Radu Horaud
Fadi Dornaika

LIFIA–IMAG and Inria Rhône-Alpes
46, avenue Félix Viallet
38031 Grenoble, France

Hand-Eye Calibration

Abstract

Whenever a sensor is mounted on a robot hand, it is important to know the relationship between the sensor and the hand. The problem of determining this relationship is referred to as the hand-eye calibration problem. *Hand-eye calibration is important in at least two types of tasks: (1) map sensor centered measurements into the robot workspace frame and (2) tasks allowing the robot to precisely move the sensor. In the past some solutions were proposed, particularly in the case of the sensor being a television camera. With almost no exception, all existing solutions attempt to solve a homogeneous matrix equation of the form $AX = XB$. This article has the following main contributions. First we show that there are two possible formulations of the hand-eye calibration problem. One formulation is the classic one just mentioned. A second formulation takes the form of the following homogeneous matrix equation: $MY = M'YB$. The advantage of the latter formulation is that the extrinsic and intrinsic parameters of the camera need not be made explicit. Indeed, this formulation directly uses the 3×4 perspective matrices (**M** and **M'**) associated with two positions of the camera with respect to the calibration frame. Moreover, this formulation together with the classic one covers a wider range of camera-based sensors to be calibrated with respect to the robot hand: single scan-line cameras, stereo heads, range finders, etc. Second, we develop a common mathematical framework to solve for the hand-eye calibration problem using either of the two formulations. We represent rotation by a unit quaternion and present two methods: (1) a closed-form solution for solving for rotation using unit quaternions and then solving for translation and (2) a nonlinear technique for simultaneously solving for rotation and translation. Third, we perform a stability analysis both for our two methods and for the linear method developed by Tsai and Lenz (1989). This analysis allows the comparison of the three methods. In light of this comparison, the nonlinear optimization method, which solves for rotation and translation simultaneously, seems to be the most robust one with respect to noise and measurement errors.*

1. Introduction

Whenever a sensor is mounted on a robot hand, it is important to know the relationship between the sensor and the hand. The problem of determining this relationship is referred to as the *hand-eye calibration problem*. Hand-eye calibration is important in at least two types of tasks:

1. *Map sensor centered measurements into the robot workspace frame.* Consider for example the task of grasping an object at an unknown location. First, an object recognition system determines the position and orientation of the object with respect to the sensor; second, the object location (position and orientation) is mapped from the sensor frame to the gripper (hand) frame. The robot may then direct its gripper toward the object and grasp it (Bard et al. In press).
2. *Tasks allowing the robot to precisely move the sensor.* This is necessary for inspecting complex 3-D parts (Horaud et al. 1992; 1993), for reconstructing 3-D scenes with a moving camera (Boufama et al. 1993), or for visual servoing (using a sensor inside a control servo loop) (Espiau et al. 1992).

In the past some solutions were proposed, particularly in the case of the sensor being a television camera. With almost no exception, all existing solutions attempt to solve a homogeneous matrix equation of the form (Shiu and Ahmad 1989; Tsai and Lenz 1989; Chou and Kamel 1991; Chen 1991; Wang 1992):

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

This article has the following main contributions.

First we show that there are two possible formulations of the hand-eye calibration problem. One formulation is the classic one just mentioned. A second formulation takes the form of the following homogeneous matrix equation:

$$\mathbf{MY} = \mathbf{M'YB} \tag{2}$$

The advantage of the latter formulation is that the extrinsic and intrinsic parameters of the camera need not be made explicit. Indeed, this formulation directly uses the 3×4 perspective matrices (**M** and **M'**) associated with two positions of the camera with respect to the calibration frame. Moreover, this formulation together with the classic one covers a wider range of camera-based sensors

The International Journal of Robotics Research,
Vol. 14, No. 3, June 1995, pp. 195–210,

to be calibrated with respect to the robot hand: single scan-line cameras, stereo heads, range finders, etc.

Second, we develop a common mathematical framework to solve for the hand-eye calibration problem using either of the two formulations. We represent rotation by a unit quaternion and present two methods: (1) a closed-form solution for solving for rotation using unit quaternions and then solving for translation and (2) a nonlinear technique for simultaneously solving for rotation and translation.

Third, we perform a stability analysis both for our two methods and for the classic linear method developed by Tsai and Lenz (1989). This analysis allows the comparison of the three methods. In the light of this comparison, the nonlinear optimization method, which solves for rotation and translation simultaneously, seems to be the most robust one with respect to noise and measurement errors.

The remainder of this article is organized as follows. Section 2 states the problem of determining the hand-eye geometry from the standpoints of both the classic formulation and our new formulation. Section 3 overviews the main approaches that attempted to determine a solution. Section 4 shows that the newly proposed formulation can be decomposed into two equations. Section 5 suggests two solutions, one based on the work of Faugeras and Hebert (1986) and a new one. Both of these solutions solve for the classic and the new formulations. Section 6 compares our methods with the well known Tsai-Lenz method through a stability analysis. Finally, Section 7 describes some experimental results, and Section 8 provides a short discussion. Appendix A summarizes rotation representation using unit quaternions, and Appendix B gives a derivation of eq. (31) using properties outlined in Appendix A.

2. Problem Formulation

The hand-eye calibration problem consists of computing the rigid transformation (rotation and translation) between a sensor mounted on a robot actuator and the actuator itself (i.e., the rigid transformation between the sensor frame and the actuator frame).

2.1. The Classic Formulation

The hand-eye problem is best described in Figure 1. Let position 1 and position 2 be two positions of the rigid body formed by a sensor fixed onto a robot hand; this will be referred to as the *hand-eye device*. Both the sensor and the hand have a Cartesian frame associated with them. Let $\mathbf{A}$ be the transformation between the two positions of the sensor frame and $\mathbf{B}$ be the transformation between the two positions of the hand frame. Let $\mathbf{X}$ be the transformation between the hand frame and the sensor frame. $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{X}$ are related by the formula given by eq. (1) and they are 4×4 matrices of the form:

$$\mathbf{A} = \begin{pmatrix} \mathbf{R}_\mathbf{A} & \mathbf{t}_\mathbf{A} \\ 0 & 1 \end{pmatrix}$$

In this expression, $\mathbf{R}_\mathbf{A}$ is a 3×3 orthogonal matrix describing a rotation, and $\mathbf{t}_\mathbf{A}$ is a 3-vector describing a translation.

Throughout the article we adopt the following notation: matrix $\mathbf{T}$ ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{X}$, $\mathbf{Y}$,...) is the transformation from frame b to frame a:

$$\begin{pmatrix} x_a \\ y_a \\ z_a \\ 1 \end{pmatrix} = \mathbf{T} \begin{pmatrix} x_b \\ y_b \\ z_b \\ 1 \end{pmatrix}$$

where a 3D point indexed by a is expressed in frame a.

In the particular case of a camera-based sensor, the matrix $\mathbf{A}$ is obtained by calibrating the camera twice with respect to a fixed calibrating object and its associated frame, called the *calibration frame*. Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be the transformations from the calibration frame to the camera frame in its two different positions. We have:

$$\mathbf{A} = \mathbf{A}_2\mathbf{A}_1^{-1}. \tag{3}$$

The matrix $\mathbf{B}$ is obtained by moving the robot hand from position 1 to position 2. Let $\mathbf{B}_1$ and $\mathbf{B}_2$ be the transformations from the hand frame in positions 1 and 2 to the robot-base frame. We have:

$$\mathbf{B} = \mathbf{B}_2^{-1}\mathbf{B}_1. \tag{4}$$

2.2. The New Formulation

The previous formulation implies that the camera is calibrated at each different position i of the hand-eye device. Once the camera is calibrated, its extrinsic parameters—namely, the matrix $\mathbf{A}_i$ for position i—are made explicit. This is done by decomposing the 3×4 perspective matrix $\mathbf{M}_i$, obtained by calibration, into intrinsic and extrinsic parameters (Faugeras and Toscani 1986; Tsai 1987; Horaud and Monga 1993):

$$\begin{aligned} \mathbf{M}_i &= \mathbf{C}\mathbf{A}_i \\ &= \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R}_\mathbf{A}^i & \mathbf{t}_\mathbf{A}^i \\ 0 & 1 \end{pmatrix}. \end{aligned} \tag{5}$$

The parameters α_u, α_v, u_0, and v_0 describe the affine transformation between the camera frame and the image frame. This decomposition assumes that the camera is described by a pinhole model and that the optical axis associated with this model is perpendicular to the image plane.

