The Online Solution of the Hand–Eye Problem

Jorge Angeles, *Senior Member, IEEE*, Gilbert Soucy, and Frank P. Ferrie, *Member, IEEE*

***Abstract*—The *hand–eye problem* consists in determining the relative pose between two coordinate frames fixed to the same rigid body from measurements of the poses attained by these two frames, as the body moves. In robotics this problem arises when two frames are attached to the end-effector (EE), one of these at the gripper, the other to a sensor such as a camera or a laser range-finder. Various procedures have been proposed to solve this problem when perfect pose measurements are available at a pair of EE poses, the treatment of noisy measurements being a current research topic. Solutions proposed for the case of perfect measurements require an iterative procedure based on the singular-value decomposition, which itself relies on iterative procedures. The treatment of noisy measurements has led to offline least-square solutions. It is shown in this paper that, based on an invariant formulation of the problem at hand, a solution is possible that relies on recursive linear least squares. Thus, the procedure lends itself to an online implementation, as demonstrated here with experimental results. A major difference between the proposed procedure and those reported in the literature is that the latter are *iterative*; ours is *recursive*.**

***Index Terms*—Calibration, hand–eye, online, range-finder, recursive least squares.**

I. Introduction

THE *HAND–EYE problem* owes its origin to the photogrammetric problem known as *absolute orientation* [1]; in robotics, this problem leads to the determination of the *hand–eye transform* [2]. The photogrammetric problem mentioned above consists in determining the proper orthogonal transformation defining a rigid-body rotation that relates the coordinates of two sets of landmark points of the same body in two distinct poses. The least-square solution of the foregoing problem involving three or more landmark points, in closed form, was reported by Horn [3] using quaternions. A germane problem, known in the robotics literature as *hand–eye calibration*, appears to have been first formulated by Shiu and Ahmad [4] in 1987. An expanded version of this paper appeared in 1989 [5]. Briefly stated, this problem consists as follows.

> Given two frames, $\mathcal{G}$ and $\mathcal{H}$, attached to the gripper of a robotic end-effector (EE) and to a sensor rigidly mounted on the EE, respectively, find the relative *pose*—position and orientation—of the two frames, from measurements of the EE pose by means of sensor and joint-encoder readouts.

The problem at hand is particularly relevant in the field of reverse engineering, whereby uncertainties in camera or range-finder-mounting lead to meshing errors when two or more views are merged. Algebraically, the problem is known to lead to a linear homogeneous equation in the unknown pose matrix $\mathbf{X}$, namely

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

where $\mathbf{A}$, $\mathbf{B}$, and X are all homogeneous 4×4 transformation matrices. However, (1) alone is not sufficient to solve for $\mathbf{X}$, as Shiu and Ahmad pointed out in the foregoing references. Hence, at least two measurements are needed, which thus would yield two such equations, one for each measurement. Most of the work reported in the literature has been directed toward the development of fast algorithms for the solution of the foregoing equation, when two pairs of pose measurements $\{\mathbf{A}_i, \mathbf{B}_i\}_1^2$ are available, which is known to require an *iterative solution*. Indeed, because of the nature of matrix $\mathbf{X}$, it is nonsingular, and hence, (1) can be rewritten in the form

$$\mathbf{X}^{-1}\mathbf{AX} = \mathbf{B}$$

which is, apparently, a nonlinear equation in $\mathbf{X}$. In fact, because of the special form of 4×4 homogeneous transformation matrices [6], the above equation turns out to be quadratic in $\mathbf{X}$, the problem thus reducing to solving a system of quadratic scalar equations. The problem becomes more cumbersome, however, because the rotation block of matrix $\mathbf{X}$ must obey six additional quadratic constraints that guarantee that the said block is proper orthogonal.

The solution first proposed by Shiu and Ahmad exploited the properties of proper orthogonal matrices, while decoupling the orientation problem from the position problem. This solution, relying on two perfect measurements, led to a system of nine linear equations in four unknowns [5].

Chou and Kamel [7] streamlined Shiu and Ahmad's procedure by means of quaternions, but their approach required the introduction of an iterative procedure relying on the singular-value decomposition of a matrix at each iteration. If we consider that the singular-value decomposition is itself iterative, Chou and Kamal's procedure did not lead to a computational gain. Moreover, their procedure neither considered measurement noise. Horaud and Dornaika [8] addressed the issue of noisy measurements for the first time. Moreover, these researchers did away with the explicit representation of the intrinsic and extrinsic parameters of the camera and, upon resorting to a quaternion formulation proposed by Faugeras and Hebert [9], along with the decoupling of the rotation from

Manuscript received June 15, 1999; revised April 20, 2000. This paper was recommended for publication by Associate Editor Y. Xu and Editor V. Lumelsky upon evaluation of the reviewers comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant A4532 and Grant OGPIN 011. This paper was presented in part at the Seveth International Workshop on Computer-Aided Systems Theory and Technology (EUROCAST), Vienna, Austria, September 29–October 2, 1999.

J. Angeles is with the Department of Mechanical Engineering, Centre for Intelligent Machines, McGill University, Montreal, PQ H3A 2K6, Canada (e-mail: angeles@cim.mcgill.ca).

G. Soucy and F. P. Ferrie are with the Department of Electrical Engineering, Centre for Intelligent Machines, Montreal, PQ H3A 2A7, Canada (e-mail: soucy@cim.mcgill.ca; ferrie@cim.mcgill.ca).

Publisher Item Identifier S 1042-296X(00)11583-6.

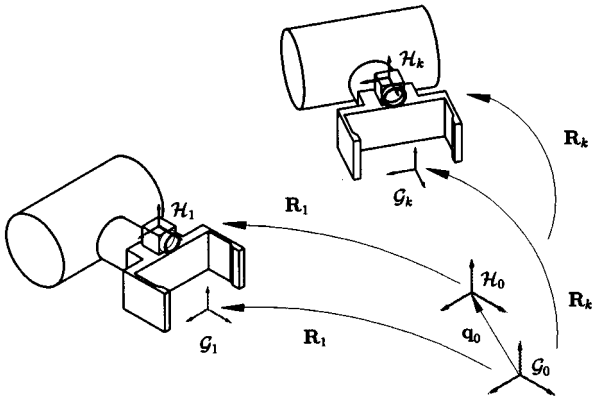


Fig. 1. Layout of the hand–eye calibration problem.

the translation, they ended up with a solution based on the calculation of the minimum eigenvalue of a 4×4 matrix. In a second approach proposed in the same paper, the authors deal with rotation and translation simultaneously, which then leads to a more complicated optimization problem. Indeed, this approach leads to an objective function that is quadratic in a nonlinear function of the decision variables, for which explicit solutions are not apparent.

In summary, what Faugeras and Hebert achieved was a formulation of the hand–eye problem that filters measurement errors by means of least-square methods. Later, Zhuang [10] pointed out that the new formulation proposed by Horaud and Dornaika is especially suited to cases where a monocular camera or a single range-finder is used; when a stereo camera, or a monocular camera supported by a laser range-finder, is used, then the original formulation, as first proposed by Shiu and Ahmad, is still valid.

In this paper we propose an algorithm to solve the hand–eye problem that relies on recursive linear least squares and is, hence, applicable online. To this end, we resort to the invariant formulation first proposed in [11] and streamlined in [6].

An *invariant formulation* is understood here in the sense of field theory [12]. That is, a formulation is *frame-invariant*, or *invariant* for brevity, if it is based on quantities that remain invariant under a change of *observer*. An observer, in turn, is understood in this context as a coordinate frame with a clock. A discussion on invariance in the realm of robotics is included in [6]. Note that invariance does not imply at all that the estimation procedure described herein produces the same results regardless of where the coordinate frames are attached. This could not possibly be the case, for the condition number of the underlying Jacobian matrix depends on the location of the origin of the frame attached to the gripper, and the accuracy of the direct kinematics, on which we rely for the computation of the pose of the gripper-frame, is dependent upon that condition number.

II. Problem Formulation

At the outset we assume the layout of Fig. 1, which shows frames $\mathcal{G}$ and $\mathcal{H}$ rigidly attached, respectively, to the gripper and to the sensor of a robotic EE.

Moreover, let $\mathbf{R}$ denote the rotation defining the orientation of each of the two frames, $\mathcal{G}$ and $\mathcal{H}$, with respect to their respective reference pose, given by frames $\mathcal{G}_0$ and $\mathcal{H}_0$. Note that, although the two frames are different, both undergo the same rotation $\mathbf{R}$ from their reference poses, which is a consequence of the two frames being attached to the same rigid body, the EE.

We now assume that pose measurements provided by the sensor and the joint encoders produce the 3×3 orthogonal matrices $\mathbf{G}$ and $\mathbf{H}$, representing $\mathbf{R}$ in $\mathcal{G}$ and $\mathcal{H}$, respectively; besides, the translations $\mathbf{s}$ and $\mathbf{t}$ undergone by the origins of $\mathcal{G}$ and $\mathcal{H}$, respectively, from their reference poses, are represented in $\mathcal{G}$ by $\mathbf{u}$ and in $\mathcal{H}$ by $\mathbf{v}$. With pose measurements of these two frames at various poses of the EE, we want to determine the relative orientation $\mathbf{Q}$ and the relative translation $\mathbf{q}$ of frame $\mathcal{H}$ with respect to frame $\mathcal{G}$.

