

Sensor-Based Self-Calibration of the iCub's Head

José Santos
Instituto de Sistemas e Robótica
Instituto Superior Técnico
jffsantos@isr.ist.utl.pt

Alexandre Bernardino
Instituto de Sistemas e Robótica
Instituto Superior Técnico
alex@isr.ist.utl.pt

José Santos-Victor
Instituto de Sistemas e Robótica
Instituto Superior Técnico
jasv@isr.ist.utl.pt

Abstract—In this paper we propose techniques for the calibration of the iCub's stereo head using vision and inertial measurements. Given that wear and tear can change the geometrical relationship between the different elements in the kinematic chain, new calibrations must be performed periodically. We propose methods that allow automatic calibration without the need for using external sensors or specially designed calibration objects. The methods can be applied at any time during the operation of the system, thus being an alternative for systems whose calibrations are imprecise or that require frequent recalibration. Results are shown both in simulations and on the iCub's stereo head.

I. INTRODUCTION

The calibration of the geometric relationships between the actuation and sensing devices of a robotic system is critical for operations which require precise measurements and control. Classical calibration methods require precise mechanical adjustments and parameter tuning that is typically done by experts. With the growing complexity of robotic devices, this can be an arduous task. The development of automated and self-calibration methods will significantly impact on the usability and ease of maintenance of such systems. More critical is the fact that classical calibration methods often rely on sensors that are different from the ones which will be used in the applications. For instance, the 3D reconstruction of world points by the means of a mobile stereo head requires the precise knowledge of the cameras positioning with respect to a certain world reference frame but their calibration rely on limit switches or mechanical hard stops. The relationship between these sensors (cameras, inertial units) is mechanically adjusted once (at assembly time). Besides being prone to errors, this initial calibration unavoidably changes with time due to wear, vibrations and other operating conditions.

The availability of multiple sensors in modern robots opens opportunities for the development of sensor based calibration methods. Such methods rely on prospective movements executed by the robot to self-calibrate the relationships between its sensors and actuators. As an outcome, the estimation of the unknown parameters is often possible to obtain.

These techniques have been applied to several types of robots and applications such as: the calibration of the extrinsic parameters of a camera system mounted on top of a mobile robot [9] and the calibration of manipulator camera systems [12], [11]. However, most of the previous work

on self-calibration still rely on prior knowledge about the environment, e.g. the use of easily detectable objects (markers, lights, etc). Few works address the problem without constraining somehow the scenario. These are found mostly on the field of mobile robot kinematics calibration from laser scans [14], [10], and vision-based alignment of stereo heads [6], [7] and pan-tilt structures [16]. To the best of our knowledge there are no methods using inertial measurements for the calibration of a humanoid head.

In this paper we use an inertial sensor emulating the vestibular sense and the cameras of the stereo setup to calibrate and initialize the neck pan-tilt-swing angles of the iCub's head and the eyes pan-tilt angles (see Fig. II). Despite only showing an application to a humanoid stereo head, the methods can easily be adapted to serial rotation joints orthogonally arranged. To calibrate the neck joints, we propose two different approaches: the first based on the Newton's Method for non-linear equations and the other based on the Broyden's method [2]. Their relative efficiency is discussed and the one which presented the best experimental results was implemented on the iCub's head.

To calibrate the eyes pan-tilt degrees of freedom we use a method inspired in [6]. By performing prospective motions on an axis and using prior knowledge of the kinematic chain, the observed image patterns only depend on the alignment between axes. We have designed a robust version of such a method in [17] and here present an extended experimental analysis of results obtained on the iCub robotic platform.

The paper is organized as follows. Section II presents a description of iCub's head and eyes kinematics. Section III discusses two different approaches for calibrating iCub's head. In section IV we present a methodology to calibrate the eyes pan-tilt degrees of freedom. Results are presented in section V and conclusions in section VI.

II. HEAD AND EYES KINEMATICS

The humanoid head system considered in this work has six degrees of freedom: neck pitch (θ_t), neck roll (θ_s), neck yaw (θ_p), eyes tilt (θ_v) and eyes pan (θ_h), as shown in Figure II.

It is important to establish a reference coordinate frame, which will be denoted by $\{0\}$. This reference coordinate frame corresponds to the body coordinate frame and is defined by the local vertical and by the rotation of the body along this axis. Considering identical reference coordinate frames for all joints in the canonical state, the rotation matrix

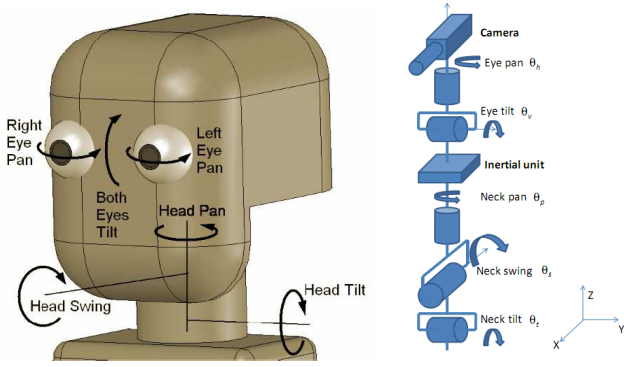


Fig. 1. Illustration of the iCub's head degrees of freedom.