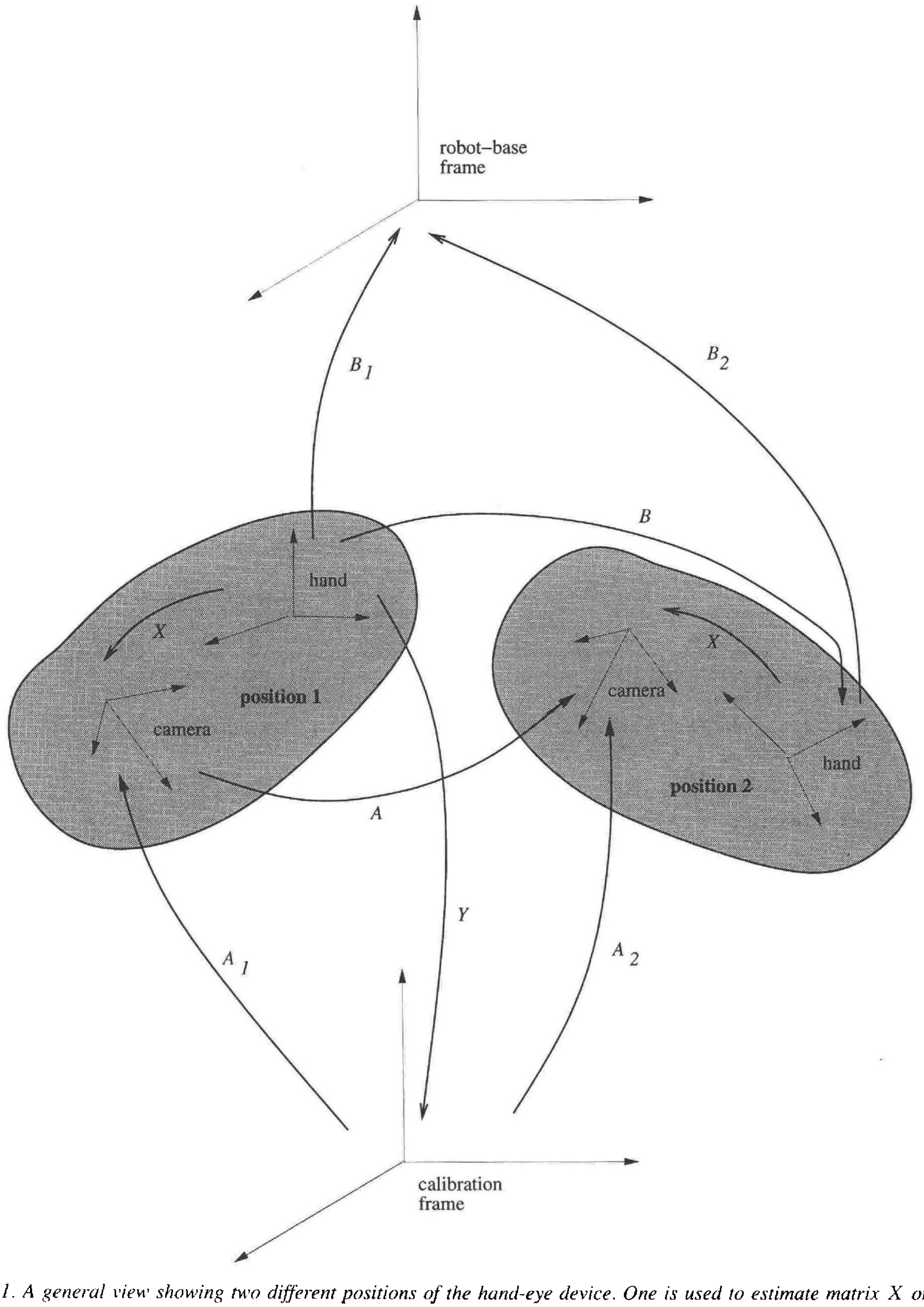

Fig. 1. A general view showing two different positions of the hand-eye device. One is used to estimate matrix X or, alternatively, matrix Y (see text).

The new formulation that we present here avoids the above decomposition. Let $\mathbf{Y}$ be the transformation matrix from the hand frame to the calibration frame, when the hand-eye device is in position 1. Clearly we have (e.g., Fig. 1):

$$\mathbf{X} = \mathbf{A}_1\mathbf{Y} \tag{6}$$

Therefore, matrix $\mathbf{Y}$ is equivalent to matrix $\mathbf{X}$, up to a rigid transformation $\mathbf{A}_1$. By substituting $\mathbf{X}$ given by this last equation and $\mathbf{A}$ given by eq. (3) into eq. (1), we obtain:

$$\mathbf{A}_2\mathbf{Y} = \mathbf{A}_1\mathbf{Y}\mathbf{B}$$

By premultiplying the terms of this equality with matrix $\mathbf{C}$ and using eq. (5) with $i = 1, 2$, we obtain:

$$\mathbf{M}_2\mathbf{Y} = \mathbf{M}_1\mathbf{Y}\mathbf{B}, \tag{7}$$

which is equivalent to eq. (2).

In this equation the unknown $\mathbf{Y}$ is the transformation from the hand frame to the calibration frame (e.g., Fig. 1). The latter frame may well be viewed as the camera frame, provided that the 3×4 perspective matrix $\mathbf{M}_1$ is known. Mathematically, choosing the calibration frame rather than the camera frame is equivalent to replacing the 3×4 perspective matrix $\mathbf{C}$ with the more general matrix $\mathbf{M}_1$. The advantage of using $\mathbf{M}_1$ rather than $\mathbf{C}$ is that one does not have to assume a perfect pinhole camera model any more. Therefore, problems caused by the decomposition of $\mathbf{M}_1$ into external and internal camera parameters (i.e., $\mathbf{M}_i = \mathbf{C}\mathbf{A}_i$) will disappear.

Referring to Figure 2, the projection of a point P onto the image is described by:

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \mathbf{M}_1 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{8}$$

or:

$$u = \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \tag{9}$$

$$v = \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \tag{10}$$

where x, y, and z are the coordinates of P in the calibration frame; u and v are the image coordinates of p, the projection of P; and the m_{ij}s are the coefficients of $\mathbf{M}_1$. Notice that these two equations can be rewritten as:

$$(m_{11} - u\,m_{31})\,x + (m_{12} - u\,m_{32})\,y + (m_{13} - u\,m_{33})\,z = u\,m_{34} - m_{14} \tag{11}$$

$$(m_{21} - v\,m_{31})\,x + (m_{22} - v\,m_{32})\,y + (m_{23} - v\,m_{33})\,z = v\,m_{34} - m_{24} \tag{12}$$

These equations may be interpreted as follows. Given a matrix $\mathbf{M}_1$ and an image point p, eqs. (11) and (12) describe a line of sight passing through the center of projection and through p. This line is given in the calibration frame, which may well be viewed as the camera frame.

The determination of the hand-eye geometry (matrix $\mathbf{X}$ in the classic formulation or matrix $\mathbf{Y}$ in our new formulation) allows one to express any line of sight associated with an image point p in the hand frame and, hence, in any robot-centered frame.

2.3. Summary

The classic and the new formulations can be summarized as follows. Let n be the number of different positions of the hand-eye device with respect to a fixed calibration frame. We have:

1. *Classic formulation.* The matrix $\mathbf{X}$ is the solution of the following set of $n - 1$ matrix equations:

$$\begin{cases} \mathbf{A}_{12}\mathbf{X} &= \mathbf{X}\mathbf{B}_{12} \\ & \vdots \\ \mathbf{A}_{i-1\,i}\mathbf{X} &= \mathbf{X}\mathbf{B}_{i-1\,i} \\ & \vdots \\ \mathbf{A}_{n-1\,n}\mathbf{X} &= \mathbf{X}\mathbf{B}_{n-1\,n} \end{cases} \tag{13}$$

   where $\mathbf{A}_{i-1\,i}$ denotes the transformation between position $i - 1$ and position i of the camera frame, and $\mathbf{B}_{i-1\,i}$ denotes the transformation between position $i - 1$ and position i of the hand frame.

2. *New formulation.* The matrix $\mathbf{Y}$ is the solution of the following set of $n - 1$ matrix equations:

$$\begin{cases} \mathbf{M}_2\mathbf{Y} &= \mathbf{M}_1\mathbf{Y}\mathbf{B}_{12} \\ & \vdots \\ \mathbf{M}_i\mathbf{Y} &= \mathbf{M}_1\mathbf{Y}\mathbf{B}_{1i} \\ & \vdots \\ \mathbf{M}_n\mathbf{Y} &= \mathbf{M}_1\mathbf{Y}\mathbf{B}_{1n} \end{cases} \tag{14}$$

   where $\mathbf{M}_i$ is the projective transformation between the calibration frame and the camera frame in position i and $\mathbf{B}_{1i}$ denotes the transformation between position 1 and position i of the hand frame.

3. Previous Approaches

Previous approaches for solving the hand-eye calibration problem attempted to solve eq. (1) ($\mathbf{AX} = \mathbf{XB}$) by further decomposing it into two equations: a matrix equation depending on rotation and a vector equation depending both on rotation and translation:

$$\mathbf{R}_\mathbf{A}\mathbf{R}_\mathbf{X} = \mathbf{R}_\mathbf{X}\mathbf{R}_\mathbf{B} \tag{15}$$

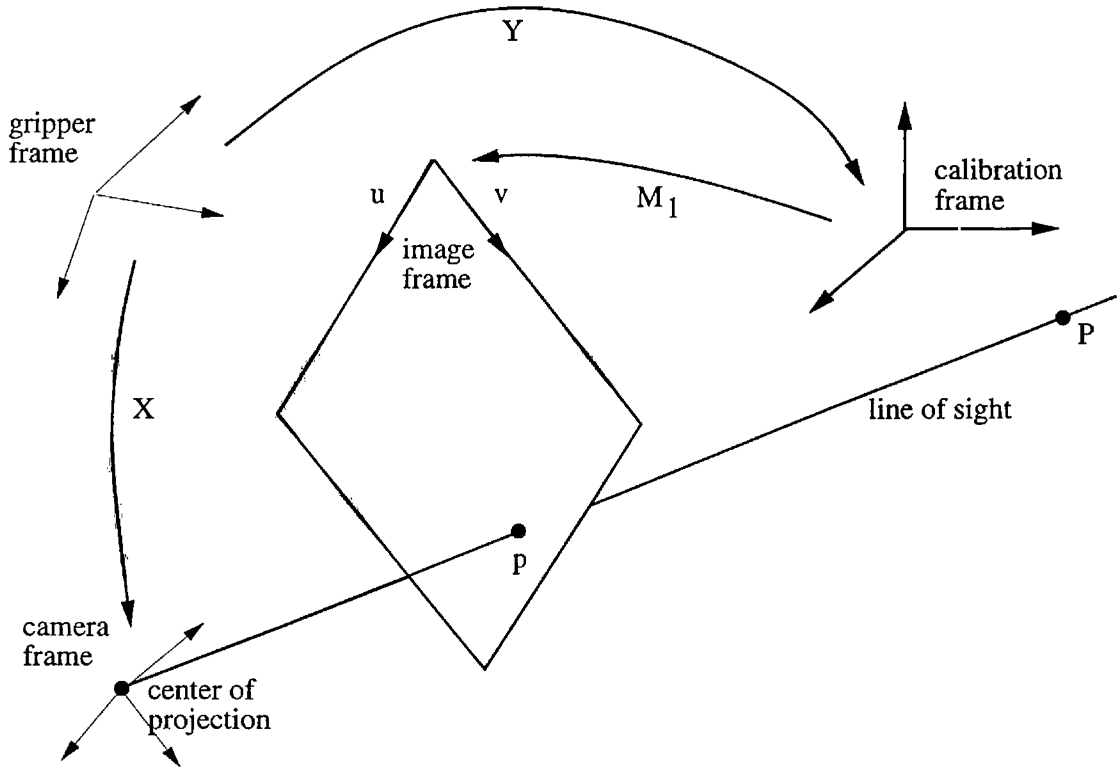

Fig. 2. The line of sight passing through the center of projection and the image point p may well be expressed in the calibration frame, using the coefficients of the perspective matrix M_1.

and:

$$(\mathbf{R_A} - \mathbf{I})\mathbf{t_X} = \mathbf{R_X}\mathbf{t_B} - \mathbf{t_A} \tag{16}$$

In this equation $\mathbf{I}$ is the 3×3 identity matrix.