Since we have two coordinate frames attaining various poses, the vectors and matrices involved in the calculations below can be represented in many frames. In order to avoid confusion, we will display the represented variable, whether a vector or a matrix, within subscripted brackets, the subscript indicating the representation frame. We thus have $\mathbf{G} \equiv [\mathbf{R}]_{\mathcal{G}}$ and $\mathbf{H} \equiv [\mathbf{R}]_{\mathcal{H}}$, and hence, the algebraic problem at hand consists in determining $[\mathbf{Q}]_{\mathcal{G}}$ and $[\mathbf{q}]_{\mathcal{G}}$ or, equivalently, $[\mathbf{Q}]_{\mathcal{H}}$ and $[\mathbf{q}]_{\mathcal{H}}$. Note that $\mathbf{G}$ can be obtained from a similarity transformation, which leads to

$$\mathbf{G} = [\mathbf{Q}]_{\mathcal{G}} \mathbf{H} [\mathbf{Q}^T]_{\mathcal{G}}$$

and hence

$$\mathbf{G}[\mathbf{Q}]_{\mathcal{G}} = [\mathbf{Q}]_{\mathcal{G}} \mathbf{H}. \tag{2}$$

Likewise, $\mathbf{u}$ and $\mathbf{v}$ are related with $[\mathbf{q}]_{\mathcal{G}}$, $\mathbf{G}$ and $\mathbf{H}$ via [11]

$$\mathbf{G}[\mathbf{q}]_{\mathcal{G}} + \mathbf{u} = [\mathbf{Q}]_{\mathcal{G}} \mathbf{v} + [\mathbf{q}]_{\mathcal{G}}$$

i.e.,

$$(\mathbf{G} - \mathbf{1})[\mathbf{q}]_{\mathcal{G}} = [\mathbf{Q}]_{\mathcal{G}} \mathbf{v} - \mathbf{u}. \tag{3}$$

III. RLS Solution of the Hand–Eye Problem

We refer to the notation and the layout of Fig. 1. From joint-encoder readouts and a robot forward-kinematics model [6], we can determine the pose of frame $\mathcal{G}$; likewise, from camera or range-finder measurements we can determine the pose of $\mathcal{H}$. What we usually do not know is the relative pose of $\mathcal{H}$ with respect to $\mathcal{G}$. This pose is given by the position vector $\mathbf{q}$ of the origin O_H of $\mathcal{H}$ from the origin O_G of $\mathcal{G}$ and the rotation matrix $\mathbf{Q}$ that represents the orientation of $\mathcal{H}$ with respect to $\mathcal{G}$. When these items, $\mathbf{q}$ and $\mathbf{Q}$, are represented either in frame $\mathcal{G}$ or in frame $\mathcal{H}$, the representations $[\mathbf{q}]_{\mathcal{G}}$ and $[\mathbf{Q}]_{\mathcal{G}}$ or their counterparts in frame $\mathcal{H}$, $[\mathbf{q}]_{\mathcal{H}}$, and $[\mathbf{Q}]_{\mathcal{H}}$, respectively, are constant. The purpose of the problem under study is to determine these representations in any of the two given frames. For conciseness, we aim at these representations in $\mathcal{G}$. Furthermore, we assume that a sequence of N measurements of $\mathcal{G}$ and $\mathcal{H}$ poses $\{\mathbf{G}_k, \mathbf{u}_k, \mathbf{H}_k, \mathbf{v}_k\}_1^N$, from reference poses $\mathcal{G}_0$ and $\mathcal{H}_0$, at which we can assume that $\mathbf{G}_0 = \mathbf{H}_0 = \mathbf{1}$, and $\mathbf{u}_0 = \mathbf{v}_0 = \mathbf{0}$, are available. In the sequel, we shall need the *axial vectors* [6] of the foregoing matrices. For quick reference, if $\mathbf{M}$ denotes an arbitrary 3×3 matrix, its axial vector, $\mathbf{m}$, represented by $\mathrm{vect}(\mathbf{M})$, is defined as one-half the ordered array of the differences of the

off-diagonal entries of $\mathbf{M}$. Thus, if $m_{i,j}$ denotes the (i, j) entry of $\mathbf{M}$, then

$$\mathbf{m} \equiv \text{vect}(\mathbf{M}) = \frac{1}{2}\begin{bmatrix} m_{32} - m_{23} \\ m_{13} - m_{31} \\ m_{21} - m_{12} \end{bmatrix}. \tag{4}$$

Now, since $\mathbf{G}_k$ and $\mathbf{H}_k$ represent the same rotation of the EE from a reference pose, albeit in different frames, $\mathcal{G}$ and $\mathcal{H}$, respectively, the vectors of these matrices, $\mathbf{g}_k$ and $\mathbf{h}_k$, respectively, obey the relation below [6]

$$\mathbf{g}_k = [\mathbf{Q}]_{\mathcal{G}}\mathbf{h}_k, \qquad k = 1, 2, \ldots, N. \tag{5}$$

We now write the above relation in the form

$$\mathbf{g}_k^T\mathbf{Q} = \mathbf{h}_k^T, \qquad k = 1, 2, \ldots, N \tag{6}$$

where we have dispensed with the subscripted brackets, for notation simplicity. Upon assembling all N vector equations (6), and after simple manipulations, we obtain

$$\mathbf{A}_N\mathbf{Q} = \mathbf{B}_N \tag{7}$$

in which the $N \times 3$ matrices $\mathbf{A}_N$ and $\mathbf{B}_N$ are given as

$$\mathbf{A}_N \equiv \begin{bmatrix} \mathbf{g}_1^T \\ \mathbf{g}_2^T \\ \vdots \\ \mathbf{g}_N^T \end{bmatrix} \qquad \mathbf{B}_N \equiv \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \vdots \\ \mathbf{h}_N^T \end{bmatrix}. \tag{8}$$

Thus, (7) represents, for $N > 3$, an overdetermined system of N three-dimensional (3-D) vector equations for the nine unknown components of $\mathbf{Q}$. The least-square approximation $\hat{\mathbf{Q}}_N$ of this system has the form of $\hat{\mathbf{x}}_N$, as given in (29) of the Appendix, namely

$$\hat{\mathbf{Q}}_N = \left(\mathbf{A}_N^T\mathbf{A}_N\right)^{-1}\mathbf{A}_N^T\mathbf{B}_N. \tag{9}$$

If we take one more measurement, $N + 1$, we obtain

$$\mathbf{g}_{N+1}^T\mathbf{Q} = \mathbf{h}_{N+1}^T \tag{10}$$

which leads correspondingly to the new least-square estimate $\hat{\mathbf{Q}}_{N+1}$ given by

$$\hat{\mathbf{Q}}_{N+1} = \left(\mathbf{A}_{N+1}^T\mathbf{A}_{N+1}\right)^{-1}\mathbf{A}_{N+1}^T\mathbf{B}_{N+1} \tag{11}$$

where

$$\mathbf{A}_{N+1} \equiv \begin{bmatrix} \mathbf{A}_N \\ \mathbf{g}_{N+1}^T \end{bmatrix} \qquad \mathbf{B}_{N+1} \equiv \begin{bmatrix} \mathbf{B}_N \\ \mathbf{h}_{N+1}^T \end{bmatrix}. \tag{12}$$

Henceforth, we refer to the recursive least-square (RLS) algorithm of the Appendix. Similar to (39), we now have

$$\mathbf{P}_N \equiv \left(\mathbf{A}_N^T\mathbf{A}_N\right)^{-1} \tag{13}$$

and hence, upon mimicking (43), we obtain

$$\mathbf{P}_{N+1} = \mathbf{P}_N - \frac{1}{1 + \mathbf{g}_{N+1}^T\mathbf{P}_N\mathbf{g}_{N+1}}\mathbf{P}_N\mathbf{g}_{N+1}\mathbf{g}_{N+1}^T\mathbf{P}_N. \tag{14}$$

Moreover, the counterpart of (46) is

$$\hat{\mathbf{Q}}_{N+1} = \hat{\mathbf{Q}}_N + \mathbf{k}_N\left(\mathbf{h}_{N+1}^T - \mathbf{g}_{N+1}^T\hat{\mathbf{Q}}_N\right) \tag{15}$$

where $\mathbf{k}_N$ is given by

$$\mathbf{k}_N = \frac{1}{1 + \mathbf{g}_{N+1}^T\mathbf{P}_N\mathbf{g}_{N+1}}\mathbf{P}_N\mathbf{g}_{N+1}. \tag{16}$$

Now, in order to set up a scheme to estimate vector $\mathbf{q}$, we need to resort to an estimate of $\mathbf{Q}$. It is apparent from (9) that $\hat{\mathbf{Q}}$ can be computed only if $N \geq 3$; otherwise, $\mathbf{A}_N$ is of rank smaller than three, and hence, $\mathbf{A}_N^T\mathbf{A}_N$ is singular and cannot be inverted. On the other hand, to obtain the first estimate of $\mathbf{q}$ we need a system of three independent translation equations (3), which means that we need at least $N = 5$ to start the estimations of $\mathbf{q}$. Thus, under the assumption that $k \geq 5$, we can use the current estimate of $\mathbf{Q}$, $\hat{\mathbf{Q}}_k$, in (3) at the kth translation measurement, to estimate the translation $\mathbf{q}$. The N pose measurements yield, then, the $N - 2$ translation relations

$$(\mathbf{G}_k - \mathbf{1})\mathbf{q} = \hat{\mathbf{Q}}_k\mathbf{v}_k - \mathbf{u}_k, \qquad k = 3, \ldots, N. \tag{17}$$

Similar to the scheme that we derived to update estimates of $\mathbf{Q}$, we would like to use the above relations to update the estimates of $\mathbf{q}$. However, we have a major difference between (6) and (17): the two sets are 3-D vector equations, the former having a matrix unknown, $\mathbf{Q}$, and a row vector coefficient at each measurement. The latter has a vector unknown and a matrix coefficient at each measurement. This means that, while we add just one row to the estimation matrix $\mathbf{A}_N$, we add a 3×3 block to the estimation matrix underlying (17) every time a new measurement is taken. The outcome is that the recursive scheme of the Appendix would not be applicable with the ease with which we have applied it to the estimate of $\mathbf{Q}$. We thus take an alternative approach to estimating $\mathbf{q}$. To this end, we simply dot-multiply the two sides of (17) by $\mathbf{u}_k$ or, equivalently, we multiply this equation by $\mathbf{u}_k^T$ from the left, thereby obtaining

$$\mathbf{u}_k^T(\mathbf{G}_k - \mathbf{1})\mathbf{q} = \mathbf{u}_k^T\hat{\mathbf{Q}}_k\mathbf{v}_k - \mathbf{u}_k^T\mathbf{u}_k, \qquad k = 3, \ldots, N.$$

Further, in order to avoid large numbers that would hamper the accuracy of the procedure, and with the purpose of rendering the foregoing scheme dimensionless, as is its orientation counterpart, we divide the two sides of the above equation by $\mathbf{u}_k^T\mathbf{u}_k = \|\mathbf{u}_k\|^2$, which thus leads, for $k = 3, \ldots, N$ to

$$\frac{1}{\|\mathbf{u}_k\|^2}\mathbf{u}_k^T(\mathbf{G}_k - \mathbf{1})\mathbf{q} = \frac{1}{\|\mathbf{u}_k\|^2}\mathbf{u}_k^T\hat{\mathbf{Q}}_k\mathbf{v}_k - 1. \tag{18a}$$

