Simultaneous Robot-World and Hand-Eye Calibration

Fadi Dornaika and Radu Horaud

***Abstract*—Recently, Zhuang, Roth, & Sudhakar [1] proposed a method that allows simultaneous computation of the rigid transformations from world frame to robot base frame and from hand frame to camera frame. Their method attempts to solve a homogeneous matrix equation of the form $\mathbf{AX} = \mathbf{ZB}$. They use quaternions to derive explicit linear solutions for X and Z. In this short paper, we present two new solutions that attempt to solve the homogeneous matrix equation mentioned above: (i) a closed-form method which uses quaternion algebra and a positive quadratic error function associated with this representation and (ii) a method based on non-linear constrained minimization and which simultaneously solves for rotations and translations. These results may be useful to other problems that can be formulated in the same mathematical form. We perform a sensitivity analysis for both our two methods and the linear method developed by Zhuang et al. [1]. This analysis allows the comparison of the three methods. In the light of this comparison the non-linear optimization method, which solves for rotations and translations simultaneously, seems to be the most stable one with respect to noise and to measurement errors.**

I. Introduction

In order to use a gripper-mounted sensor (such as a camera) for a robot task, the position and orientation of the sensor frame with respect to the gripper frame must be known. The problem of determining this relationship is referred to as the hand-eye calibration problem. One can find this relationship by moving the robot and observing the resulting motion of the sensor. This calibration problem yields a homogeneous matrix equation of the form $\mathbf{AX} = \mathbf{XB}$. Several closed-form solutions were proposed in the past to solve for $\mathbf{X}$ [2], [3], [4], [5] as well as a non-linear optimization method [6].

Recently, Zhuang et al. [1] proposed a method that allows the simultaneous estimation of both the transformations from the world-centered frame to the robot-base frame and from the gripper frame to camera frame. The identification problem is cast into the problem of solving a system of homogeneous matrix equations of the form $\mathbf{AX} = \mathbf{ZB}$, where $\mathbf{X}$ is the gripper-to-camera rigid transformation and $\mathbf{Z}$ is the robot-to-world rigid transformation. Quaternion algebra is applied to derive explicit linear solutions for $\mathbf{X}$ and $\mathbf{Z}$.

The mathematical framework of $\mathbf{AX} = \mathbf{ZB}$ allows one to solve for at least two types of robotic configurations. These configurations are shown on Figure 1 and Figure 2. It is worthwhile to notice that matrices $\mathbf{X}$ and $\mathbf{Z}$ can be estimated either sequentially or simultaneously. Therefore two approaches are possible:

1) $\mathbf{X}$ is estimated first using any hand-eye (or camera-gripper) calibration method and $\mathbf{Z}$ is estimated by solving the equation $\mathbf{AX} = \mathbf{ZB}$, or
2) $\mathbf{X}$ and $\mathbf{Z}$ are simultaneously estimated by solving $\mathbf{AX} = \mathbf{ZB}$ where both $\mathbf{X}$ and $\mathbf{Z}$ are unknowns.

This paper describes both a closed-form solution and a non-linear solution for the system of matrix equations $\mathbf{AX} = \mathbf{ZB}$. These solutions solve for two rotations and two translations that are associated with the matrices $\mathbf{X}$ and $\mathbf{Z}$. Likewise the linear method [1] the closed-form and non-linear methods yield a unique solution provided that the robot performs two motions with distinct rotation axes. The main differences between the linear method and the closed-form method introduced in this paper are the followings:

- The linear method first solves linearly for the components of two quaternions and second it normalizes these quaternions such that they represent rotations. The closed-form method solves directly for two unit quaternions and hence the constraint that these quaternions must represent two rotations is built in the resolution method.
- The linear method is not feasible for some special configurations (see [1] and below). We show that the closed-form method remains feasible for such special configurations.

We perform a sensitivity analysis for both our methods and for the linear method of Zhuang et al. [1]. This analysis allows the comparison of the three methods. In the light of both simulated and real experiments, it appears that the non-linear optimization method, which solves for rotations and translations simultaneously, performs better than the closed-form method which in turn performs slightly better than the linear method.

The remainder of this paper is organized as follows. Section II briefly recalls the problem formulation and presents the linear solution suggested by Zhuang et al. [1]. The closed-form and non-linear methods are described in Section III. Section IV compares the three methods through a sensitivity analysis. Finally, Section V describes some experimental results and Section VI provides a short discussion.

Address: Inria Grenoble, 655, avenue de l'Europe, 38330 Montbonnot Saint-Martin, France. E-mail: Radu.Horaud@inria.fr

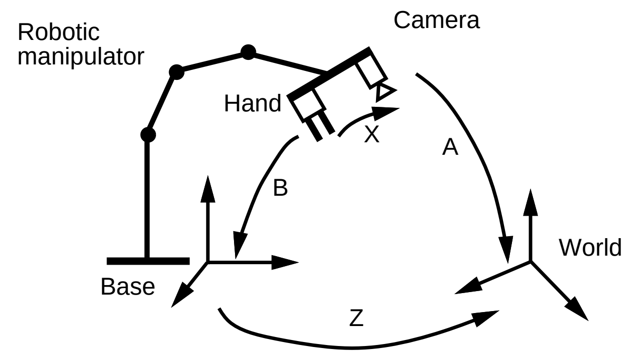


Fig. 1. Robot/world ($\mathbf{Z}$) and hand/eye ($\mathbf{X}$) calibration. The camera is mounted onto the gripper and camera motions are determined using a calibration pattern. The world frame is the frame of the calibration pattern.