To solve eq. (15) one may take advantage of the particular algebraic and geometric properties of rotation (orthogonal) matrices. Indeed, this equation can be written as:

$$\mathbf{R_A} = \mathbf{R_X}\mathbf{R_B}\mathbf{R_X}^T, \tag{17}$$

which is a similarity transformation, since $\mathbf{R_X}$ is an orthogonal matrix. Hence, matrices $\mathbf{R_A}$ and $\mathbf{R_B}$ have the same eigenvalues. A well-known property of a rotation matrix is that it has one of its eigenvalues equal to 1. Let $\mathbf{n_B}$ be the eigenvector of $\mathbf{R_B}$ associated with this eigenvalue. By postmultiplying eq. (15) with $\mathbf{n_B}$, we obtain:

$$\begin{aligned}\mathbf{R_A}\mathbf{R_X}\mathbf{n_B} &= \mathbf{R_X}\mathbf{R_B}\mathbf{n_B}\\ &= \mathbf{R_X}\mathbf{n_B}\end{aligned}$$

and we conclude that $\mathbf{R_X}\mathbf{n_B}$ is equal to $\mathbf{n_A}$, the eigenvector of $\mathbf{R_A}$ associated with the unit eigenvalue:

$$\mathbf{n_A} = \mathbf{R_X}\mathbf{n_B} \tag{18}$$

To conclude, solving for $\mathbf{AX} = \mathbf{XB}$ is equivalent to solving for eq. (18) and eq. (16). Solutions have been proposed by, among others, Shiu and Ahmad (1989), Tsai and Lenz (1989), Chou and Kamel (1991), and Wang (1992). All these authors noticed that at least three positions are necessary to uniquely determine $\mathbf{X}$ (i.e., $\mathbf{R_X}$ and $\mathbf{t_X}$). Shiu and Ahmad cast the rotation determination problem into the problem of solving for eight linear equations in four unknowns; they used standard linear algebra techniques to obtain a solution.

Tsai and Lenz (1989) suggested representing $\mathbf{R_X}$ by its unit eigenvector $\mathbf{n_X}$ and an angle θ_X. They noticed that:

$$\mathbf{n_X} \cdot (\mathbf{n_A} - \mathbf{n_B}) = 0$$

and

$$(\mathbf{n_A} - \mathbf{n_B}) \cdot (\mathbf{n_A} + \mathbf{n_B}) = 0$$

These expressions allow one to cast eq. (18) into:

$$(\mathbf{n_A} + \mathbf{n_B}) \times \mathbf{n} = \mathbf{n_A} - \mathbf{n_B} \tag{19}$$

with:

$$\mathbf{n} = \left(\tan\frac{\theta_X}{2}\right)\mathbf{n_X}$$

It is easy to see that eq. (19) is rank deficient, and hence at least two independent hand-eye motions (at least three positions) are necessary to determine a unique solution. In the general case of n motions ($n+1$ positions of the hand-eye device relative to the calibration frame), one may solve for an overconstrained set of $2n$ linear equations in three unknowns.

Chou and Kamel (1991) suggested representing rotation by a unit quaternion and then used the singular value decomposition method to solve the linear algebra. The idea of using a unit quaternion is a good one. Unfortunately, the authors were not aware of the closed-form optimal solution available for determining rotation with unit quaternions, as was proposed by both Horn (1987) and Faugeras and Hebert (1986).

Wang (1992) suggested three methods that roughly correspond to the solution proposed by Tsai and Lenz (1989). He compared his best method to the methods proposed by Shiu and Ahmad (1989) and by Tsai and Lenz and concluded that the Tsai and Lenz method yielded the best results.

Chen (1991) showed that the hand-eye geometry can be conveniently described using a screw representation for rotation and translation. This representation allows a uniqueness analysis.

All these approaches have the following features in common:

1. Rotation is decoupled from translation.
2. The solution for rotation is estimated using linear algebra techniques.
3. The solution for translation is estimated using linear algebra as well.

Decoupling rotation and translation is certainly a good idea; it leads to simple numerical solutions. However, in the presence of errors, the linear problem to be solved becomes ill-conditioned, and the solution available with the linear system is not valid. This is due to the fact that the geometric properties allowing the linearization of the rotation equation do not hold in the presence of noise. Errors may be due both to camera calibration inaccuracies and to inexact knowledge of the robot's kinematic parameters.

4. Decomposing the New Formulation

In this section we show that the new formulation introduced in Section 2.2 has a mathematical structure identical to that of the classic formulation. This will allow us to formulate a unified approach that solves for either of the two formulations.

We start by making explicit the 3×4 perspective matrix $\mathbf{M}$ as a function of intrinsic and extrinsic parameters; i.e., eq. (5):

$$\mathbf{M} = \begin{pmatrix} \alpha_u r_{11} + u_0 r_{31} & \alpha_u r_{12} + u_0 r_{32} & \alpha_u r_{13} + u_0 r_{33} & \alpha_u t_x + u_0 t_z \\ \alpha_v r_{21} + v_0 r_{31} & \alpha_v r_{22} + v_0 r_{32} & \alpha_v r_{23} + v_0 r_{33} & \alpha_v t_y + v_0 t_z \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix}$$

Notice that a matrix $\mathbf{M}_i$ of this form can be written as:

$$\mathbf{M}_i = \begin{pmatrix} \mathbf{N}_i & \mathbf{n}_i \end{pmatrix}$$

where $\mathbf{N}_i$ is a 3×3 matrix and $\mathbf{n}_i$ is a 3-vector. One may notice that in the general case $\mathbf{N}_i$ has an inverse, as the vectors $(\mathbf{r}_{11}\ \mathbf{r}_{12}\ \mathbf{r}_{13})^T$, $(\mathbf{r}_{21}\ \mathbf{r}_{22}\ \mathbf{r}_{23})^T$, and $(\mathbf{r}_{31}\ \mathbf{r}_{32}\ \mathbf{r}_{33})^T$ are mutually orthogonal and $\alpha_u \neq 0$, $\alpha_v \neq 0$. With this notation eq. (7) may be decomposed into a matrix equation,

$$\mathbf{N}_2\mathbf{R}_\mathbf{Y} = \mathbf{N}_1\mathbf{R}_\mathbf{Y}\mathbf{R}_\mathbf{B}, \tag{20}$$

and a vector equation,

$$\mathbf{N}_2\mathbf{t}_\mathbf{Y} + \mathbf{n}_2 = \mathbf{N}_1\mathbf{R}_\mathbf{Y}\mathbf{t}_\mathbf{B} + \mathbf{N}_1\mathbf{t}_\mathbf{Y} + \mathbf{n}_1. \tag{21}$$

Introducing the notation

$$\mathbf{N} = \mathbf{N}_1^{-1}\mathbf{N}_2,$$

eq. (20) becomes

$$\mathbf{N}\mathbf{R}_\mathbf{Y} = \mathbf{R}_\mathbf{Y}\mathbf{R}_\mathbf{B} \tag{22}$$

or

$$\mathbf{N} = \mathbf{R}_\mathbf{Y}\mathbf{R}_\mathbf{B}\mathbf{R}_\mathbf{Y}^T.$$

Two properties of $\mathbf{N}$ may be easily derived:

1. $\mathbf{N}$ is the product of three rotation matrices; it is therefore a rotation itself and

$$\mathbf{N}^{-1} = \mathbf{N}^T.$$

2. Because $\mathbf{R}_\mathbf{Y}$ is an orthogonal matrix, the above equation defines a similarity transformation. It follows that $\mathbf{N}$ has the same eigenvalues as $\mathbf{R}_\mathbf{B}$. In particular, $\mathbf{R}_\mathbf{B}$ has an eigenvalue equal to 1 and let $\mathbf{n}_\mathbf{B}$ be the eigenvector associated with this eigenvalue.

If we denote by $\mathbf{n}_\mathbf{N}$ the eigenvector of $\mathbf{N}$ associated with the unit eigenvalue, then we obtain:

$$\begin{aligned} \mathbf{N}\mathbf{R}_\mathbf{Y}\mathbf{n}_\mathbf{B} &= \mathbf{R}_\mathbf{Y}\mathbf{R}_\mathbf{B}\mathbf{n}_\mathbf{B} \\ &= \mathbf{R}_\mathbf{Y}\mathbf{n}_\mathbf{B} \end{aligned}$$

and hence we have

$$\mathbf{n}_\mathbf{N} = \mathbf{R}_\mathbf{Y}\mathbf{n}_\mathbf{B}. \tag{23}$$

This equation is identical to eq. (18) in the classic formulation.

By premultiplying eq. (21) with $\mathbf{N}_1^{-1}$, we obtain

$$(\mathbf{N} - \mathbf{I})\mathbf{t_Y} = \mathbf{R_Y t_B} - \mathbf{t_N}, \tag{24}$$

with

$$\mathbf{t_N} = \mathbf{N}_1^{-1}(\mathbf{n}_2 - \mathbf{n}_1),$$

and one may easily notice that this equation is identical to eq. (16) in the classic formulation.

To conclude, the classic formulation decomposes in eqs. (18) and (16) and, equivalently, the new formulation decomposes in eqs. (23) and (24).

5. A Unified Optimal Solution

In the previous sections we showed that the classic and the new formulations are mathematically equivalent. Indeed, the classic formulation, $\mathbf{AX} = \mathbf{XB}$, decomposes into eqs. (18) and (16):

$$\mathbf{n_A} = \mathbf{R_X n_B}$$
$$(\mathbf{R_A} - \mathbf{I})\mathbf{t_X} = \mathbf{R_X t_B} - \mathbf{t_A}$$

and the new formulation, $\mathbf{MY} = \mathbf{M'YB}$ decomposes into eqs. (23) and (24):

$$\mathbf{n_N} = \mathbf{R_Y n_B}$$
$$(\mathbf{N} - \mathbf{I})\mathbf{t_Y} = \mathbf{R_Y t_B} - \mathbf{t_N}$$

These two sets of equations are of the form:

$$\mathbf{v}' = \mathbf{R}\mathbf{v} \tag{25}$$
$$(\mathbf{K} - \mathbf{I})\mathbf{t} = \mathbf{R}\mathbf{p} - \mathbf{p}' \tag{26}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the parameters to be estimated (rotation and translation); $\mathbf{v}'$, $\mathbf{v}$, $\mathbf{p}'$, $\mathbf{p}$ are 3-vectors; $\mathbf{K}$ is a 3×3 orthogonal matrix; and $\mathbf{I}$ is the identity matrix.