representing the head's orientation with respect to the body reference frame depends on the tilt, swing and pan angles, respectively θ_t , θ_s and θ_p , and is given by [1]:

$$\begin{aligned} {}^0\mathbf{R}_3 &= {}^0\mathbf{R}_1 \cdot {}^1\mathbf{R}_2 \cdot {}^2\mathbf{R}_3 & (1) \\ &= \text{ROT}_y(\theta_t) \cdot \text{ROT}_x(\theta_s) \cdot \text{ROT}_z(\theta_p) & (2) \\ &= \begin{bmatrix} c_t & 0 & s_t \\ 0 & 1 & 0 \\ -s_t & 0 & c_t \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_s & -s_s \\ 0 & s_s & c_s \end{bmatrix} \cdot \begin{bmatrix} c_p & -s_p & 0 \\ s_p & c_p & 0 \\ 0 & 0 & 1 \end{bmatrix} & (3) \end{aligned}$$

with $c_t = \cos(\theta_t)$, $s_t = \sin(\theta_t)$, $c_s = \cos(\theta_s)$, $s_s = \sin(\theta_s)$, $c_p = \cos(\theta_p)$ and $s_p = \sin(\theta_p)$.

Similarly, the rotation matrix representing each of the eyes' orientation with respect to the head fixed reference frame depends on the eye pan and tilt angles, respectively θ_h and θ_v , and is given by:

$$\begin{aligned} {}^3\mathbf{R}_5 &= {}^3\mathbf{R}_4 \cdot {}^4\mathbf{R}_5 & (4) \\ &= \text{ROT}_y(\theta_v) \cdot \text{ROT}_z(\theta_h) & (5) \\ &= \begin{bmatrix} c_v & 0 & s_v \\ 0 & 1 & 0 \\ -s_v & 0 & c_v \end{bmatrix} \cdot \begin{bmatrix} c_h & -s_h & 0 \\ s_h & c_h & 0 \\ 0 & 0 & 1 \end{bmatrix} & (6) \end{aligned}$$

with $c_v = \cos(\theta_v)$, $s_v = \sin(\theta_v)$, $c_h = \cos(\theta_h)$, and $s_h = \sin(\theta_h)$.

III. HEAD CALIBRATION

A. Problem Formulation

The robot head has an inertial sensor at the end of the pan joint. The inertial sensor unit is composed of accelerometers and rate-gyros for the three axis and a magnetometer to measure the azimuth with respect to a earth fixed co-ordinate system. The inertial unit is thus able to calculate the orientation between the sensor-fixed coordinate system, denoted by $\{S\}$, and the earth fixed co-ordinate system, denoted by $\{G\}$, which is defined as a right handed Cartesian coordinate system with:

- \hat{X} positive when pointing to the local magnetic North.
- \hat{Y} according to the right handed coordinates (West).
- \hat{Z} positive when pointing up.

Note that this co-ordinate system may not correspond to the body coordinate system (that only happens if the inertial sensor is placed facing North). Moreover, the sensor fixed co-ordinate system does not correspond to the head fixed co-ordinate system (above denoted by $\{3\}$), because of the way the sensor is installed in iCub's head. This two coordinate systems are related according to the following rotation matrix:

$${}^3\mathbf{R}_S = \text{ROT}_z(\pi) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, one can say that the inertial sensor outputs the following rotation matrix:

$${}^G\mathbf{R}_S = {}^G\mathbf{R}_0 \cdot {}^0\mathbf{R}_3 \cdot {}^3\mathbf{R}_S \quad (7)$$

As already mentioned, the angular displacements given by the motor encoders are measured with respect to the position in which the robot is turned on. Thus, using the information provided by the motor differential encoders (the joint angle displacements) and the information provided by the inertial sensor (${}^G\mathbf{R}_S$), our goal is to align iCub's head with its body. That is, iCub's head should be put in a position such that ${}^3\mathbf{R}_0 = I_3$.

B. Proposed Methodology

Considering equation 7 and having performed the required calculations, one can state that:

$$({}^G\mathbf{R}_S)_{31} = r_{31}^{GS} = s_t \cdot c_p - c_t \cdot s_s \cdot s_p \quad (8)$$

$$({}^G\mathbf{R}_S)_{32} = r_{32}^{GS} = -s_t \cdot s_p - c_t \cdot s_s \cdot c_p \quad (9)$$

The goal is to place the iCub's head in a position such that ${}^0\mathbf{R}_3 = I_3$. In such a position the following equalities must be verified:

$$r_{31}^{GS} = 0 \quad (10)$$

$$r_{32}^{GS} = 0 \quad (11)$$

It is important to note that, since ${}^G\mathbf{R}_S$ is a rotation matrix, if equations 10 and 11 are verified then the following equalities must also hold:

$$r_{13}^{GS} = 0 \quad (12)$$

$$r_{23}^{GS} = 0 \quad (13)$$

$$r_{33}^{GS} = 1 \quad (14)$$

In order to put the iCub's head in a position in which equations 10 and 11 hold, we only need to change the tilt and swing angular displacements. This problem is one of solving a system of nonlinear equations (the number of equality conditions is equal to the number of variables):

$$r(\theta_t, \theta_s) = \begin{bmatrix} r_{31}^{GS}(\theta_t, \theta_s) \\ r_{32}^{GS}(\theta_t, \theta_s) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (15)$$

Note that these constraints only align the head Z axis with gravity. The azimuth (pan angle) is still undetermined and will be computed with additional constraints.

