n {}^{\mathcal{I}}\mathbf{V}_i^T {}^{\mathcal{C}}\mathbf{V}_i$ . Having

$$S_{xx} = \sum_{i=1}^n {}^{\mathcal{I}}x_i {}^{\mathcal{C}}x_i, \quad S_{xy} = \sum_{i=1}^n {}^{\mathcal{I}}x_i {}^{\mathcal{C}}y_i \quad (22)$$

and analogously for all 9 pairings of the components of the two vectors, matrix  $\mathbf{N}$  can be expressed using these sums as in Eq. (23). 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  $\lambda_{\max}$  of  $\mathbf{N}$ —see [9] for details.

#### 4.1 Measurement span for rotation estimation

The above method finds the rotation that maximizes the alignment of the rotated inertial frame verticals with the camera observed verticals expressed by Eq. (14).

The inertial frame verticals,  ${}^{\mathcal{I}}\mathbf{v}_i$ , are easily obtained. The only restriction is that the system has to be motionless, or subject to constant speed, so that the accelerometers give the direction of the gravity vector  $\mathbf{g}$ , i.e.,

$${}^{\mathcal{I}}\mathbf{v}_i = -\frac{\mathbf{g}}{\|\mathbf{g}\|} = \frac{1}{\sqrt{a_x^2 + a_y^2 + a_z^2}} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (24)$$

$$N = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix} \quad (23)$$

where  $a_x, a_y$  and  $a_z$  are the calibrated accelerometer measurements along each axis. Notice that the accelerometer will measure the reactive (upward) force to gravity.

The camera frame verticals,  ${}^c v_i$ , 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 chessboard 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.

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.

### 5. Tests and Results

The tests were performed using a DMU-FOG inertial unit from Crossbow Technology coupled with a Sony XC-999 CCD video camera.

#### 5.1 Inertial sensor calibration

The inertial sensors were calibrated using the method described in Section 2. The inertial unit was attached to the pendulum in three distinct orientations in order to obtain a significant set of measurements for each sensor and the correspondent position of the pendulum. The outputs of the sensor were registered in time and the angular position of the pendulum is given by the absolute encoder measurements. These measurements were used with Eqs. (5), (6) and (7).

Since the inertial measurement unit used in this work is a medium-grade unit, the manufacturer supplies an individual calibration table which can be used as a ground truth to evaluate our calibration procedure.

**Table 1** presents the parameters supplied by the manufacturer and compares them with the ones obtained using the calibration method described in this paper. It should be noted that in the table, the sensitivity is compared in (g/V) and (deg/sec/V), which are the inverses of the scale factors,  $s_{kk}$ , as defined in Eq. (1).

In order to evaluate the temperature dependence of the sensors parameters, Table 1 presents the obtained parameters for two different temperatures. The internal temperature of the inertial unit stabilizes after some time (from five to ten minutes) and only after that time were the calibration tests performed, in order that all the data be obtained at the same constant temperature. Since the manufacturer only presents the calibration parameters for an internal temper-

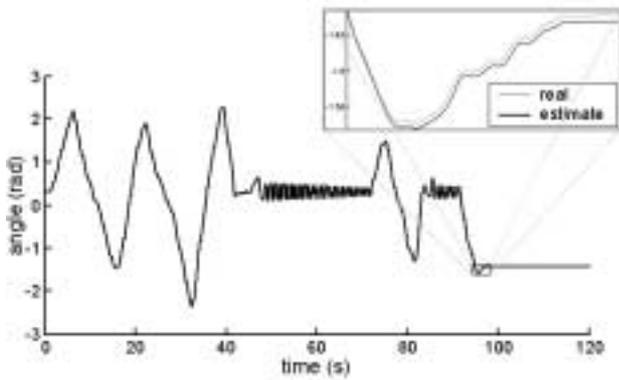
**Table 1** Comparison of the obtained inertial sensors' parameters at two different temperatures with the ones supplied by the manufacturer

Accelerometers			
	X	Y	Z
<i>Sensitivity (g/V)</i>			
Manuf. Supplied (29.82°C)	1.008	1.000	1.017
Obtained (29.68°C)	1.015	1.026	1.022
Obtained (24.45°C)	0.999	1.027	1.030
<i>Null Offset (V)</i>			
Manuf. Supplied (29.82°C)	2.485	2.519	2.455
Obtained (29.68°C)	2.539	2.514	2.456
Obtained (24.45°C)	2.526	2.510	2.446
Rate Gyros			
	X	Y	Z
<i>Sensitivity (deg/sec/V)</i>			
Manuf. Supplied (29.82°C)	102.731	101.643	102.388
Obtained (29.68°C)	102.202	102.085	102.216
Obtained (24.45°C)	102.115	102.155	102.054
<i>Null Offset (V)</i>			
Manuf. Supplied (29.82°C)	2.499	2.499	2.499
Obtained (29.68°C)	2.500	2.500	2.499
Obtained (24.45°C)	2.502	2.500	2.500