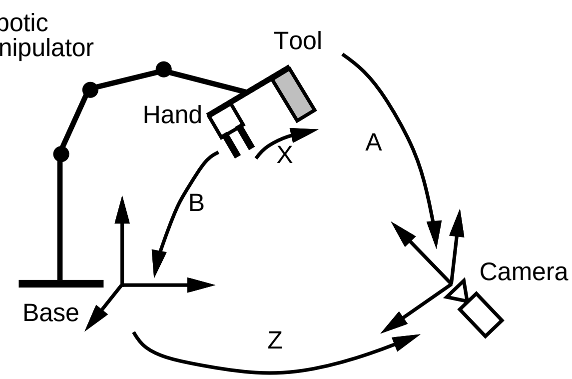


Fig. 2. Robot/eye ($\mathbf{Z}$) and hand/tool ($\mathbf{X}$) calibration. The tool is mounted onto the gripper and tool motions are determined by observing tool feature points with a camera. The world frame is, in this case, identical with the camera frame.

II. Problem formulation

We consider an arbitrary position of the robotic system (refer to Figures 1 and 2). From these figures we can write:

$$\mathbf{AX} = \mathbf{ZB} \tag{1}$$

In the particular case of a camera, the matrix $\mathbf{A}$ is obtained by calibrating the camera with respect to a fixed calibrating object and its associated frame, called the *calibration frame*. The matrix $\mathbf{B}$ is computed using the manipulator's forward kinematics whose parameters are supposed to be known (see [7] for an approach which attempts to estimate simultaneously these kinematic parameters and the hand-eye transformation). Let $\mathbf{R}_A$, $\mathbf{R}_B$, $\mathbf{R}_X$, and $\mathbf{R}_Z$ be the respective 3×3 rotation matrices of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{X}$, and $\mathbf{Z}$, and let $\boldsymbol{t}_A$, $\boldsymbol{t}_B$, $\boldsymbol{t}_X$, and $\boldsymbol{t}_Z$ be the respective 3×1 translational vectors. Equation (1) can then be written as:

$$\begin{bmatrix} \mathbf{R}_A & \boldsymbol{t}_A \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_X & \boldsymbol{t}_X \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_Z & \boldsymbol{t}_Z \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_B & \boldsymbol{t}_B \\ \mathbf{0}^T & 1 \end{bmatrix}$$

and one may easily decompose this equation into a rotation equation and a position equation:

$$\begin{aligned} \mathbf{R}_A \mathbf{R}_X &= \mathbf{R}_Z \mathbf{R}_B && (2) \\ \mathbf{R}_A\, \boldsymbol{t}_X + \boldsymbol{t}_A &= \mathbf{R}_Z\, \boldsymbol{t}_B + \boldsymbol{t}_Z && (3) \end{aligned}$$

Equation (3) is a linear equation in $\boldsymbol{t}_X$ and $\boldsymbol{t}_Z$ if $\mathbf{R}_Z$ is known.

A. Linear solution

This solution was suggested in [1]. Let $\boldsymbol{q}_A$, $\boldsymbol{q}_B$, $\boldsymbol{q}_X$, and $\boldsymbol{q}_Z$ be unit quaternions that correspond to the rotation matrices $\mathbf{R}_A$, $\mathbf{R}_B$, $\mathbf{R}_X$, and $\mathbf{R}_Z$ [8]. Since quaternions can be written as a combination of a scalar and a 3-vector, we have $\boldsymbol{q}_A^T = \left[a_0, \boldsymbol{a}^T\right]$ and so forth. The matrix equation $\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_Z\mathbf{R}_B$ is equivalent to the following quaternion equation:

$$\boldsymbol{q}_A * \boldsymbol{q}_X = \boldsymbol{q}_Z * \boldsymbol{q}_B \tag{4}$$

Expanding eq. (4) using quaternion products yields two constraints: a scalar equation and a vector equation:

$$\begin{aligned} a_0 x_0 - \boldsymbol{a}\cdot\boldsymbol{x} &= z_0 b_0 - \boldsymbol{b}\cdot\boldsymbol{z} && (5) \\ a_0\,\boldsymbol{x} + x_0\,\boldsymbol{a} + \boldsymbol{a}\times\boldsymbol{x} &= z_0\,\boldsymbol{b} + b_0\,\boldsymbol{z} - \boldsymbol{b}\times\boldsymbol{z} && (6) \end{aligned}$$

where $\cdot$ and $\times$ denote the dot-product and the vector product in the space of 3-vectors.

If $a_0 \neq 0$, x_0 can be solved from (5):

$$x_0 = (\boldsymbol{a}/a_0)\cdot\boldsymbol{x} + (b_0/a_0)\, z_0 - (\boldsymbol{b}/a_0)\cdot\boldsymbol{z} \tag{7}$$

By substitution of eq. (7) into eq. (6) and using the matrix representation to describe the vector and dot products yields:

$$\begin{aligned} &(a_0\mathbf{I} + \boldsymbol{a}\,\boldsymbol{a}^T/a_0 + \Omega(\boldsymbol{a}))\,\boldsymbol{x} + (-b_0 I - \boldsymbol{a}\,\boldsymbol{b}^T/a_0 + \Omega(\boldsymbol{b}))\,\boldsymbol{z} \\ &\quad = z_0\,\boldsymbol{b} - z_0\,(b_0/a_0)\,\boldsymbol{a} \end{aligned}$$

where $\Omega(\boldsymbol{a})$ is the skew-symmetric matrix associated with the 3-vector $\boldsymbol{a}$.