A popular way to solve this kind of problems is to use Newton's method for nonlinear equations [2]. Newton's method defines a linear model $M_k(\Delta\theta)$ of $r(\theta + \Delta\theta)$ in the following way:

$$r(\theta + \Delta\theta) = r(\theta) + J(\theta) \cdot \Delta\theta \quad (16)$$

where $J(\theta)$ denotes the jacobian of r evaluated in θ .

Newton's method in its pure form chooses the step $\Delta\theta$ to be the vector for which $M_k(\Delta\theta) = 0$, that is:

$$\Delta\theta = -J(\theta)^{-1} \cdot r(\theta) \quad (17)$$

However, in this case, since r is not known, the information required for computing the jacobian is not given. As such, we propose two different strategies to compute the initial tilt and swing angular displacements, respectively: θ_t^0 and θ_s^0 .

1) *A Systematic Approach:* One way of solving this problem is to express r_{31}^{GS} and r_{32}^{GS} in the following way:

$$r_{31}^{GS}(\alpha, \beta) = \sin(\theta_t^0 + \alpha) \cos(\theta_p^0) - \cos(\theta_t^0 + \alpha) \sin(\theta_s^0 + \beta) \sin(\theta_p^0) \quad (18)$$

$$r_{32}^{GS}(\alpha, \beta) = -\sin(\theta_t^0 + \alpha) \sin(\theta_p^0) - \cos(\theta_t^0 + \alpha) \sin(\theta_s^0 + \beta) \cos(\theta_p^0) \quad (19)$$

where α and β are suitably chosen angle displacements of the tilt and swing joints. The above equations can be rewritten as expressed below:

$$r_{31}^{GS}(\alpha, \beta) = a_1 \sin(\alpha) + a_2 \cos(\alpha) + a_3 \sin(\alpha) \sin(\beta) + a_4 \sin(\alpha) \cos(\beta) + a_5 \cos(\alpha) \sin(\beta) + a_6 \cos(\alpha) \cos(\beta) \quad (20)$$

$$r_{32}^{GS}(\alpha, \beta) = b_1 \sin(\alpha) + b_2 \cos(\alpha) + b_3 \sin(\alpha) \sin(\beta) + b_4 \sin(\alpha) \cos(\beta) + b_5 \cos(\alpha) \sin(\beta) + b_6 \cos(\alpha) \cos(\beta) \quad (21)$$

Since the values of a_1, \dots, a_6 and b_1, \dots, b_6 depend on the values of θ_t^0 , θ_s^0 and θ_p^0 , they are unknowns. However, in order to determine their values, one has simply to collect the values of r_{31}^{GS} and r_{32}^{GS} in six different points $(\alpha_1, \beta_1), \dots, (\alpha_6, \beta_6)$ and then solve the linear system of equations obtained by writing equations 20 and 21 for each one of these points. In practice, since the information provided by the inertial sensor is affected by noise, one should collect more than six points and then use the Pseudo-Inverse matrix method.

After rewriting equation 15 in terms of α and β , one obtains:

$$r(\alpha, \beta) = \begin{bmatrix} r_{31}^{GS}(\alpha, \beta) \\ r_{32}^{GS}(\alpha, \beta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (22)$$

which is known (as explained: this amounts to solve a system of linear equations). So Newton's method for nonlinear equations can now be applied to determine the angular displacements α^* and β^* which solve equation 22. Clearly, the initial tilt and swing angular displacements measured with respect to the body reference frame are given by:

$$\theta_t^0 = -\alpha^* \quad (23)$$

$$\theta_s^0 = -\beta^* \quad (24)$$

2) *An Incremental Approach:* Another way of addressing this problem consists in using Broyden's method [2]. Broyden's method is a secant method: it constructs its own approximation of the Jacobian, updating it at each iteration so that it mimics the behavior of the true Jacobian J over the step just taken.