We now have a scheme that is suitable for a RLS implementation à la the Appendix. To this end, we assemble the above $N - 2$ scalar equations in the form

$$\mathbf{C}_N\mathbf{q} = \mathbf{d}_N \tag{18b}$$

with $\mathbf{C}_N$ and $\mathbf{d}_N$ defined as

$$\mathbf{C}_N \equiv \begin{bmatrix} \left[\left(\mathbf{G}_3^T - \mathbf{1}\right)\mathbf{u}_3\right]^T / \|\mathbf{u}_3\|^2 \\ \left[\left(\mathbf{G}_4^T - \mathbf{1}\right)\mathbf{u}_4\right]^T / \|\mathbf{u}_4\|^2 \\ \vdots \\ \left[\left(\mathbf{G}_N^T - \mathbf{1}\right)\mathbf{u}_N\right]^T / \|\mathbf{u}_N\|^2 \end{bmatrix} \tag{19a}$$

$$\mathbf{d}_N \equiv \begin{bmatrix} (1/\|\mathbf{u}_3\|^2)\mathbf{u}_3^T \hat{\mathbf{Q}}_3 \mathbf{v}_3 - 1 \\ (1/\|\mathbf{u}_4\|^2)\mathbf{u}_4^T \hat{\mathbf{Q}}_4 \mathbf{v}_4 - 1 \\ \vdots \\ (1/\|\mathbf{u}_N\|^2)\mathbf{u}_N^T \hat{\mathbf{Q}}_N \mathbf{v}_N - 1 \end{bmatrix}. \tag{19b}$$

The Nth estimate of $\mathbf{q}$, $\hat{\mathbf{q}}_N$, is thus calculated as the least-square approximation of (18b), namely, as

$$\hat{\mathbf{q}}_N = \left(\mathbf{C}_N^T \mathbf{C}_N\right)^{-1} \mathbf{C}_N^T \mathbf{d}_N, \qquad N \geq 5. \tag{20}$$

Therefore, $\hat{\mathbf{q}}_N$ is updated by following the standard procedure of the Appendix. Let

$$\boldsymbol{\delta}_N \equiv (\mathbf{G}_N^T - \mathbf{1})\mathbf{u}_N. \tag{21a}$$

Then

$$\mathbf{R}_N \equiv (\mathbf{C}_N^T \mathbf{C}_N)^{-1} \tag{21b}$$

$$\mathbf{l}_N = \frac{\mathbf{R}_N \boldsymbol{\delta}_N / \|\mathbf{u}_N\|^2}{1 + \boldsymbol{\delta}_N^T \mathbf{R}_N \boldsymbol{\delta}_N / \|\mathbf{u}_N\|^4} \tag{21c}$$

$$d_{N+1} = \frac{1}{\|\mathbf{u}_N\|^2}\, \mathbf{u}_N^T \hat{\mathbf{Q}}_N \mathbf{v}_N - 1 \tag{21d}$$

$$\hat{\mathbf{q}}_{N+1} = \hat{\mathbf{q}}_N - \frac{\mathbf{l}_N}{\|\mathbf{u}_{N+1}\|^2} \boldsymbol{\delta}_N^T \hat{\mathbf{q}}_N + d_{N+1} \mathbf{l}_N \tag{21e}$$

$$\mathbf{R}_{N+1} = \mathbf{R}_N - \frac{\mathbf{R}_N \boldsymbol{\delta}_{N+1} \boldsymbol{\delta}_{N+1}^T \mathbf{R}_N^T}{\|\mathbf{u}_{N+1}\|^4 + \boldsymbol{\delta}_{N+1}^T \mathbf{R}_N \boldsymbol{\delta}_{N+1} \left[(\mathbf{G}_{N+1}^T - \mathbf{1})\mathbf{u}_{N+1}\right]}. \tag{21f}$$

The estimation of $\hat{\mathbf{q}}_N$ begins for $N = 5$, which requires a 3×3 matrix $\mathbf{R}_5$, given by

$$\mathbf{R}_5 = (\mathbf{C}_5^T \mathbf{C}_5)^{-1} = \mathbf{C}_5^{-1} \mathbf{C}_5^{-T}$$

where the $-T$ superscript denotes the transpose of the inverse or, equivalently, the inverse of the transpose. Moreover

$$\mathbf{C}_5 = \begin{bmatrix} \left[(\mathbf{G}_3^T - \mathbf{1})\mathbf{u}_3\right]^T / \|\mathbf{u}_3\|^2 \\ \left[(\mathbf{G}_4^T - \mathbf{1})\mathbf{u}_4\right]^T / \|\mathbf{u}_4\|^2 \\ \left[(\mathbf{G}_5^T - \mathbf{1})\mathbf{u}_5\right]^T / \|\mathbf{u}_5\|^2 \end{bmatrix}.$$

Likewise, we have, for $N = 5$

$$\mathbf{d}_5 = \begin{bmatrix} (1/\|\mathbf{u}_3\|^2)\mathbf{u}_3^T \hat{\mathbf{Q}}_3 \mathbf{v}_3 - 1 \\ (1/\|\mathbf{u}_4\|^2)\mathbf{u}_4^T \hat{\mathbf{Q}}_4 \mathbf{v}_4 - 1 \\ (1/\|\mathbf{u}_5\|^2)\mathbf{u}_5^T \hat{\mathbf{Q}}_5 \mathbf{v}_5 - 1 \end{bmatrix}.$$

Note that the increment $\Delta\hat{\mathbf{Q}}_N$ in $\hat{\mathbf{Q}}$ from the Nth measurement to the $(N+1)$st is readily derived from (15) as

$$\Delta\hat{\mathbf{Q}}_N = \mathbf{k}_N (\mathbf{h}_{N+1}^T - \mathbf{g}_{N+1}^T \hat{\mathbf{Q}}_N)$$

the Euclidean norm-squared of $\Delta\hat{\mathbf{Q}}_N$ being computed as

$$\|\Delta\hat{\mathbf{Q}}_N\|_2^2 = \tfrac{1}{3}\,\mathrm{tr}(\Delta\hat{\mathbf{Q}}_N \Delta\hat{\mathbf{Q}}_N^T)$$

or, after expansion

$$\|\Delta\hat{\mathbf{Q}}_N\|_2^2 = \tfrac{1}{3}\|\mathbf{k}_N\|_2^2 \|\mathbf{h}_{N+1} - \hat{\mathbf{Q}}^T \mathbf{g}_{N+1}\|_2^2. \tag{22}$$

Upon convergence, $\|\Delta\hat{\mathbf{Q}}_N\|_2^2$ will tend to zero. Therefore, a suitable convergence criterion is

$$\|\Delta\hat{\mathbf{Q}}_N\| \leq \epsilon_R \tag{23}$$

where ϵ_R is a user-specified tolerance. Moreover, the above equation need not use the Euclidean norm, which is not the cheapest one to compute. The maximum norm can be used here to avoid slowing down the procedure unnecessarily. A convergence criterion for the estimate of $\mathbf{q}$ can be obtained from (21e) upon noticing that

$$\Delta\hat{\mathbf{q}}_N = \left[d_{N+1} - \frac{\mathbf{u}_{N+1}^T (\mathbf{G}_{N+1} - \mathbf{1})\hat{\mathbf{q}}_N}{\|\mathbf{u}_{N+1}\|^2}\right] \mathbf{l}_N$$

and hence, the convergence criterion for the estimation of $\hat{\mathbf{q}}_N$ is

$$\left\| \left[d_{N+1} - \frac{\mathbf{u}_{N+1}^T (\mathbf{G}_{N+1} - \mathbf{1})\hat{\mathbf{q}}_N}{\|\mathbf{u}_{N+1}\|^2}\right] \mathbf{l}_N \right\| \leq \epsilon_T \tag{24}$$

where, again, ϵ_T is a prescribed tolerance. We have thus derived the recursive algorithm below for the online solution of the hand–eye calibration problem.

Hand–Eye Algorithm

1. $\mathtt{N} \leftarrow 3$; input ϵ_R and ϵ_T
2. Take three initial measurements $\{(\mathbf{G}_\mathtt{i}, \mathbf{u}_\mathtt{i}, \mathbf{H}_\mathtt{i}, \mathbf{v}_\mathtt{i})\}_1^3$
3. Calculate the axial vectors of $\mathbf{G}_\mathtt{i}$ and $\mathbf{H}_\mathtt{i}$, $\{(\mathbf{g}_\mathtt{i}, \mathbf{h}_\mathtt{i})\}_1^3$
4. Build matrices $\mathbf{A}_\mathtt{N}$ and $\mathbf{B}_\mathtt{N}$; compute the first row of $\mathbf{C}_5$
5. $\mathbf{A}_\mathtt{N} \leftarrow \mathbf{A}_\mathtt{N}^{-1}$% $\mathbf{A}_3$ is square
6. $\mathbf{P} \leftarrow \mathbf{A}_\mathtt{N} \mathbf{A}_\mathtt{N}^\mathtt{T}$
7. $\hat{\mathbf{Q}} \leftarrow (\mathbf{A}_\mathtt{N}^{-1})\mathbf{B}_\mathtt{N}$
8. Take one more measurement $\mathbf{G}_{\mathtt{N}+1}$, $\mathbf{u}_{\mathtt{N}+1}$, $\mathbf{H}_{\mathtt{N}+1}$, $\mathbf{v}_{\mathtt{N}+1}$ rotation update begins
9. $\mathbf{g} \leftarrow \mathtt{vect}(\mathbf{G}_{\mathtt{N}+1})$; $\mathbf{h} \leftarrow \mathtt{vect}(\mathbf{H}_{\mathtt{N}+1})$; $\mathbf{G} \leftarrow \mathbf{G}_{\mathtt{N}+1}$
10. $\mathbf{p} \leftarrow \mathbf{P}\mathbf{g}$
11. $\mathtt{D} \leftarrow 1 + \mathbf{g}^\mathtt{T}\mathbf{p}$
12. $\mathbf{k} \leftarrow (1/\mathtt{D})\mathbf{p}$
13. $\mathbf{P} \leftarrow \mathbf{P} - (1/\mathtt{D})\mathbf{p}\mathbf{p}^T$
14. $\Delta\hat{\mathbf{Q}} = -\mathbf{k}\mathbf{g}^T\hat{\mathbf{Q}} + \mathbf{k}\mathbf{h}^T$
15. $\hat{\mathbf{Q}} \leftarrow \hat{\mathbf{Q}} + \Delta\hat{\mathbf{Q}}$% rotation update ends
16. If $\mathtt{N} \leq 5$ update $\mathbf{C}_\mathtt{N}$ and $\mathbf{d}_\mathtt{N}$% translation update begins
17. If $\mathtt{N} = 5$, then $\hat{\mathbf{q}} \leftarrow \mathbf{C}_\mathtt{N}^{-1}\mathbf{d}_\mathtt{N}$; $\mathbf{R} \leftarrow \mathbf{C}_\mathtt{N}^{-1}\mathbf{C}_\mathtt{N}^{-\mathtt{T}}$; go to 8
18. $\mathbf{u} \leftarrow \mathbf{u}_{\mathtt{N}+1}$; $\mathbf{v} \leftarrow \mathbf{v}_{\mathtt{N}+1}$
19. $\mathbf{r} \leftarrow \mathbf{R}[(\mathbf{G}^T - \mathbf{1})\mathbf{u}]$
20. $\mathtt{D} \leftarrow 1 + [(\mathbf{G}^T - \mathbf{1})\mathbf{u}]^\mathtt{T}\mathbf{r}/\|\mathbf{u}\|^4$
21. $\mathtt{l} \leftarrow (1/\mathtt{D})\mathbf{r}/\|\mathbf{u}\|^2$
22. $\mathtt{d} \leftarrow (\mathbf{u}^T\hat{\mathbf{Q}}\mathbf{v})/\|\mathbf{u}\|^2 - 1$
23. If $\|[d - \mathbf{u}^T(\mathbf{G} - \mathbf{1})\hat{\mathbf{q}}]\ (\mathtt{l}_N/\|\mathbf{u}_N\|^2)\| \leq \epsilon_T$ and $\|\Delta\hat{\mathbf{Q}}_N\| \leq \epsilon_R$ stop;

else continue % check translation & rotation criteria

24. $\hat{\mathbf{q}} \leftarrow \hat{\mathbf{q}} + [d - \hat{\mathbf{q}}^T\,(\mathbf{G}^T - \mathbf{1})\mathbf{u}/\|\mathbf{u}\|^2]\mathbf{l}$