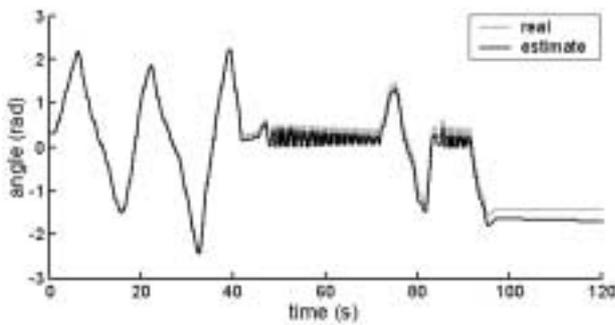
ature of 29.82°C, these should only be compared with the ones obtained at a similar internal temperature (29.68°C), which was the stabilized internal temperature of the unit when the room temperature was around 22°C. Regarding parameter variations with temperature, one can easily observe that these variations differ for each individual sensor; considering also that the stabilized internal temperature of the unit varies slightly for normal operation conditions, a lookup table for the parameters can be a simple and straightforward solution to compensate for temperature variations.

The manufacturer does not present any parameters relating to axis alignment in their unit. However, from the results of our method the system exhibits a mean cross-axis sensitivity of about 0.6%. These small cross-axis errors can cause high drifts over time if the inertial data measurements are to be used to estimate position, by integrating in time the data of the sensor.

To demonstrate the effects of this cross-axis sensitivity effect and the differences between using our calibration or the manufacturer's calibration data, a test was performed where the pendulum swung for some time with the internal temperature of the unit close to what is specified in the manufacturer's calibration sheet. During the experiment, the motion of the pendulum was sometimes forced, and



**Fig. 6** Results for the integration of inertial data calibrated by our method



**Fig. 7** Results for the integration of inertial data calibrated by manufacturer

otherwise the pendulum was left oscillating freely. The data of the sensors were recorded and afterwards the rate gyros outputs were integrated over time in order to obtain the angle of the pendulum.

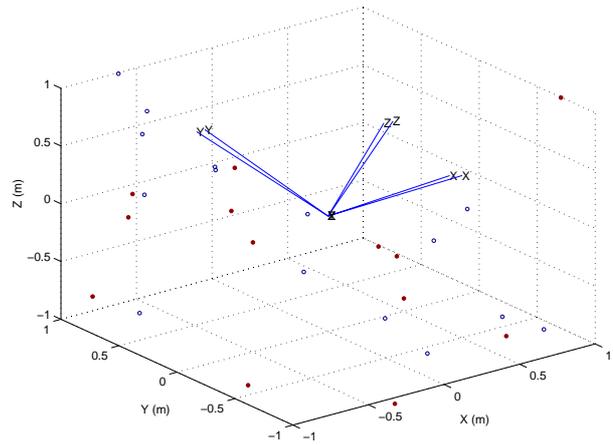
**Figure 6** presents the results obtained by the simple integration of the inertial data after being calibrated with the parameters obtained by us for the testing temperature. The inability to distinguish between the two curves shows that the real angle, obtained by the encoder readings, only has slight differences from the one obtained by the integration of the rate gyro output.

In **Fig. 7**, the results for the same experiment are presented, but this time using the manufacturer’s supplied calibration sheet to calibrate the inertial data. As can be easily seen, the estimation has a significant drift, much higher than the drift achieved when using the calibration data obtained by the procedure in this paper.

These results proved themselves satisfactory, and good enough for many mobile robotic applications. The calibration procedure presented has been able to reduce drastically the drift obtained by the integration of inertial data, by determining the sensors’ parameters with a reasonably high accuracy.