The requirement that the approximate Jacobian should mimic the behavior of the true Jacobian can be specified as follows. Let s_k denote the step from θ_k to θ_{k+1} and let y_k denote the corresponding change in r , that is:

$$s_k = \theta_{k+1} - \theta_k \quad (25)$$

$$y_k = r(\theta_{k+1}) - r(\theta_k) \quad (26)$$

Broyden's method requires that the updated Jacobian approximation B_{k+1} to satisfy the following equation, which is known as the secant equation:

$$y_k = B_{k+1} \cdot s_k \quad (27)$$

The secant equation ensures that B_{k+1} and $J(x_{k+1})$ have similar behavior along direction s_k . Broyden's method corresponds to the following update:

$$B_{k+1} = B_k + \frac{(y_k - B_k \cdot s_k) \cdot s_k^T}{s_k^T \cdot s_k} \quad (28)$$

The Broyden update makes the smallest possible change to the Jacobian (as measured by the Euclidean norm: $\|B_k - B_{k+1}\|_2$) that is consistent with the secant equation, which can be formally stated as:

$$B_{k+1} \in \arg \min_{B: y_k = B \cdot s_k} \|B - B_k\| \quad (29)$$

The specification of the algorithm is presented below.

Algorithm 1 Broyden's Method

Choose θ_0 and a nonsingular initial Jacobian approximation B_0 ;

for $k = 0, 1, 2, \dots$ **do**

 Calculate a solution $\Delta\theta_k$ to the linear equations:

$$B_k \cdot \Delta\theta_k = -r(\theta_k)$$

$$\theta_{k+1} \leftarrow \theta_k + \Delta\theta_k$$

$$s_k \leftarrow \theta_{k+1} - \theta_k$$

$$y_k \leftarrow r(\theta_{k+1}) - r(\theta_k)$$

 Obtain B_{k+1} from formula 28

end for

After applying Broyden's algorithm it is reasonable to expect that the tilt and swing displacements are almost zero. So:

$$\theta_t^0 = -\sum_{k=1}^n \Delta\theta_t^k \quad (30)$$

$$\theta_s^0 = -\sum_{k=1}^n \Delta\theta_s^k \quad (31)$$

C. Computing the Initial Pan

Having applied one of these two approaches to compute θ_t^0 and θ_s^0 , one can easily compute the initial pan displacement θ_p^0 by means of equations 8 and 9:

$$s_p^0 = \frac{b \cdot r_{31}^0 - a \cdot r_{32}^0}{a^2 + b^2} \quad (32)$$

$$c_p^0 = \frac{r_{31}^0 - b \cdot s_p^0}{a} \quad (33)$$

with $a = s_t$ and $b = -c_t \cdot s_s$. Therefore:

$$\theta_p^0 = \text{atan2}(s_p^0, c_p^0) \quad (34)$$

Here, only the information corresponding to the initial position is being used in order to compute θ_p^0 . However, both the approaches presented in this work collect information corresponding to several positions while computing θ_t^0 and θ_s^0 . These data could be used to determine θ_p^0 more precisely, using, for instance, a Weighted Least Squares estimator [8]. Observe that positions closer to the zero present a greater signal-to-noise ratio and, thus, should be assigned smaller weights.

IV. EYE CALIBRATION

From the eye-head kinematics analysis, fixed points in the head fixed reference frame, at coordinates (X, Y, Z) can be expressed in the camera frame by:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = {}^3R_5 \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} c_v c_h X + s_h Y + s_v c_h Z \\ -c_v s_h X + c_h Y - s_v s_h Z \\ -s_v X + c_v Z \end{bmatrix}$$

The perspective projection of such points in the image have the following normalized coordinates:

$$x(\theta_v, \theta_h) = \frac{Y_c}{X_c} = \frac{-c_v s_h X + c_h Y - s_v s_h Z}{c_v c_h X + s_h Y + s_v c_h Z} \quad (35)$$

$$y(\theta_v, \theta_h) = \frac{Z_c}{X_c} = \frac{-s_v X + c_v Z}{c_v c_h X + s_h Y + s_v c_h Z} \quad (36)$$

Let us consider that, at start-up, the system has initial angles θ_v^0 and θ_h^0 . These angles are unknown when the system is turned on. Then, a prospective motion of the eye tilt unit is performed: the eye tilt angle is changed by θ_v . This process is illustrated in Fig. 2. For the sake of simplicity, and without loss of generality, we can consider a null initial eye tilt angle, $\theta_v^0 = 0$. This corresponds to set the inertial reference frame aligned with the initial robot's eye tilt frame.

In the above conditions, points observed by the camera at system start-up are at coordinates:

$$x_0 = \frac{-s_{h0}X + c_{h0}Y}{c_{h0}X + s_{h0}Y} \quad (37)$$

$$y_0 = \frac{Z}{c_{h0}X + s_{h0}Y} \quad (38)$$

where $s_{h0} = \sin(\theta_{h0})$ and $c_{h0} = \cos(\theta_{h0})$.

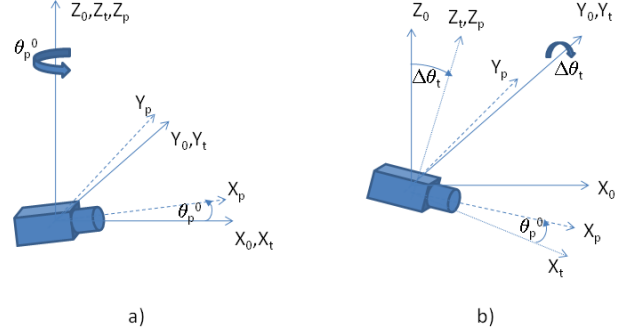


Fig. 2. The geometry of the system before (a) and after (b) the prospective motion.