Therefore, we obtain (with $z_0 \neq 0$):

$$\underbrace{\mathbf{J}}_{3\times 6}\ \underbrace{\boldsymbol{u}}_{6\times 1} = z_0\,(\boldsymbol{b} - (b_0/a_0)\,\boldsymbol{a}) \tag{8}$$

where $\boldsymbol{u}^T = \left[\boldsymbol{x}^T,\ \boldsymbol{z}^T\right]$.

Equation (8) consists of three linear constraints with six unknowns. Therefore, a unique solution for $\boldsymbol{u}$ requires multiple measurements.

The solution of $\boldsymbol{u}$ can be obtained using standard linear algebra techniques. After $\boldsymbol{u}$ is obtained, the components of both $\boldsymbol{q}_X$ and $\boldsymbol{q}_Z$ can be determined using the constraints $\|\boldsymbol{q}_X\|^2 = \|\boldsymbol{q}_Z\|^2 = 1$ and eq. (7).

Following the solution of $\mathbf{R}_X$ and $\mathbf{R}_Z$, the computation of $\boldsymbol{t}_X$ and $\boldsymbol{t}_Z$ becomes trivial. Each position of the hand provides three linear equations with six unknowns (the components of $\boldsymbol{t}_X$ and $\boldsymbol{t}_Z$).

III. Problem solution

In this section we propose two alternatives for estimating $\mathbf{R}_X$, $\mathbf{R}_Z$, $\boldsymbol{t}_X$, and $\boldsymbol{t}_Z$: A closed-form method and a non-linear method which do not suffer from the above limitations, e.g., $a_0 \neq 0$ and $z_0 \neq 0$.

The closed-form method uses algebraic properties associated with quaternions to cast a sum of squares error function into a positive semi-definite quadratic form whose minimization uses two Lagrange multipliers. The non-linear method solves for all the unknowns simultaneously using standard minimization techniques. Interesting enough, the closed-form method is similar but not equivalent to the problem of optimally estimating rigid motion from 3-D to 3-D point or line correspondences [8], [9]. The method introduced in this paper solves simultaneously for two rotations in closed form while the methods developed in the past solved for one rotation in closed form.

A. Closed-form method

We start by building a positive error function that is derived from equation (4) as follows. Since the quaternion multiplication can be written in matrix form and with the notations introduced in [8] we have:

$$\begin{aligned} \boldsymbol{q}_{Ai} * \boldsymbol{q}_X &= Q(\boldsymbol{q}_{Ai})\,\boldsymbol{q}_X \\ \boldsymbol{q}_Z * \boldsymbol{q}_{Bi} &= W(\boldsymbol{q}_{Bi})\,\boldsymbol{q}_Z \end{aligned}$$

By substituting these equations into (4), we obtain:

$$Q(\boldsymbol{q}_{Ai})\,\boldsymbol{q}_X - W(\boldsymbol{q}_{Bi})\,\boldsymbol{q}_Z = \mathbf{0}$$

With matrices $Q(\boldsymbol{q})$ and $W(\boldsymbol{q})$ being defined by:

$$Q(\boldsymbol{q}) = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & -q_z & q_y \\ q_y & q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{bmatrix}$$

$$W(\boldsymbol{q}) = \begin{bmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{bmatrix}$$

Moreover, these two matrices are orthogonal and for a unit quaternion $\boldsymbol{q}$ we have:

$$Q(\boldsymbol{q})^T Q(\boldsymbol{q}) = \boldsymbol{q}^T\boldsymbol{q}\mathbf{I} = \mathbf{I}$$
$$W(\boldsymbol{q})^T W(\boldsymbol{q}) = \boldsymbol{q}^T\boldsymbol{q}\mathbf{I} = \mathbf{I}$$

The squared norm of the corresponding error vector is given by the following positive quadratic form:

$$\begin{aligned} & \|Q(\boldsymbol{q}_{Ai})\,\boldsymbol{q}_X - W(\boldsymbol{q}_{Bi})\,\boldsymbol{q}_Z\|^2 \\ = \ & [Q(\boldsymbol{q}_{Ai})\,\boldsymbol{q}_X - W(\boldsymbol{q}_{Bi})\,\boldsymbol{q}_Z]^T\ [Q\,[\boldsymbol{q}_{Ai})\,\boldsymbol{q}_X - W(\boldsymbol{q}_{Bi})\,\boldsymbol{q}_Z] \\ = \ & \boldsymbol{q}_X^T Q(\boldsymbol{q}_{Ai})^T Q(\boldsymbol{q}_{Ai})\boldsymbol{q}_X + \boldsymbol{q}_Z^T W(\boldsymbol{q}_{Bi})^T W(\boldsymbol{q}_{Bi})\boldsymbol{q}_Z - \\ & \boldsymbol{q}_Z^T W(\boldsymbol{q}_{Bi})^T Q(\boldsymbol{q}_{Ai})\boldsymbol{q}_X \\ & -\boldsymbol{q}_X^T Q(\boldsymbol{q}_{Ai})^T W(\boldsymbol{q}_{Bi})\boldsymbol{q}_Z \end{aligned}$$

Let $\boldsymbol{v}$ be an 8-vector given by:

$$\boldsymbol{v}^T = \left[\boldsymbol{q}_X^T,\, \boldsymbol{q}_Z^T\right]$$

Thus, we can write:

$$\|Q(\boldsymbol{q}_{Ai})\,\boldsymbol{q}_X - W(\boldsymbol{q}_{Bi})\,\boldsymbol{q}_Z\|^2 = \boldsymbol{v}^T \mathbf{S}_i\,\boldsymbol{v}$$

with $\mathbf{S}_i$ being an 8×8 positive semi-definite symmetric matrix:

$$\mathbf{S}_i = \begin{bmatrix} \mathbf{I} & \mathbf{C}_i \\ \mathbf{C}_i^T & \mathbf{I} \end{bmatrix} \tag{9}$$

where $\mathbf{C}_i = -Q(\boldsymbol{q}_{Ai})^T W(\boldsymbol{q}_{Bi})$ is an orthogonal matrix of rank equal to 4.