Eqs. (25) and (26) are associated with one motion of the hand-eye device. To estimate $\mathbf{R}$ and $\mathbf{t}$, at least two such motions are necessary. In the general case of n motions, one may cast the problem of solving $2n$ such equations into the problem of minimizing two positive error functions:

$$f_1(\mathbf{R}) = \sum_{i=1}^{n} \|\mathbf{v}'_i - \mathbf{R}\mathbf{v}_i\|^2 \tag{27}$$

and

$$f_2(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{n} \|\mathbf{R}\mathbf{p}_i - (\mathbf{K}_i - \mathbf{I})\mathbf{t} - \mathbf{p}'_i\|^2. \tag{28}$$

Therefore, two approaches are possible:

1. $\mathbf{R}$ *then* $\mathbf{t}$. Rotation is estimated first by minimizing f_1. This minimization problem has a simple closed-form solution that will be detailed later. Once the optimal rotation is determined, the minimization of f_2 over the translational parameters is a linear least-squared problem.
2. $\mathbf{R}$ *and* $\mathbf{t}$. Rotation and translation are estimated simultaneously by minimizing $f_1 + f_2$. This minimization problem is nonlinear but, as it will be shown later, it provides the most stable solution.

5.1. Rotation Then Translation

To minimize f_1 given by eq. (27), we represent rotation by a unit quaternion. With this representation one may write (see Appendix A and Faugeras and Hebert [1986]):

$$\mathbf{R}\mathbf{v}_i = \mathbf{q} * \mathbf{v}_i * \bar{\mathbf{q}}$$

Moreover, using eq. (33), one may successively write:

$$\begin{aligned}\|\mathbf{v}'_i - \mathbf{q} * \mathbf{v}_i * \bar{\mathbf{q}}\|^2 &= \|\mathbf{v}'_i - \mathbf{q} * \mathbf{v}_i * \bar{\mathbf{q}}\|^2 \|\mathbf{q}\|^2 \\ &= \|\mathbf{v}'_i * \mathbf{q} - \mathbf{q} * \mathbf{v}_i\|^2 \\ &= (\mathbf{Q}(\mathbf{v}'_i)\mathbf{q} - \mathbf{W}(\mathbf{v}_i)\mathbf{q})^{\mathrm{T}}(\mathbf{Q}(\mathbf{v}'_i)\mathbf{q} - \mathbf{W}(\mathbf{v}_i)\mathbf{q}) \\ &= \mathbf{q}^{\mathrm{T}} \mathcal{A}_i \mathbf{q}\end{aligned}$$

with $\mathcal{A}_i$ being a 4×4 positive symmetric matrix:

$$\mathcal{A}_i = (\mathbf{Q}(\mathbf{v}'_i) - \mathbf{W}(\mathbf{v}_i))^{\mathrm{T}}(\mathbf{Q}(\mathbf{v}'_i) - \mathbf{W}(\mathbf{v}_i)).$$

Finally the error function becomes

$$\begin{aligned}f_1(\mathbf{R}) &= f_1(\mathbf{q}) \\ &= \sum_{i=1}^{n} \|\mathbf{v}'_i - \mathbf{q} * \mathbf{v}_i * \bar{\mathbf{q}}\|^2 \\ &= \sum_{i=1}^{n} \mathbf{q}^{\mathrm{T}} \mathcal{A}_i \mathbf{q} \\ &= \mathbf{q}^{\mathrm{T}} \left(\sum_{i=1}^{n} \mathcal{A}_i \right) \mathbf{q} \\ &= \mathbf{q}^{\mathrm{T}} \mathcal{A} \mathbf{q}\end{aligned} \tag{29}$$

with $\mathcal{A} = \sum_{i=1}^{n} \mathcal{A}_i$. One must minimize f_1 under the constraint that $\mathbf{q}$ must be a unit quaternion. This constrained minimization problem can be solved using the Lagrange multiplier:

$$\min_q f_1 = \min_q (\mathbf{q}^{\mathrm{T}} \mathcal{A} \mathbf{q} + \lambda(1 - \mathbf{q}^{\mathrm{T}}\mathbf{q})).$$

By differentiating this error function with respect to $\mathbf{q}$, one may easily find the solution in closed form:

$$\mathcal{A}\mathbf{q} = \lambda \mathbf{q}.$$

The unit quaternion minimizing f_1 is therefore the eigenvector of $\mathcal{A}$ associated with its smallest (positive) eigenvalue. This closed-form solution was introduced by Faugeras and Hebert (1986) for finding the best rotation between two sets of 3D features.

Once the rotation has been determined, the problem of determining the best translation becomes a linear least-squares problem that can be easily solved using standard linear algebra techniques.

5.2. Rotation and Translation

The problem of estimating rotation and translation simultaneously can be stated in terms of the following minimization problem:

$$\min_{q,t}(f_1 + f_2).$$

We have been unable to solve this problem in closed form. One may notice that the error function to be minimized is a sum of squares of nonlinear functions. Because of the special structure of the Jacobian and Hessian matrices associated with error functions of this type, a number of special minimization methods have been designed specifically to deal with this case (Gill et al. 1989). Among these methods, the Levenberg-Marquardt method and the trust-region method (Fletcher 1990; Phong et al. in press, 1995) are good candidates.

Using unit quaternions, the error function to be minimized is

$$\min_{q,t}(f(\mathbf{q},\mathbf{t}) + \lambda(1 - \mathbf{q}^{\mathrm{T}}\mathbf{q})^2), \tag{30}$$

with:

$$\begin{aligned} f(\mathbf{q},\mathbf{t}) &= \lambda_1 f_1(\mathbf{q}) + \lambda_2 f_2(\mathbf{q},\mathbf{t}) \\ &= \lambda_1 \sum_{i=1}^{n} \|\mathbf{v}_i' - \mathbf{q} * \mathbf{v}_i * \overline{\mathbf{q}}\|^2 \\ &\quad + \lambda_2 \sum_{i=1}^{n} \|\mathbf{q} * \mathbf{p}_i * \overline{\mathbf{q}} - (\mathbf{K}_i - \mathbf{I})\mathbf{t} - \mathbf{p}_i'\|^2, \end{aligned}$$

which has the form of sum of squares of nonlinear functions, and $\lambda(1 - \mathbf{q}^{\mathrm{T}}\mathbf{q})^2$ is a penalty function that guarantees that $\mathbf{q}$ (a quaternion) has a module equal to 1. λ_1 and λ_2 are two weights, and λ is a real positive number. High values for λ ensure that the module of $\mathbf{q}$ is close to 1. In all our experiments we have:

$$\begin{aligned} \lambda_1 = \lambda_2 &= 1 \\ \lambda &= 2 \times 10^6 \end{aligned}$$

There are two possibilities for solving the nonlinear minimization problem of equation (30). The first possibility is to consider it as a classic nonlinear least squares minimization problem and to apply standard nonlinear optimization techniques, such as Newton's method and Newton-like methods (Gill et al. 1989; Fletcher 1990). In the next two sections we give some results obtained with the Levenberg-Marquardt nonlinear minimization method as described in Press et al. (1988).

The second possibility is to try to simplify the expression of the error function to be minimized. Using properties associated with quaternions, the error function may indeed be simplified. We already obtained a simple analytic form for f_1 (i.e., eq. (29)). Similarly, f_2 simplifies as well.

Indeed, f_2 is the sum of terms such as

$$\|\mathbf{q} * \mathbf{p}_i * \overline{\mathbf{q}} - (\mathbf{K}_i - \mathbf{I})\mathbf{t} - \mathbf{p}_i'\|^2,$$

and we have

$$\|\mathbf{q}*\mathbf{p}_i*\overline{\mathbf{q}}-(\mathbf{K}_i-\mathbf{I})\mathbf{t} - \mathbf{p}_i'\|^2\|\mathbf{q}\|^2 = \|\mathbf{q}*\mathbf{p}_i-(\mathbf{K}_i-\mathbf{I})\mathbf{t}*\mathbf{q} - \mathbf{p}_i'*\mathbf{q}\|^2.$$

Using the matrix representation for quaternion multiplication, one can easily obtain (see Appendix B for the derivation of this equation)

$$\begin{aligned} f_2(\mathbf{q},\mathbf{t}) = \mathbf{q}^{\mathrm{T}}(\sum_{i=1}^{n} \mathcal{B}_i)\mathbf{q} + \mathbf{t}^{\mathrm{T}}(\sum_{i=1}^{n} \mathcal{C}_i)\mathbf{t} + (\sum_{i=1}^{n} \boldsymbol{\delta}_i)\mathbf{t} \\ + (\sum_{i=1}^{n} \boldsymbol{\varepsilon}_i)\mathbf{Q}(\mathbf{q})^{\mathrm{T}}\mathbf{W}(\mathbf{q})\mathbf{t}. \end{aligned} \tag{31}$$

The 4×4 matrices $\mathcal{B}_i$ and $\mathcal{C}_i$ and the 1×4 vectors $\boldsymbol{\delta}_i$ and $\boldsymbol{\varepsilon}_i$ are:

$$\begin{aligned} \mathcal{B}_i &= (\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_i + \mathbf{p}_i'^{\mathrm{T}}\mathbf{p}_i')\mathbf{I} - \mathbf{W}(\mathbf{p}_i)^{\mathrm{T}}\mathbf{Q}(\mathbf{p}_i') - \mathbf{Q}(\mathbf{p}_i')^{\mathrm{T}}\mathbf{W}(\mathbf{p}_i) \\ \mathcal{C}_i &= \mathbf{K}_i^{\mathrm{T}}\mathbf{K}_i - \mathbf{K}_i - \mathbf{K}_i^{\mathrm{T}} + \mathbf{I} \\ \boldsymbol{\delta}_i &= 2\mathbf{p}_i'^{\mathrm{T}}(\mathbf{K}_i - \mathbf{I}) \\ \boldsymbol{\varepsilon}_i &= -2\mathbf{p}_i^{\mathrm{T}}(\mathbf{R}_{\mathbf{B}_i} - \mathbf{I}) \end{aligned}$$

With the notations $\mathcal{B} = \sum_{i=1}^{n} \mathcal{B}_i$, $\mathcal{C} = \sum_{i=1}^{n} \mathcal{C}_i$, $\boldsymbol{\delta} = \sum_{i=1}^{n} \boldsymbol{\delta}_i$, $\boldsymbol{\varepsilon} = \sum_{i=1}^{n} \boldsymbol{\varepsilon}_i$, and with $\mathcal{A}$ already defined, we obtain the following nonlinear minimization problem:

$$\min_{q,t}(\mathbf{q}^{\mathrm{T}}(\mathcal{A}+\mathcal{B})\mathbf{q} + \mathbf{t}^{\mathrm{T}}\mathcal{C}\mathbf{t}+\boldsymbol{\delta}\mathbf{t}+\boldsymbol{\varepsilon}\mathbf{Q}(\mathbf{q})^{\mathrm{T}}\mathbf{W}(\mathbf{q})\mathbf{t}+\lambda(1-\mathbf{q}^{\mathrm{T}}\mathbf{q})^2), \tag{32}$$

which is the sum of five terms. The number of parameters to be estimated is seven (four for the unit quaternion and three for the translation). It is worthwhile to notice that the number of terms of this error function is constant with respect to n (i.e., the number of hand-eye motions). For such minimization problems, one may use constrained step methods such as the trust-region method (Fletcher 1990; Yassine 1989).