After the eye tilt's prospective motion, these points move to the new coordinates:

$$x_1 = \frac{-c_v s_{h0}X + c_{h0}Y - s_v s_{h0}Z}{c_v c_{h0}X + s_{h0}Y + s_v c_{h0}Z} \quad (39)$$

$$y_1 = \frac{-s_v X + c_v Z}{c_v c_{h0}X + s_{h0}Y + s_v c_{h0}Z} \quad (40)$$

Let us recall that we are willing to estimate θ_{h0} , the unknown eye pan angle at start-up. θ_v is a known, actuated angle. The image coordinates x_0, y_0, x_1 and y_1 can be measured from the images by suitable image feature detectors and trackers. Additional, it is important to emphasize that X, Y and Z are unknown with respect to the head fixed reference frame. Thus, the goal is to eliminate these coordinates from the previous equations.

From equations (37) and (38) we can obtain the following constraints:

$$\frac{Y}{X} = \frac{x_0 c_{h0} + s_{h0}}{c_{h0} - s_{h0} x_0} \quad (41)$$

$$\frac{Z}{X} = \frac{y_0}{c_{h0} - s_{h0} x_0} \quad (42)$$

Now, dividing both the numerator and denominator of Eqs. (39) and (40) by X , and introducing the constraints in Eqs. (41) and (42), we obtain:

$$x_1 = \frac{-c_v s_{h0} c_{h0} + c_v s_{h0}^2 x_0 + c_{h0}^2 x_0 + c_{h0} s_{h0} - s_v s_{h0} y_0}{c_v c_{h0}^2 - c_v c_{h0} s_{h0} x_0 + s_{h0} c_{h0} x_0 + s_{h0}^2 + s_v c_{h0} y_0} \quad (43)$$

$$y_1 = \frac{-s_v c_{h0} + s_v s_{h0} x_0 + c_v y_0}{c_v c_{h0}^2 - c_v c_{h0} s_{h0} x_0 + s_{h0} c_{h0} x_0 + s_{h0}^2 + s_v c_{h0} y_0} \quad (44)$$

These equations can be rewritten in the homography form:

$$\begin{bmatrix} x_1 \\ y_1 \\ \lambda_1 \end{bmatrix} = H \cdot \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (45)$$

$$H = \begin{bmatrix} c_v s_{h0}^2 + c_{h0}^2 & -s_v s_{h0} & c_{h0} s_{h0} (1 - c_v) \\ s_v s_{h0} & c_v & -s_v c_{h0} \\ c_{h0} s_{h0} (1 - c_v) & s_v c_{h0} & c_v c_{h0}^2 + s_{h0}^2 \end{bmatrix} \quad (46)$$

A close inspection to the homography matrix shows that it has some repeated entries and only 6 of them are different.

It has the form:

$$H = \begin{bmatrix} h_1 & -h_2 & h_3 \\ h_2 & h_4 & -h_5 \\ h_3 & h_5 & h_6 \end{bmatrix} \quad (47)$$

For the estimation of the homography we need to extract a set of points visible on the first image (before the prospective motion) and track their location during the prospective motion:

$$\{(x_0^i, y_0^i) \mapsto (x_1^i, y_1^i), i = 1 \dots N\} \quad (48)$$

In the above equation, the lower index represents the image (0 for the image before and 1 for the image after the rotation), and the upper index represents the index of the point in the set, from 1 to N. We use the corner detector [15] and the sparse iterative version of Lucas-Kanade optical flow method, both implemented in the OpenCV library ¹.

For estimating the homography we use a linear method minimizing the algebraic error. We exploit the particular form of homography arising in our problem to estimate its parameters. For each point match obtained in the tracking procedure, we know from Eqs. (46) and (47) that:

$$x_1^i = \frac{h_1 x_0^i - h_2 y_0^i + h_3}{h_3 x_0^i + h_5 y_0^i + h_6} \quad (49)$$

$$y_1^i = \frac{h_2 x_0^i - h_4 y_0^i - h_5}{h_3 x_0^i + h_5 y_0^i + h_6} \quad (50)$$

Rearranging the previous equations we get the system:

$$\begin{cases} -x_0^i h_1 + y_0^i h_2 + x_0^i x_1^i h_3 - h_3 + y_0^i x_1^i h_5 + x_1^i h_6 = 0 \\ -x_0^i h_2 + x_0^i y_1^i h_3 + y_0^i h_4 + y_0^i y_1^i h_5 + h_5 + y_1^i h_6 = 0 \end{cases} \quad (51)$$