Finally, the error function that will allow us to compute $\boldsymbol{q}_X$ and $\boldsymbol{q}_Z$ becomes (n is the number of different positions of the robot):

$$f(\boldsymbol{q}_X, \boldsymbol{q}_Z) = \sum_{i=1}^{n} \boldsymbol{v}^T \mathbf{S}_i\,\boldsymbol{v} = \boldsymbol{v}^T \left(\sum_{i=1}^{n} \mathbf{S}_i\right) \boldsymbol{v} = \boldsymbol{v}^T \mathbf{S}\,\boldsymbol{v} \tag{10}$$

with:

$$\mathbf{S} = \begin{bmatrix} n\mathbf{I} & \mathbf{C} \\ \mathbf{C}^T & n\mathbf{I} \end{bmatrix}$$

Notice that $\mathbf{C} = \sum_{i=1}^{n} \mathbf{C}_i$ is the sum of n orthogonal matrices. In the general case $\mathbf{C}$ has full rank and there may be geometric configurations for which $\mathbf{C}$ is rank deficient. However, such geometric configurations are very rare in practice and, without loss of generality, one may assume that $\mathbf{C}$ has always full rank. The function $f(\boldsymbol{q}_X, \boldsymbol{q}_Z)$ is a positive semi-definite quadratic form and one way to minimize it is to use two Lagrange multipliers:

$$\begin{aligned} \min_{\boldsymbol{v}} f &= \min_{\boldsymbol{q}_X, \boldsymbol{q}_Z} ((\boldsymbol{q}_X\ \boldsymbol{q}_Z)^T \mathbf{S} (\boldsymbol{q}_X\ \boldsymbol{q}_Z) \\ &\quad + \lambda_1 (1 - \boldsymbol{q}_X^T \boldsymbol{q}_X) + \lambda_2 (1 - \boldsymbol{q}_Z^T \boldsymbol{q}_Z)) \end{aligned}$$

By developing and grouping terms we obtain:

$$\begin{aligned} f(\boldsymbol{q}_X, \boldsymbol{q}_Z) &= (n-\lambda_1)\boldsymbol{q}_X^T \boldsymbol{q}_X + (n-\lambda_2)\boldsymbol{q}_Z^T \boldsymbol{q}_Z \\ &\quad + \boldsymbol{q}_X^T \mathbf{C} \boldsymbol{q}_Z + \boldsymbol{q}_Z^T \mathbf{C}^T \boldsymbol{q}_X + \lambda_1 + \lambda_2 \end{aligned} \tag{11}$$

This function passes through a minimum when the first derivatives vanish. By differentiating with respect to the components of $\boldsymbol{q}_X$ and $\boldsymbol{q}_Z$ we obtain:

$$(n-\lambda_1)\boldsymbol{q}_X + \mathbf{C}\boldsymbol{q}_Z = 0 \tag{12}$$

$$(n-\lambda_2)\boldsymbol{q}_Z + \mathbf{C}^T\boldsymbol{q}_X = 0 \tag{13}$$

From equation (12) we obtain:

$$\boldsymbol{q}_X = \frac{1}{\lambda_1 - n}\mathbf{C}\boldsymbol{q}_Z \tag{14}$$

and by substituting $\boldsymbol{q}_X$ in equation (13) we obtain:

$$\mathbf{C}^T \mathbf{C} \boldsymbol{q}_Z = (\lambda_1 - n)(\lambda_2 - n)\boldsymbol{q}_Z \tag{15}$$

Therefore $\boldsymbol{q}_Z$ is an eigenvector of the symmetric positive semi-definite matrix $\mathbf{C}^T\mathbf{C}$. Such a matrix has four real positive eigenvalues $\alpha_i,\ i = \{1...4\}$ and we have an eigenvector $\boldsymbol{e}_i$ for each eigenvalue:

$$\mathbf{C}^T \mathbf{C} \boldsymbol{e}_i = \alpha_i \boldsymbol{e}_i$$

Notice that by substituting equations (14) and (15) into equation (11) we obtain the value of the error function at the point where the first derivatives vanish:

$$f(\boldsymbol{q}_X, \boldsymbol{q}_Z) = \lambda_1 + \lambda_2$$

Therefore, we must choose an eigenvalue α_i which minimizes $\lambda_1 + \lambda_2$. Let us consider the fact that $\boldsymbol{q}_X$ must be a unit quaternion. We obtain from equations (12) and (15):

$$\begin{aligned} \boldsymbol{q}_X^T \boldsymbol{q}_X &= \frac{1}{(\lambda_1 - n)^2} \boldsymbol{q}_Z^T \mathbf{C}^T \mathbf{C} \boldsymbol{q}_Z \\ &= \frac{1}{(\lambda_1 - n)^2} \boldsymbol{q}_Z^T (\lambda_1 - n)(\lambda_2 - n) \boldsymbol{q}_Z \\ &= \frac{\lambda_2 - n}{\lambda_1 - n} = 1 \end{aligned}$$

Hence, we must have:

$$\lambda_1 = \lambda_2 \neq 0$$

The relationship between $\lambda_1 = \lambda_2 = \lambda$ and α_i, i.e., equation (15) is:

$$(\lambda - n)^2 = \alpha_i$$

which yields the following solutions for λ:

$$\lambda = n \pm \sqrt{\alpha_i}$$

Since λ must be a positive number, one has to select among the four positive eigenvalues, the eigenvalue α_i such that $n \pm \sqrt{\alpha_i}$ is the smallest positive number.