6. Stability Analysis and Method Comparison

One of the most important qualities of any hand-eye calibration method is its stability with respect to perturbations. There are two main sources of perturbations:

errors associated with camera calibration and errors associated with robot motion. Indeed, the parameters of both the direct and inverse kinematic models of robots are not perfect. As a consequence the real motions associated with the hand and camera are only known up to some uncertainty. It follows that the estimation of the hand-eye transformation has errors associated with it, and it is important to quantify these to determine the stability of a given method and to compare various methods.

To perform this stability analysis we developed the following method:

1. Nominal values for the parameters of the hand-eye transformation ($\mathbf{X}$ or $\mathbf{Y}$) are provided.
2. Also provided are $n+1$ matrices $\mathbf{A}_1, \ldots \mathbf{A}_{n+1}$ from which n hand motions can be computed either with (see Section 2)
$$\mathbf{B}_{i\,i+1} = \mathbf{X}^{-1}\mathbf{A}_{i+1}\mathbf{A}_i^{-1}\mathbf{X}$$
or with
$$\mathbf{B}_{1i} = \mathbf{Y}^{-1}\mathbf{A}_1^{-1}\mathbf{A}_i\mathbf{Y}.$$
3. Gaussian or uniform noise is added to both the camera and hand motions, and $\mathbf{X}$ (or $\mathbf{Y}$) is estimated in the presence of this noise using three different methods: Tsai-Lenz, closed-form solution, and nonlinear optimization.
4. We study the variations of the estimated hand-eye transformation as a function of the noise added and/or as a function of the number of motions (n).

Because 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 the text that follows the level of noise is represented as a ratio: the values of the actual random numbers divided by the values of the perturbed parameters.

In the case of a rotation, the vector associated with this rotation has a modulus 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:

$$\|\mathbf{t}_{\text{nominal}}\| = \frac{\sum_{i=1}^{n}(\|\mathbf{t}_{\mathbf{A}_{ii+1}}\| + \|\mathbf{t}_{\mathbf{B}_{ii+1}}\|)}{2n},$$

where $\mathbf{t}_{\mathbf{A}_{ii+1}}$ is the translation vector associated with $\mathbf{A}_{ii+1}$.

For each noise level and for a large number J of trials we compute the errors associated with rotation and translation as follows:

$$e_{\text{rot}} = \sqrt{\frac{1}{J}\sum_{j=1}^{J}\|\tilde{\mathbf{R}}_j - \mathbf{R}\|^2}$$

and

$$e_{\text{tr}} = \frac{\sqrt{\frac{1}{J}\sum_{j=1}^{J}\|\tilde{\mathbf{t}}_j - \mathbf{t}\|^2}}{\|\mathbf{t}\|},$$

where $\mathbf{R}$ and $\mathbf{t}$ are the nominal values of the transformation being estimated ($\mathbf{X}$ or $\mathbf{Y}$), $\tilde{\mathbf{R}}_j$ and $\tilde{\mathbf{t}}_j$ are the estimated rotation and translation for some trial j, and J is the number of trials for each noise level (defined either by C or by σ). In all our experiments we set $J = 1000$ mm and $\|\mathbf{t}\| = 157$ mm.

The following figures show the above errors as a function of the percentage of noise. The noise varies from 0.5% to 6%. The full curves (—) correspond to the method of Tsai and Lenz (1989), the dotted curves (...) correspond to the closed-form method and the dashed curves (- - -) correspond to the nonlinear method. Figures 3 through 10 correspond to two motions ($n = 2$) of the hand-eye device, whereas in Figures 11 and 12 the number of motions varies from 2 to 9.

Figures 3 and 4 show the rotation and translation errors as a function of uniform noise added to the rotational part of the hand and camera motions. Figures 5 and 6 show the rotation and translation errors as a function of

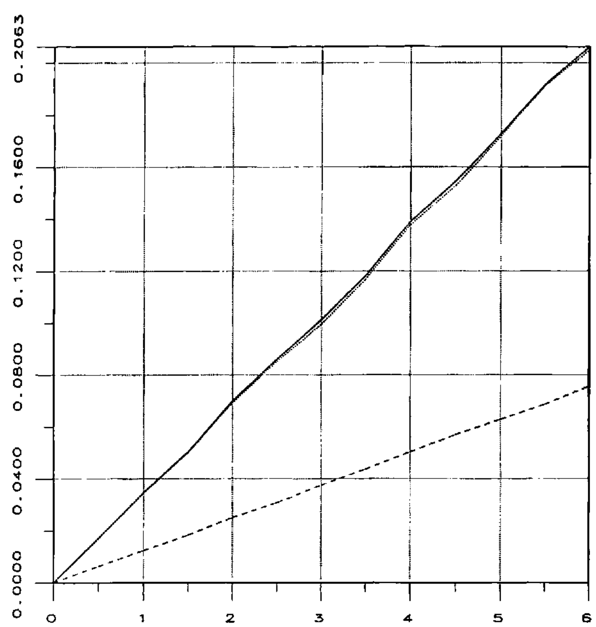

Fig. 3. Error in rotation in the presence of uniform noise perturbing the rotation axes.

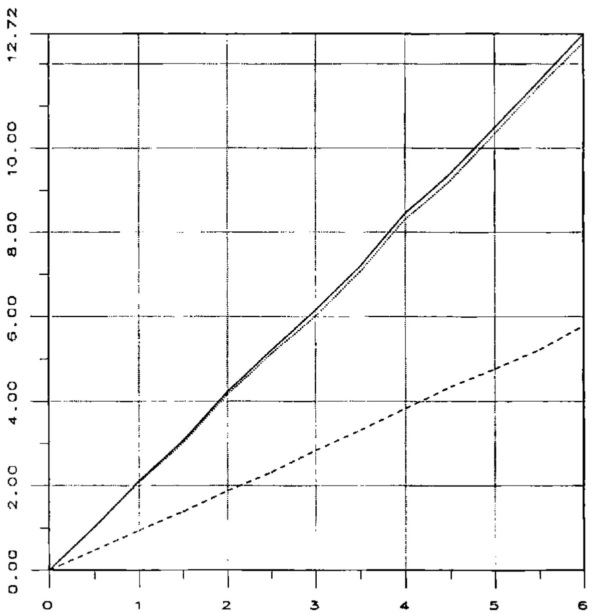

Fig. 4. Error in translation in the presence of uniform noise perturbing the rotation axes.

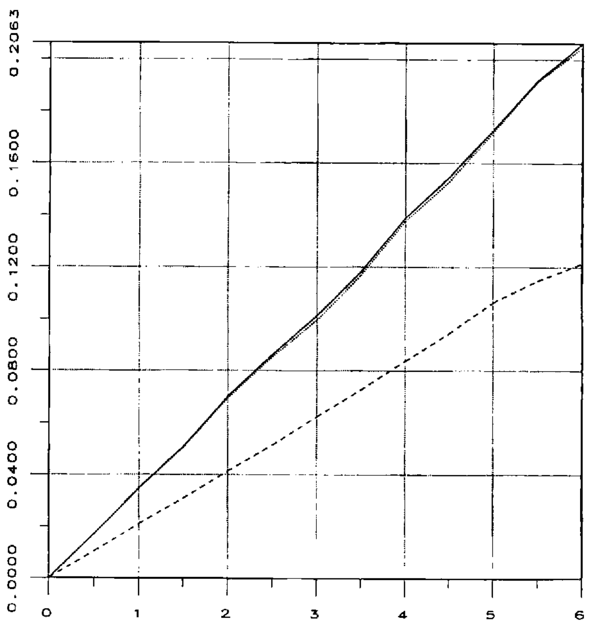

Fig. 5. Error in rotation in the presence of uniform noise perturbing the translation vectors and the rotation axes.

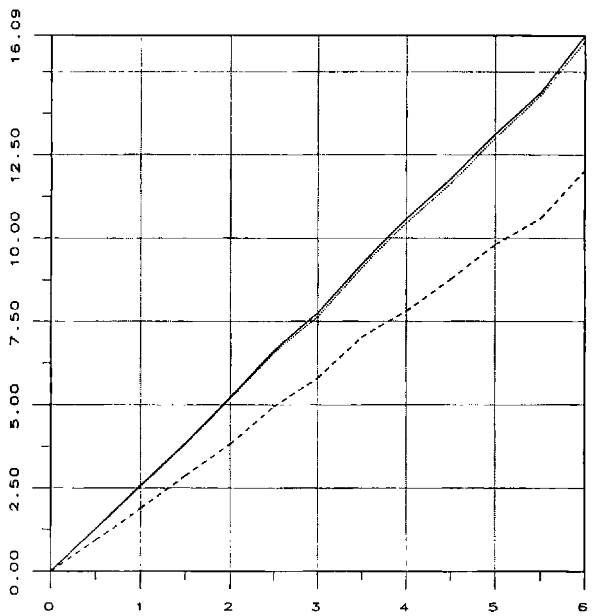

Fig. 6. Error in translation in the presence of uniform noise perturbing the translation vectors and the rotation axes.

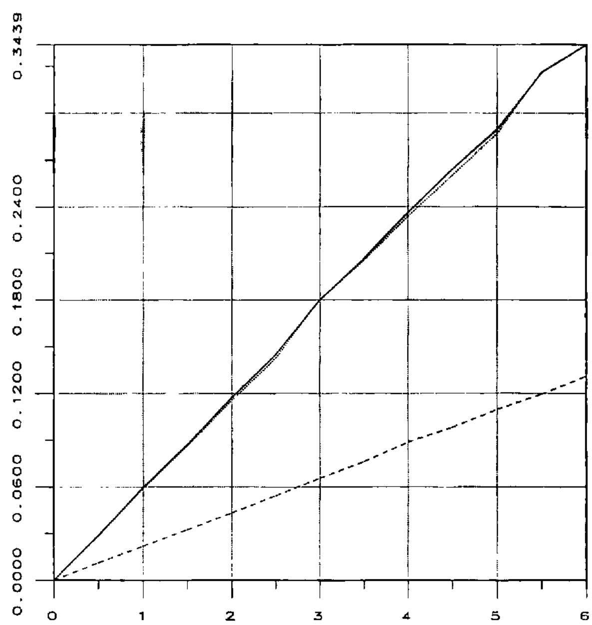

Fig. 7. Error in rotation in the presence of Gaussian noise perturbing the rotation axes.

uniform noise added to both the rotational and the translational parts of the camera and hand motions. Figures 7 through 10 are similar to Figures 3 through 6, but the uniform distribution of the noise has been replaced by a Gaussian distribution.