This can be put in vector form:

$$\begin{cases} a_x^i \cdot h = 0 \\ a_y^i \cdot h = 0 \end{cases} \quad (52)$$

with

$$\begin{aligned} h &= [h_1 \quad h_2 \quad h_3 \quad h_4 \quad h_5 \quad h_6]^T \\ a_x^i &= [-x_0^i \quad y_0^i \quad x_0^i x_1^i - 1 \quad 0 \quad y_0^i x_1^i \quad x_1^i] \\ a_y^i &= [0 \quad -x_0^i \quad x_0^i y_1^i \quad y_0^i \quad y_0^i y_1^i + 1 \quad y_1^i] \end{aligned}$$

Given a set of N corresponding points, we can form the following linear system of equations:

$$Ah = 0 \quad (53)$$

where:

$$A = [a_x^1 \quad a_y^1 \quad \dots \quad a_x^N \quad a_y^N]^T$$

Since the homography has 6 different entries we need at least 3 points to estimate it (each point contributes with two equations). We compute h through the Singular Value Decomposition (SVD) of A . From the SVD we take the right singular vector which corresponds to the smallest singular value. Finally we reshape the entries of h into the homography matrix H .

The above homography estimation method works well when there are no erroneous correspondences between the points in both images. Unfortunately, the tracking method sometimes provides false point matches that will degrade the results of the homography estimation. In order to address this problem, we use a well known robust estimation method that is able to eliminate the false matches (outliers) from the estimation process: the Random Sample Consensus (RANSAC) [4]. The application of this method in homography estimation problems is presented in [5].

Once the homography is estimated, we can compute the unknown angle θ_{h0} by, e.g.:

$$\theta_{h0} = \text{atan2}(h_2, h_5) \quad (54)$$

V. EXPERIMENTAL RESULTS

A. Head calibration

Both Newton's Method and Broyden's Method (as described in algorithm 1) were implemented in Matlab in order to assess the way each one converges when applied to the problem of aligning the iCub's head with its body. We assume that, initially, each joint angle (tilt, swing and pan) is reasonably close to 0 with respect to the body reference frame (every joint angle is assumed to be lower than 30°).

1) *Inertial Sensor Noise Characterization*: As was stated in section II, the inertial sensor outputs a rotation matrix, ${}^G R_S$, which describes the orientation of the sensor fixed coordinate system, $\{S\}$, with respect to the earth fixed coordinate system, $\{G\}$. Naturally, the information provided by this sensor includes noise. Therefore, in order to simulate it properly one needs to characterize the noise variance.

We have evaluated the sample variance of the rotation matrix provided by the inertial sensor in a wide range of positions. In each position we recorded 100 samples and then computed the sample variance. The maximum variance registered was 0.00334233. Hence, in each of the simulations presented in this section we shall assume a zero-mean gaussian white noise with variance 0.0034.

2) *Matlab Simulations*: For both methods we performed several simulations with different initial conditions and measured the error between the ground truth position (zero angles) and the output of the algorithms. For each initial condition, ten experiments were performed. In each experiment, only ten iterations of the algorithms are executed, since we experimentally verified that when the algorithms converge, they typically converge quickly. Nevertheless, in some experiments the algorithms did not converge; therefore, only the successful trials are considered when computing the average error between the ground truth position and the output of the algorithm (we consider a trial to be successful if the error corresponding to each one of the joint angles is less than 0.5rad). Tables I and II show the average errors (absolute average errors) expressed in radians as well as the number of non-convergent trials out of ten.

3) *Implementation on iCub*: Broyden's algorithm was successfully implemented on the iCub. Considering the nature of the problem at hand, it is quite difficult to evaluate its

¹Intel Research. Open Computer Vision Library. <http://www.intel.com/research/mrl/research/opencv/>

TABLE I
APPLICATION OF NEWTON'S METHOD TO FOUR DISTINCT INITIAL CONFIGURATIONS

Initial Configuration	θ_{tilt}	θ_{swing}	θ_{pan}	#Failures
$\frac{\pi}{6}, \frac{\pi}{6}$	0.0509	0.0629	0.1067	1
$\frac{\pi}{6}, \frac{\pi}{12}$	0.0493	0.0431	0.0706	0
$\frac{\pi}{12}, \frac{\pi}{12}$	0.0436	0.0902	0.1460	1
$\frac{\pi}{12}, \frac{\pi}{12}$	0.0593	0.0298	0.1233	1

TABLE II
APPLICATION OF NEWTON'S METHOD TO FOUR DISTINCT INITIAL CONFIGURATIONS

Initial Configuration	θ_{tilt}	θ_{swing}	θ_{pan}	#Failures
$\frac{\pi}{6}, \frac{\pi}{6}$	0.0812	0.0309	0.1334	1
$\frac{\pi}{6}, \frac{\pi}{12}$	0.0778	0.0709	0.1404	1
$\frac{\pi}{12}, \frac{\pi}{12}$	0.0177	0.0228	0.0807	0
$\frac{\pi}{12}, \frac{\pi}{12}$	0.0222	0.0179	0.0692	2

results in practice, since the real zero is not known. Nevertheless, we can illustrate the application of the algorithm by showing how the entries of the rotation matrix provided by inertial sensor change during the corresponding application. When the head of the robot is aligned with its body, the orientation matrix provided by the inertial sensor must be a rotation about the Z axis. So, the entries r_{31} , r_{32} , r_{13} and r_{23} must be zero and entry r_{33} must be 1. Table III presents the evolution of these entries when Broyden's algorithm is applied to the real robot. Naturally, it is only possible to estimate the initial joint angles after applying the algorithm and assuming that the head is then aligned with the body.