Once the rotations, $\mathbf{R}_X$ and $\mathbf{R}_Z$, have been determined, the problem of determining the best translations, $\boldsymbol{t}_X$ and $\boldsymbol{t}_Z$, becomes a linear least-squares problem that can be easily solved using standard linear algebra techniques.

1) Configurations defeating the linear method: There are two configurations for which the linear method fails to provide a solution: $z_0 = 0$ and $a_0 = 0$ (see Section II-A). Clearly the closed-form solution is able to deal with situations for which $z_0 = 0$. The case $a_0 = 0$ is a little bit more complex to analyse. First, notice that the 4×4 matrices Q($\boldsymbol{q}$) and W($\boldsymbol{q}$) have full rank for all non null quaternions $\boldsymbol{q}$. Let, for some i, $\boldsymbol{q}_{Ai} = [0, \boldsymbol{a}_i^T]^T$. $Q(\boldsymbol{q}_{Ai})$ becomes a skew-symmetric matrix of full rank for all $\boldsymbol{a}_i \neq \mathbf{0}$. Hence, the rank of $\mathbf{S}_i$ in equation (9) is not affected by such a special case. However there is an ambiguity associated with purely imaginary unit quaternions because the quaternions $\boldsymbol{q}_{Ai} = [0, \boldsymbol{a}_i^T]^T$ and $\boldsymbol{q}_{Ai} = [0, -\boldsymbol{a}_i^T]^T$ describe the same rotation matrix $\mathbf{R}_{Ai}$. Hence, one has two consider two distinct matrices associated with this special configuration:

$$\mathbf{S}_i^+ = \begin{bmatrix} \mathbf{I} & \mathbf{C}_i \\ \mathbf{C}_i^T & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{S}_i^- = \begin{bmatrix} \mathbf{I} & -\mathbf{C}_i \\ -\mathbf{C}_i^T & \mathbf{I} \end{bmatrix}$$

Therefore, any time such a special configuration is present in the data, one has to consider two distinct error functions. There will be two two distinct solutions for $\boldsymbol{q}_X$ and $\boldsymbol{q}_Z$. One may simply consider, among these two solutions, the solution yielding the smallest minimum.

B. Non-linear method

There are several disadvantages associated with the above methods:

1) The unknowns are estimated in sequence, rotations first and then translations. Errors from the first stage propagate to the second stage;
2) It is well known that the performance of linear resolution methods degrades in the presence of noise, and
3) Unlike non-linear minimization, linear and closed-form solutions do not allow a characterization of both the quality of the solution and the confidence associated with the solution.

In this Section, we propose to overcome the disadvantages mentioned above. For this purpose we estimate simultaneously the rotations and translations associated with $\mathbf{X}$ and $\mathbf{Z}$. This leads to a non-linear minimization problem. There are 24 parameters associated with two rotation matrices (18 parameters) and two translation vectors (6 parameters). The initialization of these unknowns is straightforward because one can use either of the two methods outlined above. Non-linear minimization provides information about both the quality of the solution (the depth of the global minimum) and the confidence associated with this solution (the width of the global minimum).

If we have n positions of the robot, the calibration problem becomes the problem of solving for a set of $2n$ non-linear constraints derived from equations (2) and (3), or equivalently, the problem of minimizing the following error function:

$$\begin{aligned} f(\mathbf{R}_X, \mathbf{R}_Z, \boldsymbol{t}_X, \boldsymbol{t}_Z) &= \mu_1 \sum_{i=1}^{n} \left(\|\mathbf{R}_{Ai}\,\mathbf{R}_X - \mathbf{R}_{Bi}\,\mathbf{R}_Z\|^2 \right) \\ &+ \mu_2 \sum_{i=1}^{n} \left(\|\mathbf{R}_{Ai}\,\boldsymbol{t}_X + \boldsymbol{t}_{Ai} - \mathbf{R}_Z\,\boldsymbol{t}_{Bi} - \boldsymbol{t}_Z\|^2 \right) \\ &+ \mu_3 \|\mathbf{R}_X \mathbf{R}_X^T - \mathbf{I}\|^2 + \mu_4 \|\mathbf{R}_Z \mathbf{R}_Z^T - \mathbf{I}\|^2 \end{aligned}$$

The criterion to be minimized is of the form:

$$\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) = \frac{1}{2} \sum_{j=1}^{m} \Phi_j^2(\boldsymbol{x}) : \boldsymbol{x} \in I\!R^{24} \right\}$$

Therefore, the problem becomes a classical non-linear least-squares constrained minimization problem and one can apply standard non-linear optimization techniques, such as Newton's method and Newton-like methods [10], [11]. In this error function, the terms Φ_j are quadratic in the unknowns. Notice that the last two terms are *penalty* functions which constrain the matrices $\mathbf{R}_X$ and $\mathbf{R}_Z$ to be rotations. The parameters μ_1 through μ_4 are real positive numbers. High values for μ_3 and μ_4 inforce the role of the penalty functions In all our experiments we have set these parameters to the following values: $\mu_1 = \mu_2 = 1$ and $\mu_3 = \mu_4 = 10^6$ In the next two sections we give some results obtained with the Levenberg-Marquardt non-linear minimization method as described in [12] and in [11].