It is interesting to notice that the Tsai-Lenz and closed-form methods have almost the same behavior, whereas the nonlinear method provides more accurate results in all the situations. The fact that the results obtained with the first two methods are highly correlated may be due to the fact that both of these methods decouple the estimation of rotation from the estimation of translation. This behavior seems to be independent with respect to the noise type (uniform or Gaussian) and to whether only rotation is perturbed or rotation and translation are perturbed

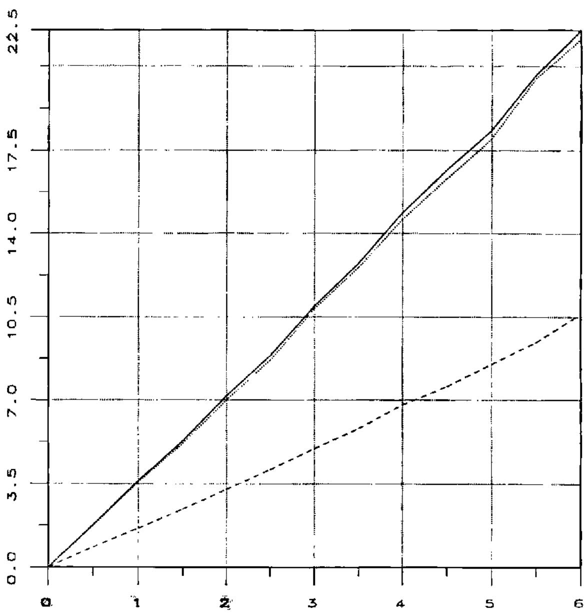

Fig. 8. Error in translation in the presence of Gaussian noise perturbing the rotation axes.

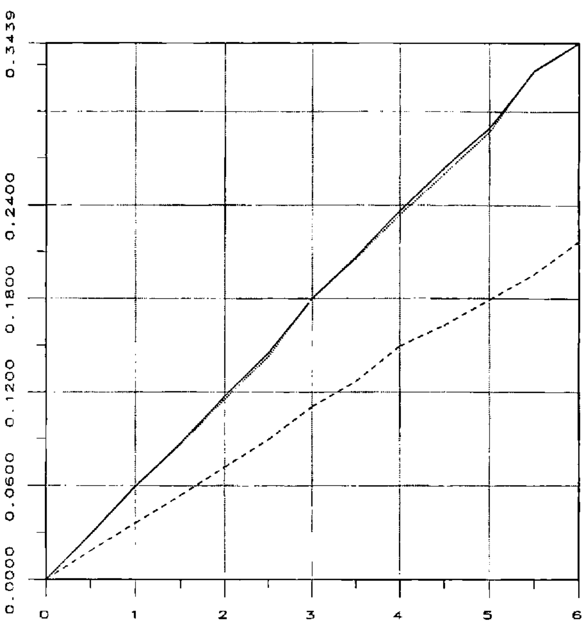

Fig. 9. Error in rotation in the presence of Gaussian noise perturbing the translation vectors and the rotation axes.

simultaneously. We conclude that the decoupling of rotation and translation introduces a bias in the estimation of the hand-eye transformation.

As other authors have done in the past, it is interesting to analyze the behavior of hand-eye calibration with respect to the number of motions. To perform this analysis we have to fix the percentage of noise. Figures 11 and 12

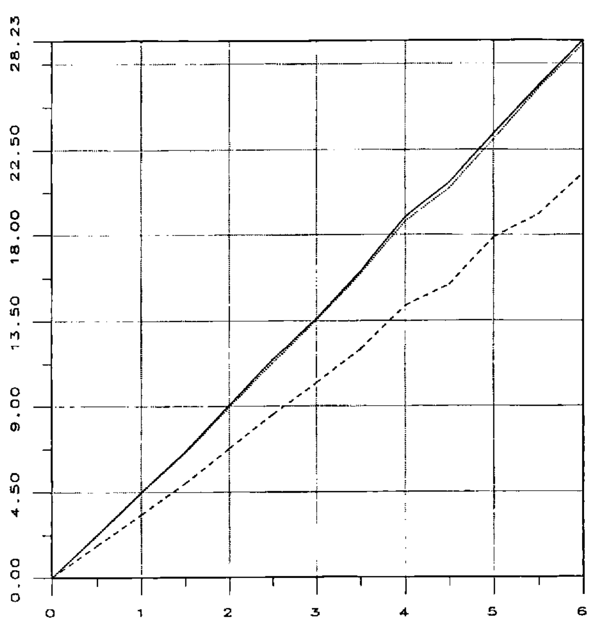

Fig. 10. Error in translation in the presence of Gaussian noise perturbing the translation vectors and the rotation axes.

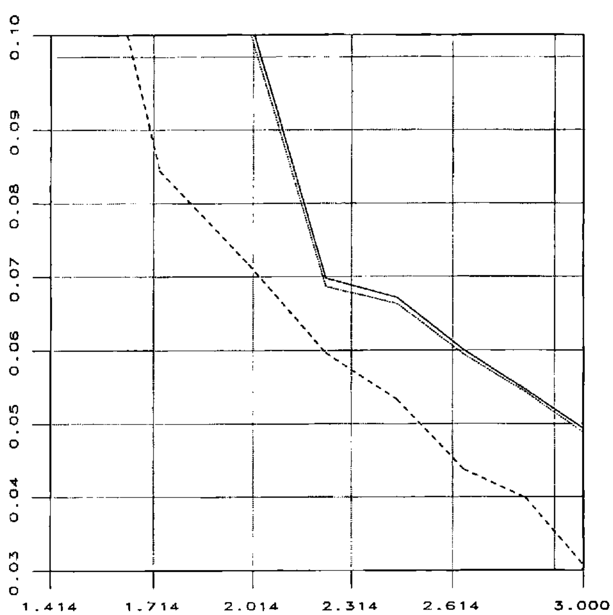

Fig. 11. Error in rotation as a function of the number of motions.

show the rotational and translational errors as a function of the square root of the number of motions ($\sqrt{n}$ varies from 1.414 to 3). The noise ratio has been fixed to 6% (the worst case for rotations) and to 2% for translations. Both rotational and translational noise distributions are Gaussian. For example, for four motions, the error in

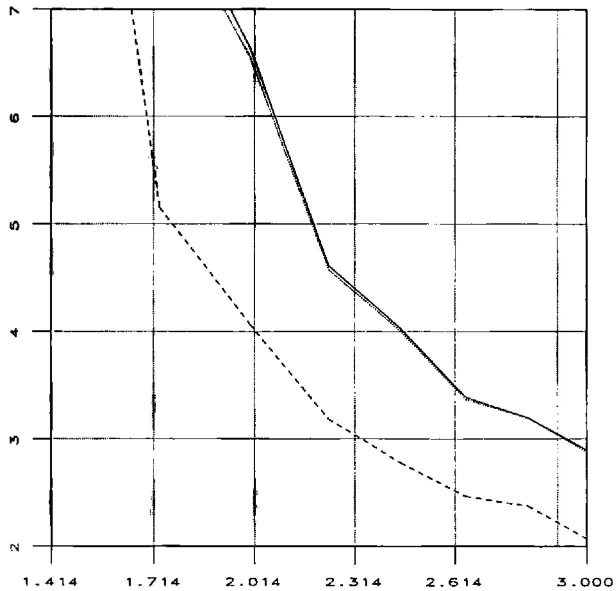

Fig. 12. Error in translation as a function of the number of motions.

translation is 4% for the nonlinear method and 6.5% for the other two methods.

7. Experimental Results

In this section we report some experimental results obtained with three sets of data. The first set was provided by François Chaumette from the Institut de Recherche en Informatique et Systèmes aléatoires (IRISA) and the second and third data sets were obtained at the Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle (LIFIA). The first set was obtained with 17 different positions of the hand-eye device with respect to a calibrating object. The second set was obtained with seven such positions. The third set was obtained with six positions. For the first set, only the extrinsic camera parameters were provided, whereas for the latter sets, we had access to the full 3×4 perspective matrices. Therefore, the latter sets allowed us to test both the classic and the new formulations. The only restriction imposed on the robot motions is the requirement that the camera must see the calibration pattern in each of its positions.

To calibrate the camera we used the method proposed by Faugeras and Toscani (1986) and the following setup. The calibration pattern consists of a planar grid of size 200×300 mm that can move along an axis perpendicular to its plane. The distance from this grid to the camera varies between 600 mm and 1000 mm during hand-eye calibration. This setup, combined with the Faugeras-Toscani method, provides very accurate calibration data. This is mainly due to the accuracy of the grid points (0.1 mm), the accuracy of point localization in the image (0.1 pixels), and the large number of calibration points being used (460 points). Hence, camera calibration errors can be neglected relative to robot calibration errors (see later discussion).

Our tests compare the classic Tsai-Lenz method with the two methods developed in this article. Tables 1, 2, and 3 summarize the results obtained with the three data sets mentioned above. The lengths of the translation vectors thus obtained are $\|\mathbf{t_X}\| = 93$ mm and $\|\mathbf{t_Y}\| = 681$ mm. The second columns of these tables show the sum of squares of the absolute error in rotation. The third columns show the sum of squares of the relative error in translation.

These experimental results seem to confirm that, on one hand, the nonlinear method provides a better estimation of the translation vector—at the cost of a sometimes slightly less good rotation[1]—and, on the other hand, the new formulation provides a better estimation of the transformation parameters than the classic formulation.

Although the nonlinear technique provides the most accurate results with simulated data, the linear and closed-form techniques sometimes provide a better estimation of rotation with real data. 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. The linear and closed-form techniques estimate the rotation parameters independently of the robot's translation parameters, and therefore the rotation thus estimated is not affected by translation errors. However, in practice we prefer the nonlinear technique.