25. $\mathbf{R} \leftarrow \mathbf{R} - (1/D)\mathbf{r}\mathbf{r}^T/\|\mathbf{u}\|^4$
26. go to 8 % end of translation update

Note that the square matrices $\mathbf{A}_3$ and $\mathbf{C}_5$ need not be orthogonal, and hence, provisions must be taken to deal with their condition numbers. In order to compute their inverses safely, at the beginning of the algorithm, it is recommended to do this using Gram–Schmidt orthogonalization.

IV. Post-Processing

The hand–eye algorithm, however, yields only an *estimate* $\hat{\mathbf{Q}}$ of $\mathbf{Q}$ that is not necessarily orthogonal. Nevertheless, under the assumption that the approximation at hand is close enough to the actual $\mathbf{Q}$, the nonorthogonal part of the above estimate can be readily obtained using the *polar-decomposition theorem* [13]. The error $\mathbf{E}$ of the approximation is then

$$\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T = \mathbf{1} + \mathbf{E} \tag{25}$$

with $\mathbf{1}$ denoting the 3×3 identity matrix. Now, let the polar decomposition of $\hat{\mathbf{Q}}$ be

$$\hat{\mathbf{Q}} = \tilde{\mathbf{Q}}\mathbf{U} \tag{26}$$

where $\tilde{\mathbf{Q}}$ is a *filtered* estimate of $\mathbf{Q}$ that is proper orthogonal, while $\mathbf{U}$ is symmetric and positive-definite. Moreover, if the difference between $\hat{\mathbf{Q}}$ and $\tilde{\mathbf{Q}}$ is small, then $\mathbf{U}$ must be of the form

$$\mathbf{U} = \mathbf{1} + \mathbf{F} \tag{27}$$

where $\mathbf{F}$ is a symmetric, although not-necessarily sign-definite matrix, of norm smaller than unity. The polar decomposition required above is implemented here with the aid of Higham's algorithm [14]. The estimated values $\hat{\mathbf{Q}}$ and $\hat{\mathbf{q}}$ will be henceforth referred to as the *hand–eye transform*.

V. Experiments

We demonstrate the foregoing algorithm on a setup composed of a laser range-finder attached to the EE of a gantry robot. As shown in Fig. 2, this type of setup can be used to acquire multiple 3-D views of any object placed in its workspace.

The laser range-finder provides precise 3-D measurements of the object expressed in its own intrinsic coordinate frame $\mathcal{H}$ in two views, as shown in Fig. 2, at frames $\mathcal{H}_1$ and $\mathcal{H}_2$. In order to merge the data sets 1 and 2 of Fig. 2, we need to know the displacement of the range-finder from view 1 to view 2, i.e., the relation between the two views at hand.

Obviously, from the robot joint encoders, we know the displacement incurred. In order to map this information into the range-finder frame, we need to know the hand–eye transform $\{\hat{\mathbf{Q}}, \hat{\mathbf{q}}\}$. Using this transform and the robot motion, we can map the two data sets into one unique coordinate frame forming a more complete 3-D model of the object scanned. Any mistake on the hand–eye transform computation will be reflected as an error in the merging of the data sets. The more precise the hand–eye transform, the more precise the merging of the data. The correspondence error of the merged range images can therefore be used as a measure of the precision of the hand–eye transform found, as we will see presently.

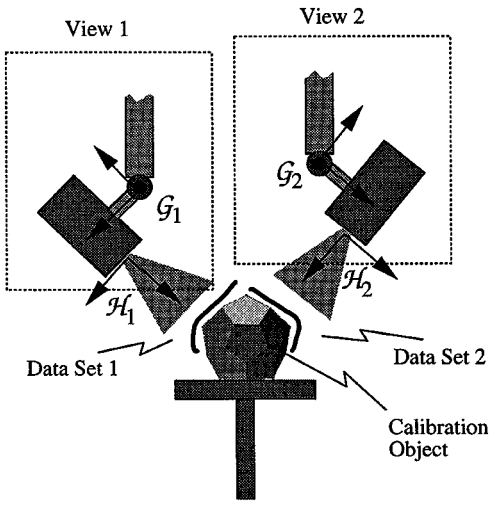


Fig. 2. Multiple 3-D view acquisition using a laser range-finder mounted on a robot.

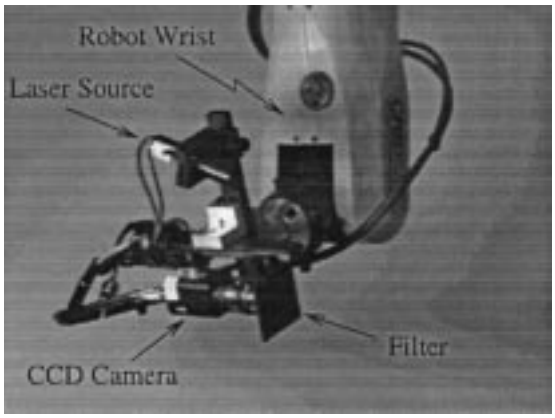


Fig. 3. Laser range-finder used for the experiments. A laser source projects a line of light permitting surface measurements by the triangulation principle.

Data-acquisition and pose-estimation are done on a sequence of stop-scan-move cycles, each taking about 10 sec. Most of this time is taken up by the data-acquisition, with pose-estimation being done in about 1% of the cycle time. The computer platform used for the tasks is a Pentium II 500-MHz running Red Hat Linux 6.1. It is to be noted that the underlying software has not been optimized for real-time applications, with all coding done in Perl, a scripting language.

We start this section by describing the range-finder and the robot used in this experiment. Then, we describe how we acquired the data to perform the hand–eye calibration of our setup and show the estimation histories through the steps. Finally, we verify the resulting hand–eye transform obtained by merging two range images of a test object.

A. Experimental Setup

1) The Range-Finder: Fig. 3 shows the laser range-finder used for the experiment. This is a custom-designed range-finder based on the triangulation principle. Its working range goes from 200 to 750 mm with a variable precision of about 0.3 mm

at best. It is noteworthy that the last joint of the gantry robot provides the motion required for the scanning process.

The result of the calibration of this scanner is a set of three look-up tables, one for each of the directions x, y, and z. These look-up tables are used to convert directly a pixel (i, j) to its corresponding Cartesian coordinates (x, y, z) in mm.

A special filter (model 510-670-S from Newport Inc., of Irvine, CA), tuned to the laser wavelength of 670 nm, is placed in front of the camera and blocks most of the ambient light, letting the CCD camera see only the laser line generated by the laser source. The laser source of 30 mW, a model SNF-501L-670S-30-45 from Lasiris Inc., of St-Laurent, Quebec, is focused at about 300 mm, the minimum width of the laser line being of about 100 μm.

The range-finder is assembled so that the laser line and the camera axis intersect at about 350 mm with a baseline of about 150 mm. In this configuration, when the range-finder is held fixed in front of an object, the CCD camera sees a profile of this object because of the laser line generated by the laser source and projected at an angle with respect to the camera axis. For close objects, the profile will appear at the top of the image. For far objects, the line appears at the bottom of the image. When a pixel of the CCD camera (see Fig. 3) is lit by the laser light reflected from a point in the scene, the look-up tables provide the relative position of that point with respect to an arbitrarily fixed coordinate frame attached to the range-finder.

One image from the CCD camera provides enough information to recover one calibrated profile of the object. In order to measure a surface, the whole CCD camera-laser source is rotated about a fixed axis by the last joint of the robot. During this motion, the CCD camera acquires more profiles of the object (usually 200), thus producing what is referred to as a "range image." The whole process takes about 10 s.

The calibration of the range-finder is divided into two steps, static and dynamic. The static part of the calibration is performed through the acquisition of many views, taken at different distances, of a flat grid made of horizontal and vertical black and white stripes. The distance between the black lines is 25 mm. This grid provides the x and y components of the look-up tables. Another set of images of the laser projected on a flat surface, taken with the filter on, provides the z component of the look-up tables. In the dynamic part of the calibration, the axis of rotation of the setup (last joint of the robot) is determined by taking many profiles of a plane at various angles. The axis of rotation of the setup is the one that minimizes the sum of the distances of the profiles to a plane. During this minimization, the profiles are remapped from their static intrinsic range-finder coordinates to their estimated real 3-D position by using the robot angle and the current estimate of the axis of rotation.

There exist many high-precision commercial laser range-finders, e.g., the HyScan 45C[1] and the W-Series by 3-D Scanners Ltd.[2] Although the intrinsic precision of these instruments is in the order of one μm, they still require a calibration technique to be able to merge multiple views of an object.

[1]Hymarc 3-D vision, Hymarc Ltd. Available at http://www.hymarc.com.

[2]3D scanners—W-series, 3-D Scanners Ltd. Available at http://www.3dscanners.com.