IV. Sensitivity analysis and method comparison

One of the most important merits of any calibration method is its sensitivity with respect to various perturbations. In our problem, there are two main sources of perturbations: errors associated with camera calibration and errors associated with the robot position. Indeed, the parameters of both the direct and inverse kinematic models of robots are not perfect. In order to investigate the behaviour of the methods in the presence of measurement noise we designed a sensitivity analysis based on the following grounds:

- Nominal values for the parameters of both the hand-eye transformation $\mathbf{X}$ and the robot-to-world transformation $\mathbf{Z}$ are provided;
- Also are provided n matrices $\mathbf{A}_1, \ldots \mathbf{A}_n$ from which n hand positions can be computed with:

$$\mathbf{B}_i = \mathbf{Z}^{-1} \mathbf{A}_i \mathbf{X}$$

- Either Gaussian noise or uniform noise is added to both camera and robot positions; the homogeneous transformations, ($\mathbf{X}$ and $\mathbf{Z}$), are estimated in the presence of this noise using the three methods described in this paper: the linear method, the closed-form method and the non-linear method, and
- We study the variations of the estimation of the hand-eye transformation and the robot-to-world transformation as a function of the noise being added and/or as a function of the number of positions (n).

Since both rotations and translations may be represented as vectors, adding noise to a transformation consists of adding random numbers to each one of the vectors' components. Random numbers simulating noise are obtained using a random number generator either with a uniform distribution in the interval $[-C/2, +C/2]$, or with a Gaussian distribution with a standard deviation equal to σ. Therefore the level of noise that is added is associated either with the value of C or with the value of σ (or, more precisely, with the value of 2σ). In what follows the level of noise is in fact represented as a ratio: the amplitude of the actual random numbers (C or 2σ) divided by the nominal values of the perturbed parameters.

In the case of a rotation, the vector (quaternion) associated with this rotation has a module equal to 1 and therefore the ratio is simply either C or 2σ. In the case of a translation the ratio is computed with respect to a nominal value estimated over all the perturbed translations:

$$\|\boldsymbol{t}_{nominal}\| = \frac{\sum_{i=1}^{n} (\|\boldsymbol{t}_{A_i}\| + \|\boldsymbol{t}_{B_i}\|)}{2n}$$

where $\boldsymbol{t}_{A_i}$ is the translation vector associated with $\mathbf{A}_i$.

For each noise level and for a large number N of trials we compute the errors as follows. These errors are: orientation error and position error. The orientation error is defined as the rotation angle in degrees required to align the coordinate system of $\mathbf{X}$ or $\mathbf{Z}$ in its computed orientation with the coordinate system in its theoretical orientation. The position error is defined as the norm of the vector which represents the difference between the two translation vectors: the computed one and the theoretical one, divided by the norm of the second vector.

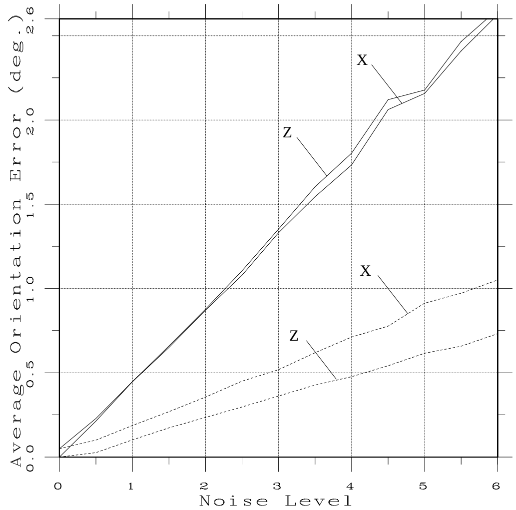
In all our simulations we set $N = 500$, $\|\boldsymbol{t}_X\| = 229$mm, and $\|\boldsymbol{t}_Z\| = 768$mm.

The following figures show the average of the above errors as a function of the percentage of noise. The percentage of noise varies from 1% to 6%. The full curves (—) correspond to the method in [1], the dotted curves (. . .) correspond to the closed-form method, and the dashed curves (- - -) correspond to the non-linear method.

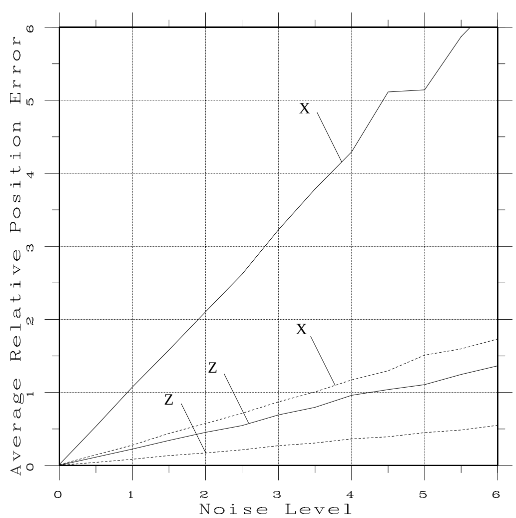
Figures 3 and 4 correspond to three positions ($n = 3$) of the robot while on Figure 5 the number of positions varies from 3 to 8.