8. Discussion

In this article we attacked the problem of hand-eye calibration. In addition to the classic formulation (i.e., $\mathbf{AX} = \mathbf{XB}$), we suggest a new formulation that directly uses the 3×4 perspective matrices available with camera calibration ($\mathbf{MY} = \mathbf{M'YB}$). The advantage of the new formulation with respect to the classic one is that it avoids the decomposition of the perspective matrix into intrinsic and extrinsic camera parameters. Indeed, it has long been recognized in computer vision research that this decomposition is unstable.

Moreover, we show that the new formulation has a

1. However, one must be aware of the fact that smaller error does not imply more accurate. The best way to measure accuracy is the width of a minimum and not its depth.

Table 1. The Classic Formulation Used With the First Data Set

$\mathbf{AX = XB}$	$\sum \|\|\mathbf{R_A R_X} - \mathbf{R_X R_B}\|\|^2$	$\frac{\sum \|\|(\mathbf{R_A} - \mathbf{I})\mathbf{t_X} - \mathbf{R_X t_B} + \mathbf{t_A}\|\|^2}{\sum \|\|\mathbf{R_X t_B} - \mathbf{t_A}\|\|^2}$
Tsai-Lenz	0.0006	0.032
Closed-form solution	0.0005	0.029
Nonlinear optimization	0.0003	0.019

Table 2. The Classic and the New Formulations Used With the Second Data Set

$\mathbf{AX = XB}$	$\sum \|\|\mathbf{R_A R_X} - \mathbf{R_X R_B}\|\|^2$	$\frac{\sum \|\|(\mathbf{R_A} - \mathbf{I})\mathbf{t_X} - \mathbf{R_X t_B} + \mathbf{t_A}\|\|^2}{\sum \|\|\mathbf{R_X t_B} - \mathbf{t_A}\|\|^2}$
Tsai-Lenz	0.0014	0.036
Closed-form solution	0.0014	0.023
Nonlinear optimization	0.0017	0.015
$\mathbf{MY = M'YB}$	$\sum \|\|\mathbf{NR_Y} - \mathbf{R_Y R_B}\|\|^2$	$\frac{\sum \|\|(\mathbf{N} - \mathbf{I})\mathbf{t_Y} - \mathbf{R_Y t_B} + \mathbf{t_N}\|\|^2}{\sum \|\|\mathbf{R_Y t_B} - \mathbf{t_N}\|\|^2}$
Tsai-Lenz	0.0031	0.0021
Closed-form solution	0.0015	0.001
Nonlinear optimization	0.0013	0.0006

Table 3. The Classic and the New Formulations Used With the Third Data Set

$\mathbf{AX = XB}$	$\sum \|\|\mathbf{R_A R_X} - \mathbf{R_X R_B}\|\|^2$	$\frac{\sum \|\|(\mathbf{R_A} - \mathbf{I})\mathbf{t_X} - \mathbf{R_X t_B} + \mathbf{t_A}\|\|^2}{\sum \|\|\mathbf{R_X t_B} - \mathbf{t_A}\|\|^2}$	CPU time*
Tsai-Lenz	0.014	0.23	0.08
Closed-form solution	0.036	0.223	0.06
Nonlinear optimization	0.258	0.058	0.21
$\mathbf{MY = M'YB}$	$\sum \|\|\mathbf{NR_Y} - \mathbf{R_Y R_B}\|\|^2$	$\frac{\sum \|\|(\mathbf{N} - \mathbf{I})\mathbf{t_Y} - \mathbf{R_Y t_B} + \mathbf{t_N}\|\|^2}{\sum \|\|\mathbf{R_Y t_B} - \mathbf{t_N}\|\|^2}$	CPU time*
Tsai-Lenz	0.038	0.039	0.06
Closed-form solution	0.035	0.037	0.08
Nonlinear optimization	0.04	0.034	0.25

* Indicated in seconds on a Sparc-10 Sun computer.

mathematical structure that is identical to the mathematical structure of the classic formulation. The advantage of this mathematical analogy is that for the two formulations, being variations of the same one, any method for solving the problem applies to both formulations.

We develop two resolution methods. The first one solves for rotation and then for translation, whereas the second one solves simultaneously for rotation and translation. Using unit quaternions to represent rotations, the first method leads to a closed-form solution introduced by Faugeras and Hebert (1986), whereas the second one is new and leads to nonlinear optimization. Among the many robust nonlinear optimization methods that are available, we chose the Levenberg-Marquardt technique.

Both the stability analysis and the results obtained with experimental data from two laboratories show that the nonlinear optimization method yields the best results. Linear algebra techniques (the Tsai-Lenz method) and the closed-form method (using unit quaternions) are of comparable accuracy.

The new formulation provides much more accurate hand-eye calibration results than the classic formulation. This improvement in accuracy seems to confirm that the decomposition of the perspective matrix into intrinsic and extrinsic parameters introduces some errors. Nevertheless, the intrinsic parameters, even if they do not need to be made explicit, are assumed to be constant during calibration. We are perfectly aware that this assumption is not very realistic and may cause problems in practice, and we are currently investigating ways to give up this assumption.

Also, we are investigating ways to perform hand-eye calibration and robot calibration simultaneously. Indeed, in many applications, such as nuclear and space environments, it may be useful to calibrate a robot simply by calibrating a camera mounted on the robot.

Appendix A: Rotation and Unit Quaternion

The use of unit quaternions to represent rotations is justified by an elegant closed-form solution associated with the problem of optimally estimating rotation from 3D to 3D vector correspondences (Faugeras and Hebert 1986; Horn 1987; Sabata and Aggarwal 1992; Horaud and Monga 1993). In Section 5 we stressed the similarity between the hand-eye calibration problem and the problem of optimally estimating the rotation between sets of 3D features. In this appendix we briefly recall the definition of quaternions, some useful properties of the quaternion multiplication operator, and the relationship between 3×3 orthogonal matrices and unit quaternions.

Quaternions are 4-vectors that may be viewed as a generalization of complex numbers to have one real part and three imaginary parts:

$$\mathbf{q} = q_0 + iq_x + jq_y + kq_z$$

with

$$i^2 = j^2 = k^2 = ijk = -1.$$

One may define quaternion multiplication (denoted by $*$) as follows:

$$\mathbf{r} * \mathbf{q} = (r_0 + ir_x + jr_y + kr_z)(q_0 + iq_x + jq_y + kq_z),$$

which can be written using a matrix notation:

$$\mathbf{r} * \mathbf{q} = \mathbf{Q}(\mathbf{r})\mathbf{q} = \mathbf{W}(\mathbf{q})\mathbf{r},$$

with

$$\mathbf{Q}(\mathbf{r}) = \begin{pmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & -r_z & r_y \\ r_y & r_z & r_0 & -r_x \\ r_z & -r_y & r_x & r_0 \end{pmatrix}$$

and

$$\mathbf{W}(\mathbf{r}) = \begin{pmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & r_z & -r_y \\ r_y & -r_z & r_0 & r_x \\ r_z & r_y & -r_x & r_0 \end{pmatrix}.$$

One may easily verify the following properties:

$$\begin{aligned} \mathbf{Q}(\mathbf{r})^T\mathbf{Q}(\mathbf{r}) = \mathbf{Q}(\mathbf{r})\mathbf{Q}(\mathbf{r})^T &= \mathbf{r}^T\mathbf{r}\mathbf{I} \\ \mathbf{W}(\mathbf{r})^T\mathbf{W}(\mathbf{r}) = \mathbf{W}(\mathbf{r})\mathbf{W}(\mathbf{r})^T &= \mathbf{r}^T\mathbf{r}\mathbf{I} \\ \mathbf{Q}(\mathbf{r})\mathbf{q} &= \mathbf{W}(\mathbf{q})\mathbf{r} \\ \mathbf{Q}(\mathbf{r})^T\mathbf{r} = \mathbf{W}(\mathbf{r})^T\mathbf{r} &= \mathbf{r}^T\mathbf{r}\mathbf{e} \\ \mathbf{Q}(\mathbf{r})\mathbf{Q}(\mathbf{q}) &= \mathbf{Q}(\mathbf{Q}(\mathbf{r})\mathbf{q}) \\ \mathbf{W}(\mathbf{r})\mathbf{W}(\mathbf{q}) &= \mathbf{W}(\mathbf{W}(\mathbf{r})\mathbf{q}) \\ \mathbf{Q}(\mathbf{r})\mathbf{W}(\mathbf{q})^T &= \mathbf{W}(\mathbf{q})^T\mathbf{Q}(\mathbf{r}) \end{aligned}$$

with $\mathbf{e}$ being the unity quaternion: $\mathbf{e} = (1\ 0\ 0\ 0)$.

The dot product of two quaternions is:

$$\mathbf{r} \cdot \mathbf{q} = r_0q_0 + r_xq_x + r_yq_y + r_zq_z.$$

The conjugate quaternion of $\mathbf{q}$, $\overline{\mathbf{q}}$ is defined by

$$\overline{\mathbf{q}} = q_0 - iq_x - jq_y - kq_z,$$

and obviously we have

$$\mathbf{q} * \overline{\mathbf{q}} = \mathbf{q} \cdot \mathbf{q} = \|\mathbf{q}\|^2.$$

An interesting property that is straightforward and will be used in the next section is

$$\|\mathbf{r} * \mathbf{q}\|^2 = \|\mathbf{r}\|^2\|\mathbf{q}\|^2. \tag{A1}$$

A 3-vector may well be viewed as a purely imaginary quaternion (its real part is equal to zero). One may notice that $\mathbf{W}(\mathbf{v})$ and $\mathbf{Q}(\mathbf{v})$ associated with a 3-vector $\mathbf{v}$ are skew-symmetric matrices.