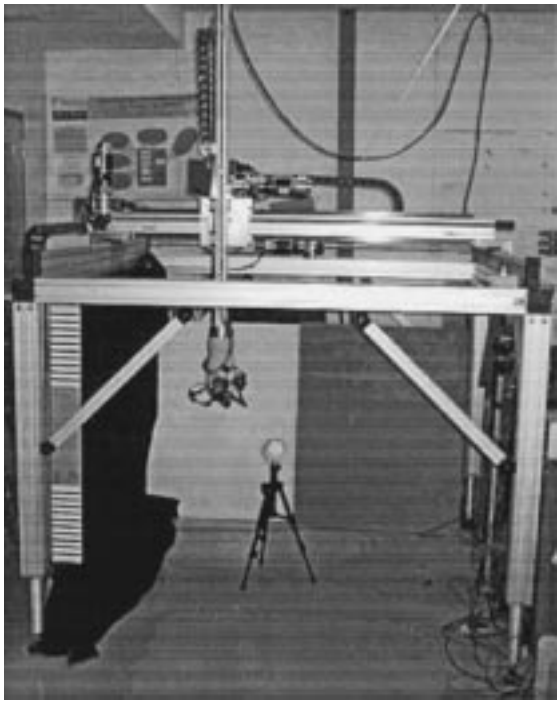


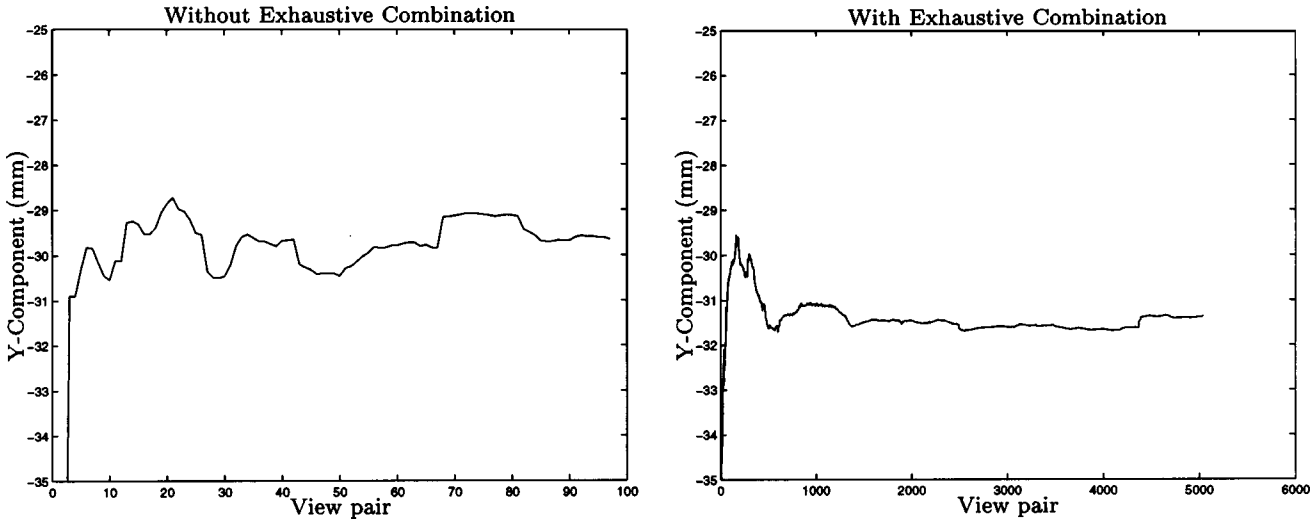
Fig. 4. Six-axis gantry robot used for hand–eye calibration.

2) The Manipulator: A CRS Plus G3000 gantry robot, shown in Fig. 4, is used for this experiment. This six-axis robot has a workspace of 1 m^3 and a repeatability of 0.3 mm. The global absolute precision is variable and depends on the particular physical installation. Prior to any calibration, the global precision was about $+/-10$ mm. The robot calibration was divided into two steps: *i*) calibration of the prismatic joints and *ii*) calibration of the wrist, as described below:

i) The compensation for the assembly errors in the translation axes (x, y, and z) was determined by measuring precisely the angles between the axes of the prismatic joints, i.e., of the robot-supporting beams, and the reference vertical and horizontal planes. The model takes all alignment imperfections into account while assuming perfectly straight beams.

In order to measure the absolute angles of the beams, a large rigid and flat platform is placed on the floor in the workspace of the robot. This platform is then adjusted to level in all directions with the help of a 700-mm long level. The precision in this adjustment is estimated to be around 2 mm over the length of the level, which means $\pm 0.16^\circ$. This platform provides the horizontal reference for the rest of the calibration.

The vertical reference is provided by one 750 mm $\times$ 100 mm panel suspended at each of the four corners of the robot, two of them perpendicular to the x axis and two of them perpendicular to the y axis. These panels are estimated to be vertical with a precision of about 1 mm over half the length of the panels (375 mm), which means $\pm 0.15^\circ$ of orientation accuracy. The absolute angles of the robot beams are obtained by moving the robot back and forth parallel to a specific reference plane and measuring the variation of the distance between the robot and



(a) (b)

Fig. 5. Comparison of the convergence behavior of the y component of $\hat{\mathbf{q}}$ by (a) combining only the two last physical measurements and (b) combining exhaustively all the physical measurements pairwise.

that plane. The laser range-finder attached to the end of the robot is used to measure this variation. For this purpose, the laser range-finder does not need to be calibrated: we simply match the raw readings from the range-finder at both ends of the reference plane by moving the robot closer or farther to the reference plane. The precision of this measurement depends on the raw accuracy of the range-finder, on the precision of the robot and on the backlash of the joints. This precision is estimated to be of about 0.5 mm over the usable length of the planes (680 mm), which means about $\pm 0.04^\circ$ of orientation accuracy. The overall precision on the absolute angles of the beams is then about $\pm 0.2^\circ$ $(0.04^\circ + 0.16^\circ)$. The biggest correction applied with the current calibration is 0.53°.

ii) The wrist was calibrated using a general three-revolute model [6]. A special calibration probe, attached to the EE, was used to point on a paper grid while recording the robot encoder values. A total of 215 points were collected in this manner. After a minimization procedure implemented in Matlab, the model parameters were found and led to an average error of 1.07 mm, the difference between the grid reading and the resulting forward kinematics model.

B. Hand–Eye Calibration

As described in Section II, we need a means to measure the position of the EE and of the laser range-finder in order to apply the calibration algorithm proposed in this paper. In general, it is not possible to measure the pose of the laser range-finder in the robot frame.[3] However, we can measure the relative motion of the range-finder with respect to some other coordinate frame whose location is not known with respect to the robot coordinates. Fig. 2 shows how we acquire the data necessary for the hand–eye calibration of our set up.

The EE poses $\mathcal{G}_i$ can be measured with respect to the robot base $\mathcal{B}$ through the forward kinematics model. The laser range-finder poses $\mathcal{H}_i$ can be measured, in turn, with respect to some stationary calibration object whose geometry is completely known. Many geometric objects can be used for this purpose. Here, we use a regular dodecahedron because it is a Platonic solid, whose vertices provide an optimally conditioned set of data points for pose estimation [15]. Besides, each view of this object contains at least three faces, which are necessary to compute a complete coordinate transform with respect to the laser range-finder. We obtain this transform through a standard range-image segmentation of the flat faces and an implicit knowledge of its geometry.

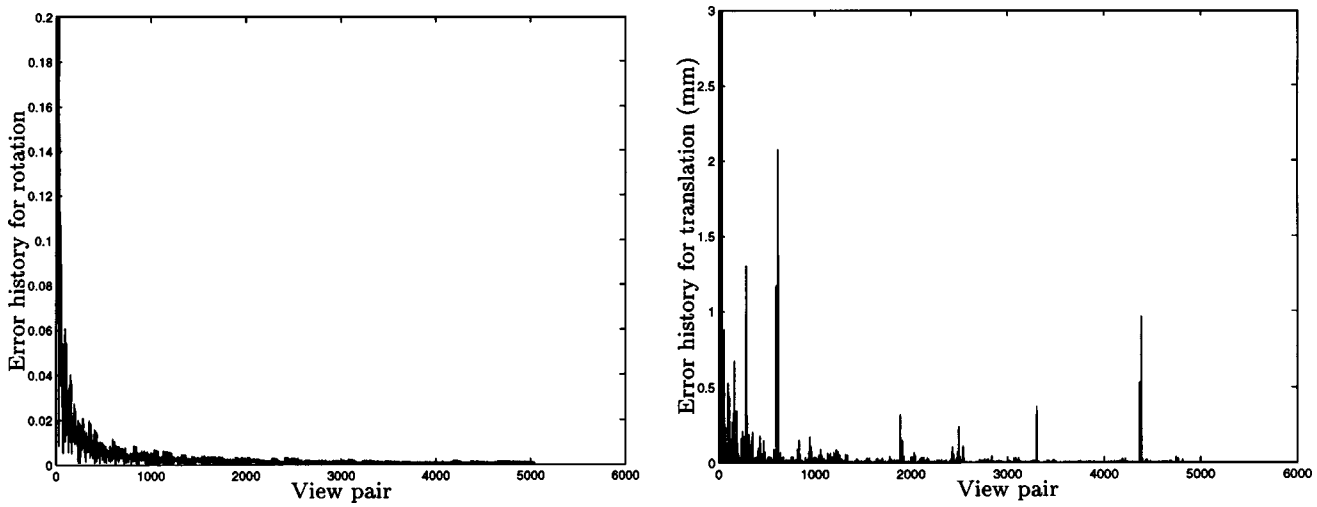
Obviously, the dodecahedron can only be used to refine an estimate of the range-finder orientation: because of its geometric regularity, we cannot tell one face from the other. This orientation estimate can be very rough,[4] within about 30°, and can therefore be done manually. Alternatively, one could choose to implement a rough hand–eye calibration manually and obtain the laser range-finder orientation estimate through the robot kinematic chain. We use the latter approach in this paper.

Combining the new measurements with the previous ones is useful to enhance the algorithm convergence rate. The specific formulation of the hand–eye problem reported in this paper uses *relative* displacements between measurements. For this reason, each combination of two absolute measurements (or physical measurements) provides a unique and new set of invariants $\{\mathbf{g}, \mathbf{h}\}$ relating frames $\mathcal{G}$ and $\mathcal{H}$ via the unknown hand–eye transform $\mathbf{Q}$, which can be added to the minimization. From an estimation point of view, the exhaustive combination of all the physical measurements ensures that the result of the minimization is independent of the ordering of the data. Fig. 5 shows the effect of exhaustively combining all the physical measurements on the convergence of the algorithm.