B. Eye Calibration

To illustrate the visual based calibration method for the eye joints, we have performed several experiments with the iCub platform. The standard calibration procedure was executed as follows: joints were moved to the hardware limits and then were displaced by a predetermined amount in order to (coarsely) point to the frontal direction. This is considered the "zero" angle of the system. Let us denote this reference as the canonical eye pan.

In a first experience we have set the canonical eye pan to the values $-20, -15, -10, -5, 0, 5, 10, 15$ and 20 degrees, and measured the true eye pan angle using the proposed method. The prospective eye tilt movements had amplitudes $\pm 20, \pm 10, \pm 5$ and ± 2 degrees. The results of this experiment are shown in Fig. 3. The first observation to make is that the canonical and the measured angles are offset by about five degrees. This is not surprising given that the canonical calibration was done once at assembly time and since then no adjustments were made. Another observation to make is that the relationship is approximately linear with the range of prospective motions employed. Furthermore, we noticed that for eye tilt prospective motions smaller than one degree, the variance of the measurements increase significantly and such values should be avoided.

To better evaluate the behavior of the method in the vicinity of the true zero, a finer discretization of the canonical angles was tested: 2, 3, 4, 5, 6, 7 and 8 degrees. A single

TABLE III
EVOLUTION OF THE INERTIAL SENSOR ORIENTATION MATRIX ALONG THE ITERATIONS OF THE METHOD. INITIAL CONFIGURATION:
 $\theta_{tilt}^0 = 0.7561rad$ $\theta_{swing}^0 = 0.3047rad$ $\theta_{pan}^0 = 0.5927rad$

Iterations	r_{31}	r_{32}	r_{13}	r_{23}	r_{33}
0	0.448124	-0.565588	-0.365259	-0.622327	0.692311
1	0.404702	0.03108	0.220109	-0.34103	0.91392
2	0.030612	0.262968	0.250975	0.084264	0.964319
3	-0.138035	0.059402	-0.008302	0.150045	0.988644
4	-0.064814	-0.097917	-0.116638	0.013573	0.993082
5	0.054952	-0.052626	-0.019847	-0.073453	0.997101
6	0.042871	0.023935	0.041518	-0.02621	0.998794
7	0.003065	0.041944	0.038273	0.017432	0.999115
8	-0.029368	-0.002641	-0.016464	0.024462	0.999565
9	0.003777	-0.005103	-0.002637	-0.005775	0.99998

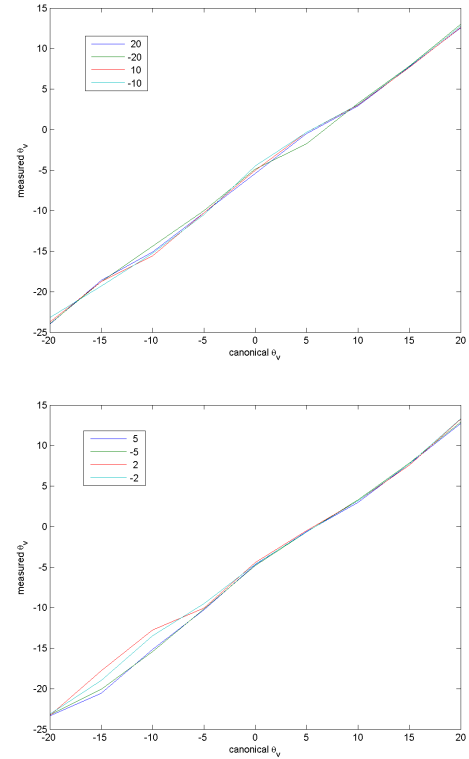


Fig. 3. Relationship between the canonical and measured eye pan values, for different eye tilt motion amplitudes.

amplitude was used in the eye tilt prospective motion: 20 degrees. The results are shown in Fig. 4. Again, an almost linear characteristic and the 5 degree bias are noticeable.

A third experiment was performed to evaluate the precision of the measurements close to the true zero. The canonical angle was set to 5 degrees and twelve measurements were taken in different image conditions. Again a eye tilt amplitude of 20 degrees was used. Computing the variance of the data set we obtained a value of about 0.6 degrees. Thus, the precision of the proposed method is below one degree in the vicinity of the true zero.