Let $\mathbf{q}$ be a unit quaternion (that is, $\mathbf{q} \cdot \mathbf{q} = 1$), and let $\mathbf{v}$ be a purely imaginary quaternion. We have:

$$\begin{aligned} \mathbf{v}' &= \mathbf{q} * \mathbf{v} * \overline{\mathbf{q}} \\ &= (\mathbf{Q}(\mathbf{q})\mathbf{v}) * \overline{\mathbf{q}} \\ &= (\mathbf{W}(\mathbf{q})^T\mathbf{Q}(\mathbf{q}))\mathbf{v} \end{aligned} \tag{A2}$$

and one may easily figure out that

$$\mathbf{W}(\mathbf{q})^T\mathbf{Q}(\mathbf{q}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_xq_y - q_0q_z) & 2(q_xq_z + q_0q_y) \\ 0 & 2(q_xq_y + q_0q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_yq_z - q_0q_x) \\ 0 & 2(q_xq_z - q_0q_y) & 2(q_yq_z + q_0q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix}$$

is an orthogonal matrix. Hence, $\mathbf{v}'$ given by eq. (A2) is a 3-vector (a purely imaginary quaternion) and is the image of $\mathbf{v}$ by a rotation transformation $\mathbf{R}$:

$$\mathbf{R} = \begin{pmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix}$$

Appendix B: Derivation of Eq. (31)

The expression of $f_2(\mathbf{q}, \mathbf{t})$ (i.e., eq. (31)) can be easily derived using the properties of $\mathbf{W}(\mathbf{q})$ and $\mathbf{Q}(\mathbf{q})$ outlined in Appendix A:

$$\begin{aligned} &\|\mathbf{q} * \mathbf{p}_i - (\mathbf{K}_i - \mathbf{I})\mathbf{t} * \mathbf{q} - \mathbf{p}'_i * \mathbf{q}\|^2 \\ &= \left(\mathbf{W}(\mathbf{p}_i)\mathbf{q} - \mathbf{W}(\mathbf{q})(\mathbf{K}_i - \mathbf{I})\mathbf{t} - \mathbf{Q}(\mathbf{p}'_i)\mathbf{q}\right)^{\mathrm{T}} \\ &\quad \cdot \left(\mathbf{W}(\mathbf{p}_i)\mathbf{q} - \mathbf{W}(\mathbf{q})(\mathbf{K}_i - \mathbf{I})\mathbf{t} - \mathbf{Q}(\mathbf{p}'_i)\mathbf{q}\right) \\ &= \mathbf{q}^{\mathrm{T}}\mathcal{B}_i\mathbf{q} + \mathbf{t}^{\mathrm{T}}\mathcal{C}_i\mathbf{t} + \delta_i\mathbf{t} - 2\mathbf{q}^{\mathrm{T}}\mathbf{W}(\mathbf{p}_i)^{\mathrm{T}}\mathbf{W}(\mathbf{q})(\mathbf{K}_i - \mathbf{I})\mathbf{t}, \end{aligned}$$

where the expressions of $\mathcal{B}_i$, $\mathcal{C}_i$, and δ_i are:

$$\begin{aligned} \mathcal{B}_i &= (\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_i + \mathbf{p}_i'^{\mathrm{T}}\mathbf{p}'_i)\mathbf{I} - \mathbf{W}(\mathbf{p}_i)^{\mathrm{T}}\mathbf{Q}(\mathbf{p}'_i) - \mathbf{Q}(\mathbf{p}'_i)^{\mathrm{T}}\mathbf{W}(\mathbf{p}_i) \\ \mathcal{C}_i &= \mathbf{K}_i^{\mathrm{T}}\mathbf{K}_i - \mathbf{K}_i - \mathbf{K}_i^{\mathrm{T}} + \mathbf{I} \\ \delta_i &= 2\mathbf{p}_i'^{\mathrm{T}}(\mathbf{K}_i - \mathbf{I}). \end{aligned}$$

The last term may be transformed as follows:

$$\begin{aligned} -2\mathbf{q}^{\mathrm{T}}\mathbf{W}(\mathbf{p}_i)^{\mathrm{T}}\mathbf{W}(\mathbf{q})(\mathbf{K}_i - \mathbf{I})\mathbf{t} &= -2\mathbf{p}_i^{\mathrm{T}}\mathbf{Q}(\mathbf{q})^{\mathrm{T}}\mathbf{W}(\mathbf{q})(\mathbf{K}_i - \mathbf{I})\mathbf{t} \\ &= -2\mathbf{p}_i^{\mathrm{T}}\left(\mathbf{W}(\mathbf{q})^{\mathrm{T}}\mathbf{Q}(\mathbf{q})\right)^{\mathrm{T}}(\mathbf{K}_i - \mathbf{I})\mathbf{t}. \end{aligned}$$

The matrix $\mathbf{W}(\mathbf{q})^{\mathrm{T}}\mathbf{Q}(\mathbf{q})$ is the unknown rotation and is equal to either $\mathbf{R_X}$ or $\mathbf{R_Y}$. The matrix $\mathbf{K}_i$ is a rotation as well and is equal to either $\mathbf{R}_{\mathbf{A}_i}$ or $\mathbf{N}_i$. Notice that we have, from eqs. (15) and (22),

$$\begin{aligned} \mathbf{R_X}^{\mathrm{T}}\mathbf{R}_{\mathbf{A}_i} &= \mathbf{R}_{\mathbf{B}_i}\mathbf{R_X}^{\mathrm{T}} \\ \mathbf{R_Y}^{\mathrm{T}}\mathbf{N}_i &= \mathbf{R}_{\mathbf{B}_i}\mathbf{R_Y}^{\mathrm{T}}. \end{aligned}$$

Therefore, one may write:

$$\left(\mathbf{W}(\mathbf{q})^{\mathrm{T}}\mathbf{Q}(\mathbf{q})\right)^{\mathrm{T}}(\mathbf{K}_i - \mathbf{I}) = \left(\mathbf{R}_{\mathbf{B}_i} - \mathbf{I}\right)\left(\mathbf{W}(\mathbf{q})^{\mathrm{T}}\mathbf{Q}(\mathbf{q})\right)^{\mathrm{T}}.$$

Finally we obtain for the last term:

$$\begin{aligned} -2\mathbf{q}^{\mathrm{T}}\mathbf{W}(\mathbf{p}_i)^{\mathrm{T}}\mathbf{W}(\mathbf{q})(\mathbf{K}_i - \mathbf{I})\mathbf{t} &= -2\mathbf{p}_i^{\mathrm{T}}\left(\mathbf{W}(\mathbf{q})^{\mathrm{T}}\mathbf{Q}(\mathbf{q})\right)^{\mathrm{T}}(\mathbf{K}_i - \mathbf{I})\mathbf{t} \\ &= -2\mathbf{p}_i^{\mathrm{T}}(\mathbf{R}_{\mathbf{B}_i} - \mathbf{I})\left(\mathbf{W}(\mathbf{q})^{\mathrm{T}}\mathbf{Q}(\mathbf{q})\right)^{\mathrm{T}}\mathbf{t} \\ &= -2\mathbf{p}_i^{\mathrm{T}}(\mathbf{R}_{\mathbf{B}_i} - \mathbf{I})\mathbf{Q}(\mathbf{q})^{\mathrm{T}}\mathbf{W}(\mathbf{q})\mathbf{t} \end{aligned}$$

and

$$\varepsilon_i = -2\mathbf{p}_i^{\mathrm{T}}(\mathbf{R}_{\mathbf{B}_i} - \mathbf{I}).$$

Acknowledgments

This work has been supported by the Esprit programme through the SECOND project (Esprit-BRA No. 6769). The authors acknowledge Roger Mohr, Long Quan, Thai Quynh Phong, Bill Triggs, Sthéphane Lavallée, François Chaumette, Claude Inglebert, Christian Bard, Thomas Skordas, and Boguslaw Lorecki for fruitful discussions and for their many insightful comments. Many thanks go to Patrick Fulconis and Christian Bard for the many hours they have spent with the robot. François Chaumette kindly provided robot and camera calibration data.

References

Bard, C., Bellier, C., Troccaz, J., Laugier, C., Triggs, B., and Vercelli, G. In press. Achieving dextrous grasping by integrating planning and vision based sensing. *Int. J. Robot. Res.*

Boufama, B., Mohr, R., and Veillon, F. 1993 (May, Berlin). Euclidean constraints for uncalibrated reconstruction. In *Proceedings Fourth International Conference on Computer Vision.* Los Alamitos, CA: IEEE Computer Society Press, pp. 466–470.

Chen, H. 1991 (Hawaii, June). A screw motion approach to uniqueness analysis of head-eye geometry. In *Proceedings Computer Vision and Pattern Recognition*, pp. 145–151.

Chou, J. C. K., and Kamel, M. 1991. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *Int. J. Robot. Res.* 10(3):240–254.

Espiau, B., Chaumette, F., and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Trans. Robot. Automation* 8(3):313–326.

Faugeras, O. D., and Toscani, G. 1986 (June, Miami Beach). The calibration problem for stereo. In *Proc. Computer Vision and Pattern Recognition*, pp. 15–20.

Faugeras, O. D., and Hebert, M. 1986. The representation, recognition, and locating of 3D objects. *Int. J. Robot. Res.* 5(3):27–52.

Fletcher, R. 1990. *Practical Methods of Optimization.* New York: John Wiley & Sons.

Gill, P. E., Murray, W., and Wright, M. H. 1989. *Practical Optimization.* London: Academic Press.

Horaud, R., Mohr, R., and Lorecki, B. 1992 (May, Nice, France). Linear-camera calibration. In *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 1539–1544.

Horaud, R., Mohr, R., and Lorecki, B. 1993. On single-scanline camera calibration. *IEEE Trans. Robot. Automation* 9(1):71–75.

Horaud, R., and Monga, O. 1993. *Vision par Ordinateur: Outils Fondamentaux*. Paris: Editions Hermès.

Horn, B. K. P. 1987. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A.*, 4(4):629–642.

Phong, T. Q., Horaud, R., Yassine, A., and Pham, D. T. 1995. In press. Object pose from 2-D to 3-D point and line correspondences. Technical report RT 95. LIFIA-IMAG. *International Journal of Computer Vision.*

Phong, T. Q., Horaud, R., Yassine, A., and Pham, D. T. 1993 (May, Berlin). Optimal estimation of object pose from a single perspective view. In *Proceedings Fourth International Conference on Computer Vision*, Los Alamitos, CA: IEEE Computer Society Press, pp. 534–539.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Wetterling, W. T. 1988. *Numerical Recipes in C: The Art of Scientific Computing*. New York: Cambridge University Press.

Sabata, B., and Aggarwal, J. K. 1991. Estimation of motion from a pair of images: A review. *CGVIP-Image Understanding* 54(3):309–324.

Shiu, Y. C., and Ahmad, S. 1989. Calibration of wrist mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$. *IEEE J. Robot. Automation* 5(1):16–29.

Tsai, R. Y. 1987. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robot. Automation* RA-3(4):323–344.

Tsai, R. Y., and Lenz, R. K. 1989. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE J. Robot. Automation* 5(3):345–358.

Wang, C.-C. 1992. Extrinsic calibration of a robot sensor mounted on a robot. *IEEE Trans. Robot. Automation* 8(2):161–175.

Yassine, A. 1989. Etudes Adaptives et Comparatives de Certains Algorithmes en Optimisation. Implémentation Effectives et Applications. Ph.D. thesis, Université Joseph Fourier, Grenoble, France.