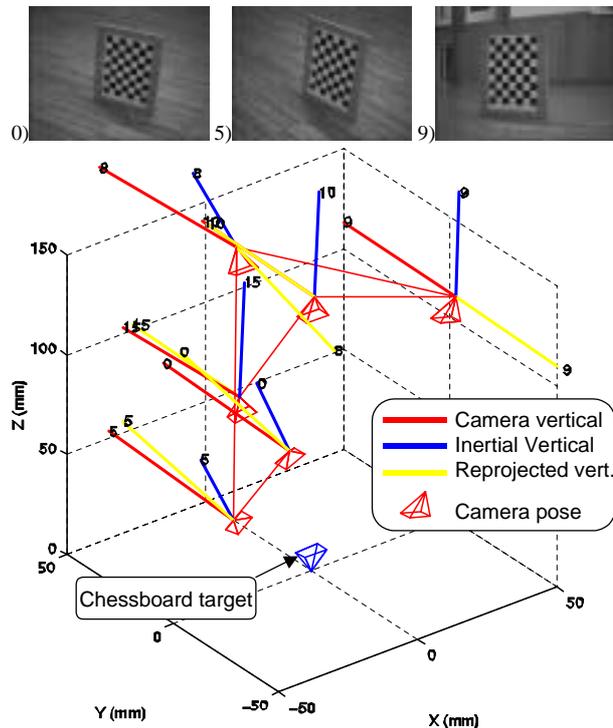
### 5.2 Rotation estimation

To validate the above method, a simulation was performed, using a known rotation with added noise. **Figure 8** shows the data set with added noise used, the known rotation, and the estimated rotation. The adjoining table indicates the estimated quaternions and the angular errors.



20 random positions, rotated with added noise		
real	0.98079 < 0.18110, 0.040244, 0.060366 >	error (deg)
SNR=5	0.97568 < 0.18435, 0.065185, 0.099079 >	5.31°
SNR=10	0.97702 < 0.19855, 0.062037, 0.046521 >	3.57°
SNR=20	0.98476 < 0.15888, 0.041402, 0.057394 >	2.58°
SNR=40	0.98072 < 0.18157, 0.040012, 0.060126 >	0.07°

**Fig. 8** Plot of simulation data for 20 random positions, filled markers indicate initial points, and circles the same points after rotation (22.5° about axis [0.9, 0.2, 0.3]) and with added white Gaussian noise SNR=5; the plotted axes indicate the known and estimated rotation quaternions; the table shows a set of simulation results, indicating angular error of the estimated rotations for different noise values



**Fig. 9** Some of the images used and result obtained, showing camera pose, camera sensed verticals, inertial sensed verticals and inertial verticals re-projected to camera frame of reference

With improving signal to noise ratio, the estimated rotation approaches the real value.

The camera calibration toolbox provided by Intel Open Source Computer Vision Library [10] was used to provide

**Table 2** Rotation estimation error: table shows how the mean angular error for the re-projection of the complete set shown in Fig. 10 varies for increasing measurements used in computing the unknown rotation

$n$	1	1:2	1:3	1:4	1:5	1:6
error	46.71°	1.854°	2.683°	2.410°	2.384°	1.784°
$n$	1:7	1:8	1:9	1:10	1:12	1:14
error	1.660°	1.628°	1.666°	1.655°	1.627°	1.610°

**Table 3** Rotation estimation error after outlier removal

frames	1	1:2	1:4	1:5	1:6	1:7
error	43.45°	1.587°	1.487°	1.464°	1.365°	1.414°
frames	1:8	1:9	1:10	1:11	1:12	1:14
error	1.473°	1.303°	1.307°	1.314°	1.305°	1.312°

a standard camera calibration method. The calibration used images of a chessboard target in several positions and recovers the intrinsic parameters of the camera, 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 [11].

In a test sequence, the camera was moved through several poses with the vertical chessboard target in sight, and all IMU data and images were logged. The camera calibration was performed with images sampled from the complete set recorded. **Figure 9** shows some of the images used and the reconstructed camera positions.

Having calibrated the camera, the vertical vanishing point was determined for each image, providing a set of measurements  ${}^c v_i$ . Having the corresponding  ${}^I v_i$  from the inertial sensors, the estimation method was applied to the data set. Figure 9 shows the result obtained for a real data set.

The estimated rotation has an angle  $91.25^\circ$  about an axis  $(0.89, -0.27, -0.3582)$ , and is about the expected one, given the mechanical mount, of a near right angle approximately about the  $x$  axis. Re-projecting the inertial sensor data showed consistency of the method. The mean-square error in the re-projected verticals was  $1.570^\circ$ .

**Figure 10** shows another test, where 14 observations were made. The unit sphere is shown for each frame, with the projected image, the vanishing point construction, the IMU measured vertical and its re-projection to the camera frame. The mean-square error in the re-projected verticals was  $1.312^\circ$ .

**Table 2** shows the mean error of the 14 re-projected verticals for different sequential frame sets used in computing the unknown rotation. The table shows that the data from frame 002 is a clear outlier from the rest, raising the error significantly. **Table 3** shows the error after outlier removal.