Figs. 6–8 show the results of the hand–eye calibration of our setup. The abscissa of all these plots is the view-pair number: each new physical measurement is combined with all the $N -$

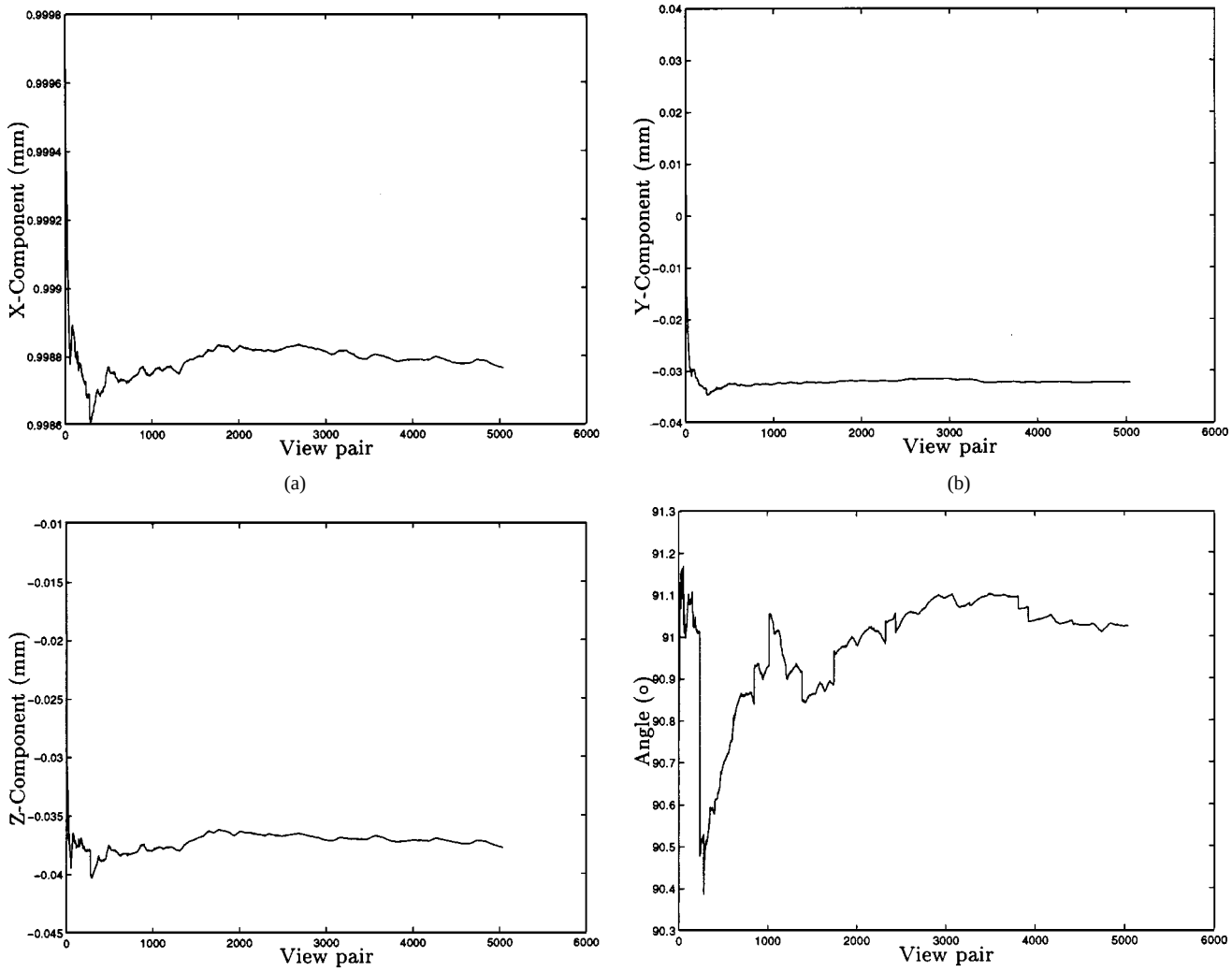
[3]If this were possible, the hand–eye calibration would be trivial.

[4]The orientation estimate should only be precise enough to distinguish between the faces of the dodecahedron which are 63° apart.



(a) (b)

Fig. 6. Estimation histories of the hand–eye algorithm: (a) the norm of the orientation error and (b) the norm of the translation error.



(a) (b) (c) (d)

Fig. 7. Estimated orientation parameters: (a) x; (b) y; (c) z-components of the unit vector $\hat{\mathbf{e}}$ of the axis of rotation of $\hat{\mathbf{Q}}$; and (d) the angle of rotation of $\hat{\mathbf{Q}}$.

1 previous ones to generate $N-1$ new relative motion pairs $\{\mathcal{G}, \mathcal{H}\}$ which we call view-pairs. The RLS algorithm is in fact applied to these view-pairs, turning N physical measurements into $(N-1)N/2$ view-pairs.

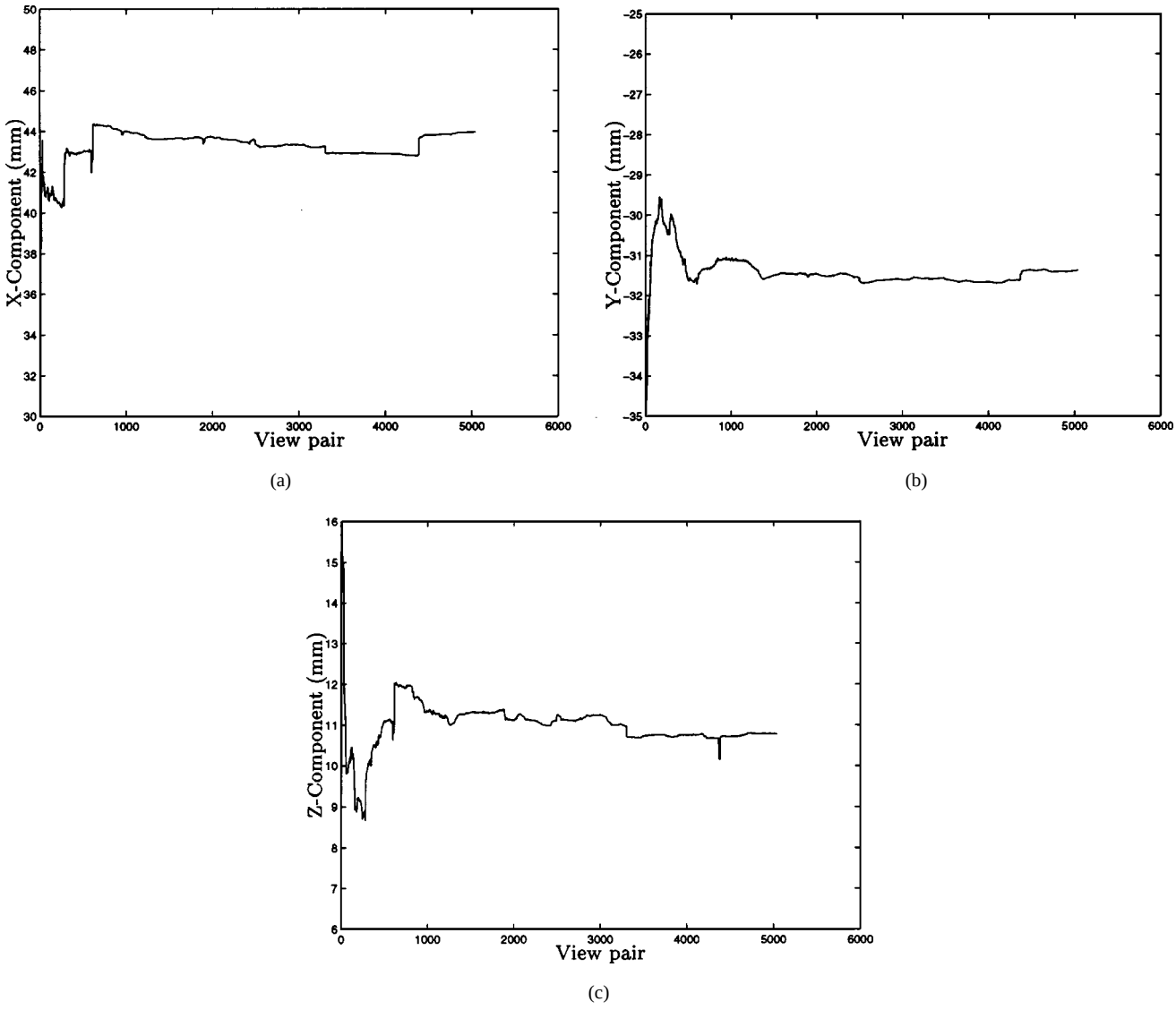


Fig. 8. Estimated translation parameters: (a) x; (b) y; and (c) z-components of $\hat{\mathbf{q}}$, in millimeters.

Plots (a) and (b) in Fig. 6 show the histories of the rotation and the translation error norms, as defined in (23) and (24), respectively. For this experiment, we acquired 100 raw range images, which resulted in 4950 view-pairs.

Fig. 7 shows the invariants of the rotation matrix $\hat{\mathbf{Q}}$ through the steps, while Fig. 8 shows the components of the translation vector $\hat{\mathbf{q}}$.

The jumps in Fig. 6(b) can be attributed to a wrist change of branch, in its two inverse kinematics solutions, between consecutive views. Although the wrist was calibrated prior to the experiment, the model used does not include a provision to tell the proximity of a singularity, and hence the risk of branching. The jumps are within the range of the average error encountered during the calibration of the wrist, i.e., 1.07 mm.

1) Evaluating the Hand–Eye Calibration: To test the hand–eye transform obtained, we took two range views of the test object shown in Fig. 9(a) and (b); then, we merged them using our calibration results. Fig. 9(c) shows the correspondence of these two views, where one view is shown in gray shading and the other one as a black grid. The error computed as the distance to the closest point in the other image is shown in Fig. 9(e) as a gray-scale rendering of the surface of the first view. We found an average error of 1.43 mm between the two surfaces.

This kind of result, while already sufficient for some robotic applications, can be used as a seed value to initiate a standard minimization [16] in order to refine it. The result of refining the transform found is shown in Fig. 9(d) and (f) and led to an improved average error of 0.72 mm. Such a minimization can be applied only if we already have a *good* estimate of the transform $\{\hat{\mathbf{Q}}, \hat{\mathbf{q}}\}$. The level of precision required on the estimate depends on the shape of the object and its texture. For some objects, an *infinite* precision is required because the surface provides no clue as to how to match two views (e.g., two views of a sphere). In practice, small repetitive surface patterns or surface noise can produce many local minima that can only be resolved by computing the best estimate possible. The more precise this estimate, the more likely the success of the minimization process to find the correct minimum for a larger class of objects.