Figure 3 shows the rotation and translation errors as a function of uniform noise added to the rotational part of the robot and camera positions. Figure 4 shows the rotation and translation errors as a function of Gaussian noise added to the rotational part of the hand and camera positions. These errors are obtained with the three methods. We can conclude that the closed-form method is more accurate than the linear method proposed in [1].

As other authors have done in the past, it is interesting to analyze the behaviour of calibration methods with respect to the number of positions. In order to perform this analysis we have to fix the percentage of noise. Figure 5 shows the rotational and translational errors as a function of the square root of the number of motions ($\sqrt{n}$ varies from 1.732 to 2.828). The noise ratio has been fixed to the worst case for rotations, e.g., 6% and to 2% for translations. Both rotational and translational noise distributions are Gaussian.

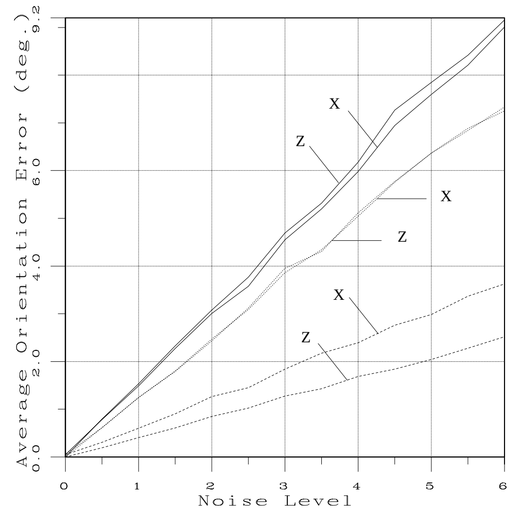


(a) Orientation errors.

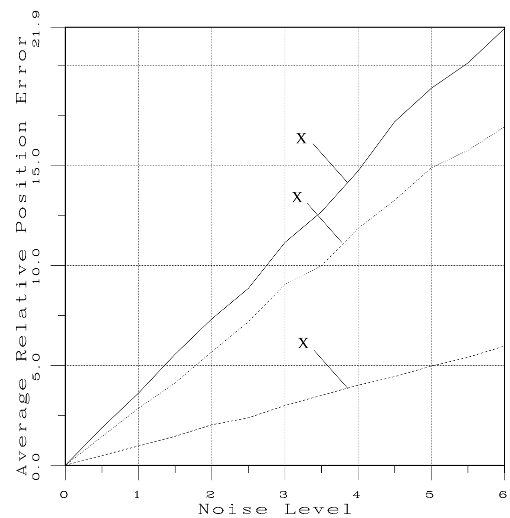


(b) Relative position errors.

Fig. 3. Errors in orientations and positions in the presence of uniform noise perturbing the rotation axes. The full curves (—) correspond to the method of [1] and the dashed curves (- - -) correspond to the non-linear method.



(a) Orientation errors.



(b) Relative position errors.

Fig. 4. Errors in orientations and positions in the presence of Gaussian noise perturbing the rotation axes. The full curves (—) correspond to the method of [1], the dotted curves (. . .) correspond to the closed-form solution and the dashed curves (- - -) correspond to the non-linear method.

V. EXPERIMENTAL RESULTS

In this Section we report some experimental results obtained with two sets of data. The first data set was obtained with 17 different positions of the hand-eye device with respect to a calibrating object. The second data set was obtained with 7 such positions. In order to calibrate the camera we used the classical method proposed by Faugeras & Toscani described in [9].

Our tests compare the linear method [1] with the two methods developed in this paper. Table I and Table II summarize the results obtained with the two data sets mentioned above. The second columns of these tables show the sum of squares of the absolute error in rotation. The third columns show the relative error in translation, namely

$$E_{\mathbf{R}} = \sum \|\mathbf{R}_A \mathbf{R}_X - \mathbf{R}_Z \mathbf{R}_B\|^2 \tag{16}$$

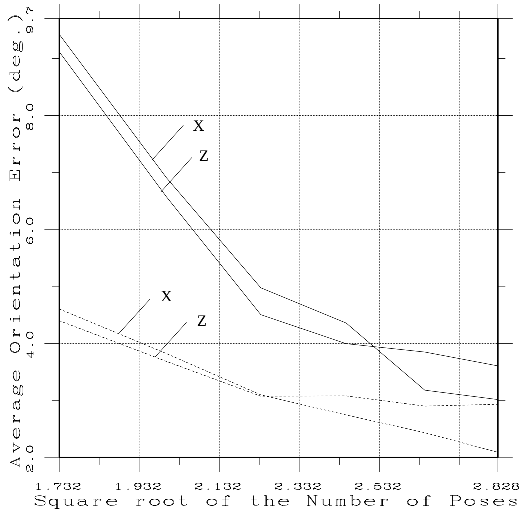
$$E_{\boldsymbol{t}} = \left(\frac{\sum \|(\mathbf{R}_A t_X + t_A - \mathbf{R}_Z t_B - t_Z\|^2}{\sum \|\mathbf{R}_A t_X + t_A\|^2} \right)^{1/2} \tag{17}$$