## 6. Conclusions

In order to use off-the-shelf inertial sensors and cameras for computer vision, appropriate modeling and calibration techniques are required. This article presents a technique for modeling and calibrating a camera with integrated low-cost inertial sensors.

Using a pendulum with an encoded shaft, inertial sen-

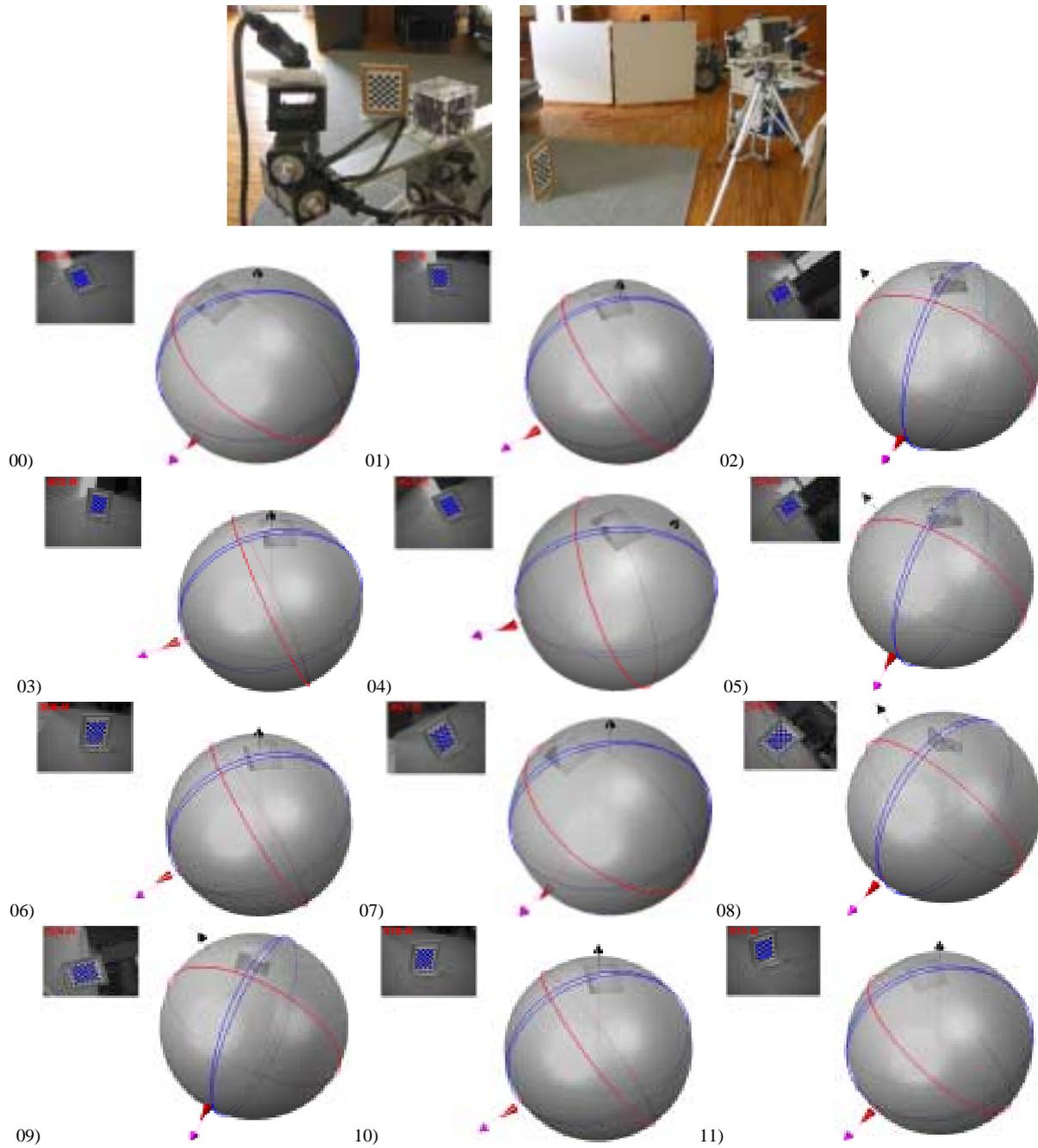
sor alignment, bias and scale factor can be estimated, for both accelerometers and gyros. With the inertial sensors rigidly fixed to the camera, the rotation between the two frames of reference can be found by moving the system and observing the vertical direction with both sensors. The inertial sensors when static only sense gravity, providing a vertical reference. Sets of parallel vertical edges provide the vertical vanishing point, giving the vertical direction in the camera frame of reference. The two sets of measurements allow the estimation of the rotation between the sensors. Knowing this rotation, the inertial sensor data can be mapped to the camera frame of reference, and used in image processing tasks.