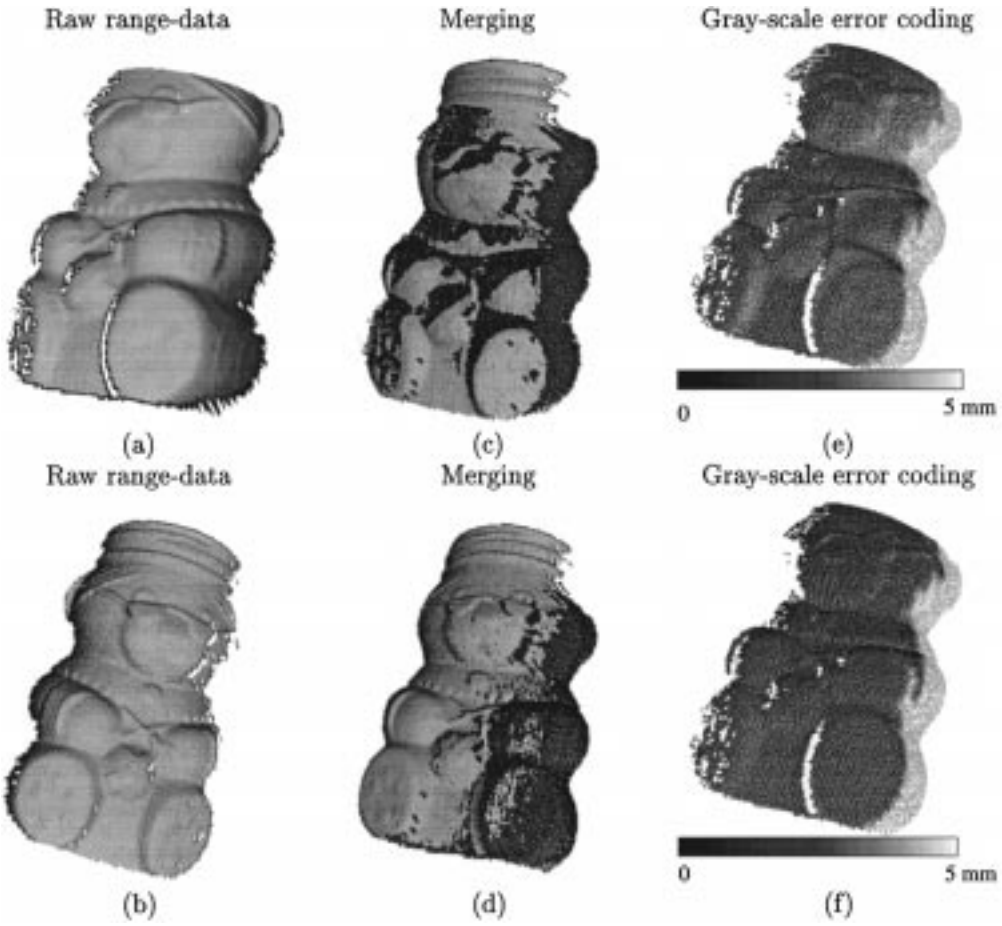


Fig. 9. Evaluation of the hand–eye calibration. (c) shows the result of merging the views shown in (a) and (b) using the hand–eye transform found. (e) shows a gray-scale coding of the merging error. (d) and (f) show similar results after refining the estimate obtained from the hand–eye transform. Note the subtle improvement in the merging errors from (c) to (d) and from (e) to (f).

VI. Conclusions

A recursive linear least-square solution to the hand–eye problem was proposed in this paper that is feasible for online implementation. The procedure is based on the linear invariants of rotation matrices. The online feasibility of the algorithm proposed here was demonstrated with the aid of experiments. The results show that, with the first three measurements, the relative orientation matrix is known with two digits of precision. Further measurements rapidly produce one more digit, but thence, the procedure stalls. We believe that the reasons for not gaining further digits lies in various sources of uncertainty, like the after-calibration geometric errors in the six-axis robot and the errors in the parameters of the laser range-finder. Contrary to methods reported in the literature that are *iterative*, ours is *recursive*.

Appendix
Basic Algorithm for the Recursive Estimation of a Vector Quantity

For completeness, we include below the recursive linear least-square algorithm used to estimate an n-dimensional vector $\mathbf{x}$ from a sequence of measurements. The algorithm reproduced below is taken from [17], if with some modifications, and one key correction: The coefficient of the second term of (43) is a scalar, but in the original it appears as a matrix inverse. At the outset we assume that an estimate of $\mathbf{x}$, labeled $\hat{\mathbf{x}}_N$, has been obtained from $N > n$ observations via the least-square approximation of an over-determined system of linear equations, namely

$$\mathbf{A}_N\hat{\mathbf{x}}_N = \mathbf{b}_N \tag{28}$$

where $\mathbf{A}_N$ is an $N \times n$ matrix and $\mathbf{b}_N$ is an N-dimensional vector. The least-square approximation of the above system, leading to the estimate $\hat{\mathbf{x}}_N$, can be expressed in terms of the Moore–Penrose generalized inverse in the form [18]

$$\hat{\mathbf{x}}_N = (\mathbf{A}_N^T\mathbf{A}_N)^{-1}\mathbf{A}_N^T\mathbf{b}_N. \tag{29}$$

Upon performing an additional observation, a correspondingly additional equation is derived, that leads to an updated estimation $\hat{\mathbf{x}}_{N+1}$

$$\mathbf{a}_{N+1}^T\hat{\mathbf{x}}_{N+1} = b_{N+1}. \tag{30}$$

If now the additional equation is adjoined to the first N equations, an over-determined system of $N+1$ equations in n unknowns is derived, i.e.,

$$\mathbf{A}_{N+1}\hat{\mathbf{x}}_{N+1} = \mathbf{b}_{N+1} \tag{31}$$

where $\mathbf{A}_{N+1}$ is now an $(N+1) \times n$ matrix and $\mathbf{b}_{N+1}$ is an $(N+1)$-dimensional vector, namely

$$\mathbf{A}_{N+1} \equiv \begin{bmatrix} \mathbf{A}_N \\ \mathbf{a}_{N+1}^T \end{bmatrix} \qquad \mathbf{b}_{N+1} \equiv \begin{bmatrix} \mathbf{b}_N \\ b_{N+1} \end{bmatrix}. \tag{32}$$

The least-square approximation of (31) is similar to that of (28)

$$\hat{\mathbf{x}}_{N+1} = (\mathbf{A}_{N+1}^T \mathbf{A}_{N+1})^{-1} \mathbf{A}_{N+1}^T \mathbf{b}_{N+1}. \tag{33}$$

The above expression is only of theoretical value and should not be used "verbatim." Indeed, it is possible to find $\hat{\mathbf{x}}_{N+1}$ without having to solve from scratch a new over-determined system, by using, instead, the previous estimate $\hat{\mathbf{x}}_N$, as described below: First, we expand the right-hand side of (33) by substituting $\mathbf{A}_{N+1}$ and $\mathbf{b}_{N+1}$ by their corresponding expressions, as given by (32), thus obtaining

$$\begin{aligned}\hat{\mathbf{x}}_{N+1} &= \left(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T\right)^{-1} \\ &\quad \times \left(\mathbf{A}_N^T \mathbf{b}_N + b_{N+1}\mathbf{a}_{N+1}\right) \\ &\equiv \left(\mathbf{A}_N^T \mathbf{A}_N\right)^{-1} \mathbf{A}_N^T \mathbf{b}_N - \left(\mathbf{A}_N^T \mathbf{A}_N\right)^{-1} \mathbf{A}_N^T \mathbf{b}_N \\ &\quad + \left(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T\right)^{-1} \\ &\quad \times \left(\mathbf{A}_N^T \mathbf{b}_N + b_{N+1}\mathbf{a}_{N+1}\right)\end{aligned}$$

where we have added the term $\left(\mathbf{A}_N^T \mathbf{A}_N\right)^{-1} \mathbf{A}_N^T \mathbf{b}_N$, and then subtracted it, thereby leaving the equation unperturbed. Upon rearrangement of the right-hand side, we obtain

$$\begin{aligned}\hat{\mathbf{x}}_{N+1} = &\left[(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} - (\mathbf{A}_N^T \mathbf{A}_N)^{-1}\right] \mathbf{A}_N^T \mathbf{b}_N \\ &+ (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} b_{N+1}\mathbf{a}_{N+1} \\ &+ (\mathbf{A}_N^T \mathbf{A}_N)^{-1} \mathbf{A}_N^T \mathbf{b}_N\end{aligned}$$

or, if we recall (29)

$$\begin{aligned}\hat{\mathbf{x}}_{N+1} = &\hat{\mathbf{x}}_N \\ &+ \left[(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} - (\mathbf{A}_N^T \mathbf{A}_N)^{-1}\right] \\ &\cdot \mathbf{A}_N^T \mathbf{b}_N \\ &+ (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} b_{N+1}\mathbf{a}_{N+1}.\end{aligned} \tag{34}$$

Furthermore, the second term of the right-hand side of the above equation is transformed into a more useful form upon factoring out the two inverses appearing in the brackets, the first one to the left, the second to the right. For ease of presentation, we do this in two steps

$$\begin{aligned}&\left[(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} - (\mathbf{A}_N^T \mathbf{A}_N)^{-1}\right] \mathbf{A}_N^T \mathbf{b}_N \\ &\quad = (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} \\ &\qquad \cdot \left[\mathbf{1} - (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)(\mathbf{A}_N^T \mathbf{A}_N)^{-1}\right] \mathbf{A}_N^T \mathbf{b}_N \\ &\quad = (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} \\ &\qquad \cdot \left[\mathbf{A}_N^T \mathbf{A}_N - (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)\right] \\ &\qquad \cdot (\mathbf{A}_N^T \mathbf{A}_N)^{-1} \mathbf{A}_N^T \mathbf{b}_N\end{aligned}$$

and, if we recall (29) again

$$\begin{aligned}&\left[(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} - (\mathbf{A}_N^T \mathbf{A}_N)^{-1}\right] \mathbf{A}_N^T \mathbf{b}_N \\ &\quad = -(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} \\ &\qquad \cdot \left[\mathbf{A}_N^T \mathbf{A}_N - (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)\right] \hat{\mathbf{x}}_N \\ &\quad = -(\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T \hat{\mathbf{x}}_N.\end{aligned} \tag{35}$$

Now we insert the foregoing expression into the equation giving $\hat{\mathbf{x}}_{N+1}$, (34), thereby obtaining

$$\begin{aligned}\hat{\mathbf{x}}_{N+1} = &\hat{\mathbf{x}}_N - (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T \hat{\mathbf{x}}_N \\ &+ (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} b_{N+1}\mathbf{a}_{N+1} \\ = &\hat{\mathbf{x}}_N + (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} \mathbf{a}_{N+1} \\ &\cdot \left[b_{N+1} - \mathbf{a}_{N+1}^T \hat{\mathbf{x}}_N\right].\end{aligned} \tag{36}$$

Let us introduce the definition

$$\mathbf{k}_N \equiv (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1} \mathbf{a}_{N+1} \tag{37}$$

which thus allows us to write (36) in the form

$$\begin{aligned}\hat{\mathbf{x}}_{N+1} &= \hat{\mathbf{x}}_N + \left[b_{N+1} - \mathbf{a}_{N+1}^T \hat{\mathbf{x}}_N\right] \mathbf{k}_N \\ &= (\mathbf{1} - \mathbf{k}_N \mathbf{a}_{N+1}^T)\hat{\mathbf{x}}_N + b_{N+1}\mathbf{k}_N.\end{aligned} \tag{38}$$