TABLE I

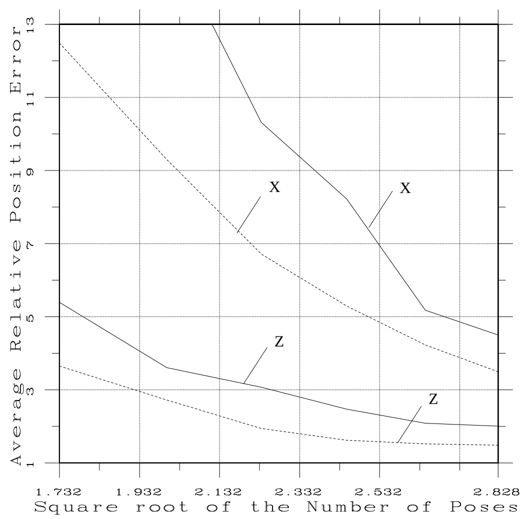
THE FORMULATION $\mathbf{AX} = \mathbf{ZB}$ USED WITH THE FIRST DATA SET (17 DIFFERENT POSITIONS OF THE HAND-EYE DEVICE). THESE DATA WERE OBTAINED WITH A PPPRRR ROBOT.

	$E_{\mathbf{R}}$, eq. (16)	$E_{\boldsymbol{t}}$, eq. (17)
Linear solution	0.00031	0.00068
Closed-form solution	0.00026	0.00075
Non-linear optimization	0.00071	0.00021

It is worthwhile to notice that the robots being used in the two experiments summarized in the tables above are not identical. The first data set (Table I) was obtained with a PPPRRR robot (three prismatic and three rotational joints) while the second data set (Table II) was obtained with a RRRRRR robot. Unlike the simulated data, these two experiments do not allow one to conclude that the closed-form solution outperforms the linear solution. In the first experiment the linear solution yields a smaller translation error than the translation error associated with the closed-form method. In the



(a) Orientation errors.



(b) Relative position errors.

Fig. 5. Errors in orientations and positions as a function of the number of positions. A Gaussian noise is added both to the robot and camera positions. The full curves (—) correspond to the method of [1] and the dashed curves (- - -) correspond to the non-linear method.

TABLE II
THE FORMULATION $\mathbf{AX} = \mathbf{ZB}$ USED WITH THE SECOND DATA SET (7 DIFFERENT POSITIONS OF THE HAND-EYE DEVICE). THESE DATA WERE OBTAINED WITH A RRRRRR ROBOT.

	$E_{\mathbf{R}}$, eq. (16)	$E_{\boldsymbol{t}}$, eq. (17)
Linear solution	0.12174	0.00738
Closed-form solution	0.00068	0.00515
Non-linear optimization	0.00109	0.00451

second experiment the translation error associated with the linear method does not seem to be affected by a large rotation error.

These experimental results seem however to confirm that the non-linear method provides a better estimation of the translation vectors *at the cost of slightly larger rotation errors*. This is due to the fact that the robot's kinematic chain is not perfectly calibrated and therefore there are errors associated with the robot's translation parameters. Obviously, these errors do not obey the noise models used for simulations.

VI. DISCUSSION

In this paper we addressed the problem of robot-to-world and hand-eye calibration. As it was proposed in [1] this problem is formulated as solving a system of homogeneous transformation equations of the form $\mathbf{AX} = \mathbf{ZB}$.

We develop two resolution methods, the first one solves for rotations and then for translations while the second one solves simultaneously for rotations and translations. The first method leads to a closed-form solution while the second one leads to non-linear optimization.

Both the sensitivity analysis and the results obtained with experimental data show that the closed-form method slightly outperforms the linear method of Zhuang et al. [1]. This is most probably due to the Euclidean nature of the error function suggested in Section III-A. However, there is no evidence that with real data the closed-form method will always perform better than the linear method: One can therefore conclude that the two methods have comparable performances.

The non-linear minimization method suggested in Section III-B yields the most accurate results and outperforms both the linear and closed-form methods. The solution obtained with either the linear or closed-form methods can be used to initialize the non-linear minimization method.

The two methods proposed in this paper together with [1] may be useful to other problems that can be formulated into homogeneous transformation equations of the form $\mathbf{AX} = \mathbf{ZB}$.

REFERENCES

[1] H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX= YB," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549–554, 1994.
[2] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Journal of Robotics and Automation*, vol. 5, no. 3, pp. 345–358, June 1989.
[3] C.-C. Wang, "Extrinsic calibration of a robot sensor mounted on a robot," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 161–175, April 1992.
[4] F. C. Park and B. J. Martin, "Robot sensor calibration: solving AX= XB on the Euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.
[5] Y. C. Shiu and S. Ahmad, "Calibration of wrist mounted robotic sensors by solving homogeneous transform equations of the form AX=XB," *IEEE Journal of Robotics and Automation*, vol. 5, no. 1, pp. 16–29, February 1989.
[6] R. Horaud and F. Dornaika, "Hand-eye calibration," *International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, June 1995.
[7] H. Zhuang, K. Wang, and Z. S. Roth, "Simultaneous calibration of a robot and a hand-mounted camera," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 5, pp. 649–660, October 1995.
[8] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-d location parameters using dual number quaternions," *CGVIP-Image Understanding*, vol. 54, no. 3, pp. 358–367, November 1991.
[9] O. D. Faugeras, *Three Dimensional Computer Vision: A Geometric Viewpoint*. Boston: MIT Press, 1993.
[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. London: Academic Press, 1989.
[11] R. Fletcher, *Practical Methods of Optimization*. John Wiley & Sons, 1990.
[12] W. Press, B. Flannery, S. Teukolsky, and W. Wetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.