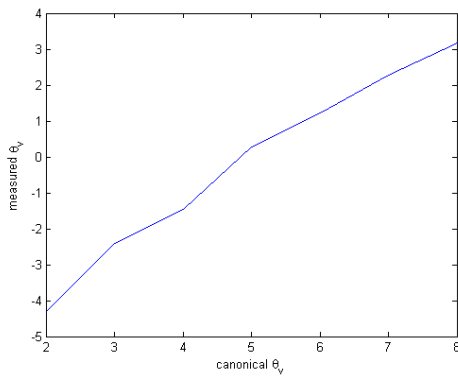


Fig. 4. Relationship between the canonical and measured eye pan values in the vicinity of the real zero.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a method to automatically calibrate the neck and the eye joints of a humanoid robot to its default configuration, using, respectively, measurements from the inertial sensor placed in the top of the head and the motion observed by the cameras. The method is suitable to be used in systems lacking absolute sensing in their actuators' joints, systems with low accuracy calibration or systems requiring frequent recalibration.

Two alternative methods for head calibration were presented: (i) a batch method requiring the acquisition of inertial measurements on several different head configurations before computing the solution; and (ii) an incremental method computing the solution while performing the prospective movements. Results show that both methods provide good results in most cases. However, in some circumstances, mainly when the head configuration is taken to the limits of the workspace, the methods may not converge to the solution. Whereas these situations can be diagnosed online using the incremental method, the batch method is not able to deal with these cases.

The Broyden's method, due to its ability to diagnose online algorithm non-convergence and to intrinsically avoid the joint limits is the method of our choice. In terms of speed, we have empirically shown that it converges to values close to zero in less than ten iterations. Errors measured in the real robot and in simulation trials with realistic noise conditions are very low and confirm the practical utility of the proposed method.

The method implemented for eye calibration is based on the computation of the homography induced by the rotation of a single axis. A set of points is tracked in the images before and after the prospective motions. A robust procedure allows the estimation of the homographies from the tracked points, even in the presence of tracker failures (outliers). By relating the homography entries with the unknown initial angles, it was possible to estimate them reliably solely using the visual measurements. Furthermore, it was possible to discover that the default calibration of the iCub right eye, in our setup, is offset by about 5 degrees.

In future work, we aim at performing a more extensive

analysis of the convergence basin of both the algorithms presented for the head calibration. Additionally, for the eye calibration, we want to develop a methodology for assessing the reliability of the measurements as a function of the amplitude of the prospective motions and the quality of the tracked visual features. Such techniques would allow rejecting unreliable measurements and thus providing a completely unsupervised self-calibration process.

VII. ACKNOWLEDGMENTS

This work was partially supported by the Portuguese Government - Fundação para a Ciência e Tecnologia (ISR/IST plurianual funding) through the PIDDAC Program funds, through project BIO-LOOK, PTDC/EEA-ACR/71032/2006 and through EU project RoboSoM, FP7-ICT-248366.

REFERENCES

- [1] R. Beira, M. Lopes, M. Praça, J. Santos-Victor, A. Bernardino, G. Metta, F. Becchi, and R. Saltarn. Design of the robot-cub (icub) head. *Proc. IEEE International Conference on Robotics and Automation, ICRA 2006*, May 2006.
- [2] G. Broyden. A class of methods for solving non-linear simultaneous equations. *Mathematics of Computation*, 19:577–593, October 1965.
- [3] J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, first edition, 1955.
- [4] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.
- [5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [6] E. Hayman, J. Knight, and D. Murray. Self-alignment of an active head from observation of rotation matrices. *Proc. of the International Conference on Pattern Recognition, ICPR 2000*, September 2000.
- [7] J. Knight and I. Reid. Automated alignment of robotic pan-tilt camera units using vision. *International Journal of Computer Vision*, 68(3):219–237, 2006.
- [8] L. Ljung. *System Identification - Theory for the User*. Prentice Hall, second edition, 1999.
- [9] A. Martinelli, D. Scaramuzza, and R. Siegwart. Automatic self-calibration of a vision system during robot motion. *Proc. IEEE International Conference on Robotics and Automation, ICRA 2006*, May 2006.
- [10] A. Martinelli, N. Tomatis, and R. Siegwart. Simultaneous localization and odometry self calibration. *Autonomous Robots*, 22:75–85, 2007.
- [11] R. Martinez-Cantin, M. Lopes, and L. Montesano. Body schema acquisition through active learning. *Proc. IEEE International Conference on Robotics and Automation, ICRA 2010*, May 2010.
- [12] Y. Meng and H. Zhuang. Self-calibration of camera-equipped robot manipulators. *The International Journal of Robotics Research*, 20(11):909–921, November 2001.
- [13] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, second edition, 1999.
- [14] N. Roy and S. Thrun. Online self-calibration for mobile robots. *Proc. IEEE International Conference on Robotics and Automation, ICRA 1999*, May 1999.
- [15] J. Shi and C. Tomasi. Good features to track. *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 1994*, 1994.
- [16] D. Song, N. Qin, and K. Goldberg. A minimum variance calibration algorithm for pan-tilt robotic cameras in natural environments. *Proc. of the IEEE International Conference on Robotics and Automation, ICRA 2006*, May 2006.
- [17] B. Tworek, A. Bernardino, and J. Santos-Victor. Visual self-calibration of pan-tilt kinematic structures. *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, ROBOTICA 2008*, April 2008.