Moreover, let

$$\mathbf{P}_N \equiv (\mathbf{A}_N^T \mathbf{A}_N)^{-1}. \tag{39}$$

Hence,

$$\mathbf{P}_{N+1} \equiv (\mathbf{A}_{N+1}^T \mathbf{A}_{N+1})^{-1} = (\mathbf{A}_N^T \mathbf{A}_N + \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T)^{-1}. \tag{40}$$

In the following step, we obtain an expression for $\mathbf{P}_{N+1}$ in terms of $\mathbf{P}_N$. To do this, we first recall the *matrix-inversion lemma* [19]: Let $\mathbf{Q}$ and $\mathbf{R}$ be $n \times n$ positive-definite matrices. For an $n \times m$ arbitrary matrix $\mathbf{M}$, we have

$$\begin{aligned}&\mathbf{Q} - \mathbf{Q}\mathbf{M}(\mathbf{R} + \mathbf{M}^T \mathbf{Q}\mathbf{M})^{-1} \mathbf{M}^T \mathbf{Q} \\ &\quad \equiv \left[\mathbf{Q}^{-1} + \mathbf{M}\mathbf{R}^{-1}\mathbf{M}^T\right]^{-1}.\end{aligned} \tag{41}$$

We can now apply the matrix-inversion lemma to the expression derived above for $\mathbf{P}_{N+1}$, if we realize that matrix $\mathbf{A}_N$ can be safely assumed to be of full rank, which then leads to a positive-definite matrix $\mathbf{A}_N^T \mathbf{A}_N$. Therefore, from (40), we obtain directly

$$\begin{aligned}\mathbf{P}_{N+1} = &(\mathbf{A}_N^T \mathbf{A}_N)^{-1} - (\mathbf{A}_N^T \mathbf{A}_N)^{-1} \mathbf{a}_{N+1} \\ &\cdot \left[1 + \mathbf{a}_{N+1}^T (\mathbf{A}_N^T \mathbf{A}_N)^{-1} \mathbf{a}_{N+1}\right]^{-1} \mathbf{a}_{N+1}^T (\mathbf{A}_N^T \mathbf{A}_N)^{-1} \\ = &(\mathbf{A}_N^T \mathbf{A}_N)^{-1} - \frac{1}{D_N} (\mathbf{A}_N^T \mathbf{A}_N)^{-1} \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T \\ &\cdot (\mathbf{A}_N^T \mathbf{A}_N)^{-1}\end{aligned} \tag{42}$$

where D_N is defined as

$$D_N \equiv 1 + \mathbf{a}_{N+1}^T (\mathbf{A}_N^T \mathbf{A}_N)^{-1} \mathbf{a}_{N+1}.$$

If we now recall definition (39), the above expression for $\mathbf{P}_{N+1}$ becomes

$$\mathbf{P}_{N+1} = \mathbf{P}_N - \frac{1}{D_N} \mathbf{P}_N \mathbf{a}_{N+1}\mathbf{a}_{N+1}^T \mathbf{P}_N \tag{43}$$

which allows us to write $\mathbf{k}_N$ in the form

$$\begin{aligned}\mathbf{k}_N &= \mathbf{P}_{N+1}\mathbf{a}_{N+1} \\ &= \mathbf{P}_N\mathbf{a}_{N+1} - \frac{1}{D_N}\mathbf{P}_N\mathbf{a}_{N+1}\mathbf{a}_{N+1}^T\mathbf{P}_N\mathbf{a}_{N+1} \\ &= \left(1 - \frac{\mathbf{a}_{N+1}^T\mathbf{P}_N\mathbf{a}_{N+1}}{D_N}\right)\mathbf{P}_N\mathbf{a}_{N+1} \\ &= \frac{1}{D_N}\mathbf{P}_N\mathbf{a}_{N+1}. \end{aligned} \tag{44}$$

In summary, the algorithm allowing the computation of $\mathbf{x}$ recursively is outlined below: We assume that a first estimate $\hat{\mathbf{x}}_N$ of $\mathbf{x}$ is available from a set of $N > n$ measurements. Then

$$\mathbf{k}_N = \frac{1}{D_N}\mathbf{P}_N\mathbf{a}_{N+1} \tag{45}$$

$$\hat{\mathbf{x}}_{N+1} = (\mathbf{1} - \mathbf{k}_N\mathbf{a}_{N+1}^T)\hat{\mathbf{x}}_N + b_{N+1}\mathbf{k}_N \tag{46}$$

$$\mathbf{P}_{N+1} = \mathbf{P}_N - \frac{1}{D_N}\mathbf{P}_N\mathbf{a}_{N+1}\mathbf{a}_{N+1}^T\mathbf{P}_N. \tag{47}$$

A new set of measurements yields new values $\mathbf{a}_{N+1}$ and $\mathbf{b}_{N+1}$, with which an updated estimate can be obtained upon repeating the foregoing calculations from (45) to (47). The algorithm stops whenever $||\hat{\mathbf{x}}_{N+1} - \hat{\mathbf{x}}_N|| \leq \epsilon$, where ϵ is a prescribed tolerance.

ACKNOWLEDGMENT

The authors would like to thank the Canadian Institute for Robotics and Intelligent Systems, a Canadian Network of Centres of Excellence.

REFERENCES

[1] C. C. Slama, C. Theurer, and S. W. Henrikson, *Manual of Photogrammetry*. Falls Church, VA: American Society of Photogrammetry, 1980.
[2] B. K. P. Horn, *Robot Vision*. New York: McGraw-Hill, 1986.
[3] ——, "Closed form solution of absolute orientation using unit quaternions," *J. Opt. Soc. A*, vol. 4, no. 4, pp. 629–642, Apr. 1987.
[4] Y. C. Shiu and S. Ahmad, "Finding the mounting position of a sensor by solving a homogeneous transform equation of the form ax equals xb," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, 1987, pp. 1666–1671.
[5] ——, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 16–29, Feb. 1989.
[6] J. Angeles, *Fundamentals of Robotic Mechanical Systems Theory, Methods, and Algorithms*. New York: Springer-Verlag, 1997.
[7] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robot. Res.*, vol. 10, no. 3, pp. 240–254, 1991.
[8] R. Horaud and F. Dornaika, "Hand–eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.
[9] O. Faugeras and M. Hébert, "The representation, recognition and locating of 3D objects," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 27–51, 1986.
[10] H. Zhuang, "A note on 'hand–eye calibration'," *Int. J. Robot. Res.*, vol. 5, no. 16, pp. 725–727, 1997.
[11] J. Angeles, "On the use of invariance in robotics-oriented redundant sensing," in *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, 1989, pp. 599–604.
[12] C. Truesdell and R. Toupin, "The classical field theories," in *Encyclopedia of Physics III/1*, S. Fhigge, Ed. Berlin, Germany: Springler-Verlag, 1960, pp. 443–463.
[13] G. Strang, *Linear Algebra and Its Applications*. New York: Harcour Brace Jovanovich, 1988.
[14] N. J. Higham, "Computing the polar decomposition—With applications," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 4, pp. 1160–1174, Oct. 1986.
[15] J. Angeles, "Rigid-body pose and twist estimation in the presence of noisy redundant measurements," in *Proc. Eighth CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, Cracow, Poland, 1990, pp. 69–78.
[16] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 239–256, Feb. 1992.
[17] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*. Upper Saddle River, NJ: Prentice-Hall, 1997.
[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1989.
[19] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control*. New York: Hemisphere Press, 1975.

Jorge Angeles (SM'99) graduated as Engineer (electromechanical systems) and received the M.Eng. degree in mechanical engineering from the Universidad Nacional Autónoma de México (UNAM), Mexico, in 1969 and 1970, respectively. He received the Ph.D. degree in applied mechanics from Stanford University, Stanford, CA, in 1973.

Between 1973 and 1984, he taught at UNAM, where he also served as Chairman of the Graduate Division of Mechanical Engineering and Associate Dean of Graduate Studies in Electrical and Mechanical Engineering. Since 1984, he has been with the Department of Mechanical Engineering, McGill University, Montréal, PQ, Canada, where he is affiliated with the McGill Centre for Intelligent Machines. He has authored or co-authored various books in the areas of kinematics and dynamics of mechanical systems as well as numerous technical papers in refereed journals and conference proceedings. His research interests focus on the theoretical and computational aspects of multibody mechanical systems for purposes of design and control. Besides his research activities, Professor Angeles is a Consultant to various Canadian and international corporations in matters of automation, design, and robotics.

Dr. Angeles is an ASME Fellow, a Fellow of the Canadian Society of Mechanical Engineering, Past-President of International Federation for the Theory of Machines and Mechanisms (IFToMM), and member of various professional and learned societies. Professional registration as an engineer includes Quebec, Mexico, and Germany. In 1991, he received an Alexander von Humboldt Research Award that allowed him to spend one year as Visiting Professor at the Technical University of Munich, Munich, Germany.

Gilbert Soucy received the B.Ing. degree in electronics from Université de Sherbrooke, Sherbrooke, PQ, Canada, in 1987, and the M.Eng. degree in electrical engineering (computer vision) from McGill University, Montréal, PQ, Canada, in 1992.

Since 1991, he has been a Research Engineer in the Computer Vision Laboratory of the Centre for Intelligent Machines, McGill University. Soucy provides technical and theoretical support to a group of graduate students conducting research in computer vision. His research interests focus on 3-D computer vision, calibration, and multiple 3-D view integration. He has been involved in numerous research projects in collaboration with industry and the Canadian Space Agency (Strategic Technologies for Autonomous Robotics projects). Since 1997, he has also been a Consultant in computer vision. As such, he built, calibrated, and installed a laser range-finder for the Robotics Laboratory of Simon Fraser University, Burnaby, BC, Canada. He is currently consulting for Visionsphere Technologies Inc., a face-recognition company.

Mr. Soucy is a Registered Engineer in Quebec.

Frank P. Ferrie (M'84) received the B.Eng., M.Eng., and Ph.D. degrees in electrical engineering from McGill University, Montréal, PQ, Canada, in 1978, 1980, and 1986, respectively.

He was appointed Assistant Professor in the Department of Electrical and Computer Engineering in 1990 and Associate Professor in 1996. In 1998, he was appointed Director of the McGill Centre for Intelligent Machines. His research interests are in the area of computer vision, primarily in the areas of 2- and 3-D shape analysis, active perception, dynamic scene analysis, and machine vision. He has published extensively in these areas and is best known for his work in active vision and environmental modeling. He is also a Principal Investigator in the IRIS and GEOIDE Networks of Centers of Excellence.

Dr. Ferrie is a member of the IEEE Computer Society.