## Acknowledgments

The authors would like to thank the support provided to the development of this work, by the CyberCars project [1] of the EC-Information Society Technologies (IST) Programme and CyberMove [2].

## References

- [1] *CyberCars—Cybernetic Technologies for the Car in the City*, <http://www.cybercars.org>.
- [2] *CyberMove—Cybernetic Transportation Systems for Cities of Tomorrow*, <http://www.cybermove.org>.
- [3] J. Lobo and J. Dias, "Integration of inertial information with vision towards robot autonomy," in *Proc. of the 1997 IEEE International Symposium on Industrial Electronics*, Guimaraes, Portugal, July 1997, pp. 825–830.
- [4] J. Lobo and J. Dias, "Ground plane detection using visual and inertial data fusion," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, Canada, October 1998, pp. 912–917.
- [5] J. Lobo and J. Dias, "Fusing of image and inertial sensing for camera calibration," in *Proc. of the International Conference on Multi-sensor Fusion and Integration for Intelligent Systems (MFI 2001)*, Baden-Baden, Germany, August 2001, pp. 103–108.
- [6] J. Lobo, C. Queiroz, and J. Dias, "World feature detection and mapping using stereovision and inertial sensors," *Robotics and Autonomous Systems Journal*, vol. 44, no. 1, pp. 69–81, July 2003.
- [7] R. Hartley, "In defense of the 8-point algorithm," in *Proc. of the International Conference on Computer Vision*, Los Alamitos, CA, Year?, pp. 1064–1070.
- [8] K. Kanatani, *Geometric Computation for Machine Vision*. Oxford, UK: Oxford University Press, 1993.
- [9] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, vol. 4, no. 4, pp. 629–642, April 1987.
- [10] Intel Microprocessor Research Laboratories, "Intel Open Source Computer Vision Library," <http://www.intel.com/research/mrl/research/opencv/>, 2003.
- [11] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. of the Seventh International Conference on Computer Vision (ICCV'99)*, Kerkyra, Greece, September 1999, pp. 666–673.
- [12] J. Lobo, "Inertial sensor data integration in computer vision systems," M.S. thesis, University of Coimbra, April 2002.
- [13] S. J. Leon, *Linear Algebra With Applications*, 5th edition. Englewood Cliffs: NJ: Prentice Hall, 1998.



**Fig. 10** Experimental setup and results for some of the 14 frames used, where the unit sphere is shown for each frame, with the projected image, the vanishing point construction, the IMU measured vertical and its re-projection to the camera frame

## Biographies

---

**João Alves** (*João Pedro Paiva Alves*) was born on the 15th of December, 1978, in Portugal. He graduated in Electrical and Computer Engineering from the University of Coimbra in October 2001 and is since then pursuing a Masters degree at the same institution. He has initiated its research activity from August to September 1999 in the Institute for Systems and Robotics, Coimbra, having conducted his research at that institute since then. Since January 2002 he has been a R&D technician at the Institute Pedro Nunes ([www.ipn.pt](http://www.ipn.pt)). His research interests include Mobile Robotics, Computer Architectures, Real-Time Systems, Distributed Computer Systems and Computer Graphics.

**Jorge Lobo** (*Jorge Nuno de Almeida e Sousa Almada Lobo*) was born on the 23rd of September, 1971, in Cambridge, UK. In 1995 he completed his 5 year course in Electrical Engineering at Coimbra University. He was a junior teacher at the Computer Science Department of the Coimbra Polytechnic School, and later joined the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra where he currently works as a teaching assistant. In April 2002 he completed his M.Sc. with the thesis "Inertial Sensor Data Integration in Computer Vision Systems". His current research is carried out at the Institute of Systems and Robotics, University of Coimbra, and is aimed at the fusion of inertial information with vision systems in mobile robots.

**Jorge Dias** (*Jorge Manuel Miranda Dias*) has a Ph.D. on Electrical Engineering by the University of Coimbra, specialization in Control and Instrumentation, November 1994. His main research area is Computer Vision and Robotics, with activities and contributions on the field since 1984. Jorge Dias has been exploring different topics on Computer Vision and Mobile Robotics towards the improvement of autonomous robotic systems. He was main researcher from several projects financed by the Portuguese institution JNIC—Junta Nacional de Investigação Científica, FCT—Portuguese Foundation for Science and Technology and projects, financed by European Community and NATO—Science For Stability Program.