On-line Hand-Eye Calibration

Nicolas Andreff, Radu Horaud, Bernard Espiau
INRIA Rhône-Alpes/ GRAVIR-IMAG
655, av. de l'Europe, 38330 Montbonnot Saint Martin, France
email: firstname.lastname@inrialpes.fr

Abstract

In this paper, we address the problem of hand-eye calibration of a robot mounted video camera. In a first time, we derive a new linear formulation of the problem. This allows an algebraic analysis of the cases that usual approaches do not consider. In a second time, we extend this new formulation into an on-line hand-eye calibration method. This method allows to get rid of the calibration object required by the standard approaches and use unknown scenes instead. Finally, experimental results validate both methods.

1. Introduction

The goal of this paper is to determine the spatial relationship between a camera mounted onto a robot (Figure 1) and the robot end-effector without using any calibration object. This spatial relationship is a rigid transformation, composed of a rotation and a translation, known as the hand-eye transformation.

The practical procedure for on-line hand-eye calibration is the following. The hand-eye device moves with known robot motions. While it moves, the camera observes an unknown rigid scene. Then, a structure from motion process determines both the camera motion and the Euclidean structure of the scene. The camera motion parameters thus extracted are combined with the known robot motions to estimate the hand-eye transformation.

This method has several advantages when compared to the standard approaches [19, 4, 8, 10, 16, 5, 20, 15]. The first one is that there is no need to use a calibrated object with known geometry. One does not either need to perform 2D to 3D matching between each one of the images and the object points. On the opposite, structure from motion requires only image-to-image point tracking. Moreover, since structure from motion can accommodate with bad intrinsic camera parameters, our method is also tolerant to such camera calibration errors. Finally, the method can be used with small robot motions. Indeed, rotations are represented by their associated (3×3) orthogonal matrices, which are always defined. On the opposite, the reduced representations, used in the standard approaches, based on the axis/angle decomposition of a rotation are ill-defined in the case of small rotations.

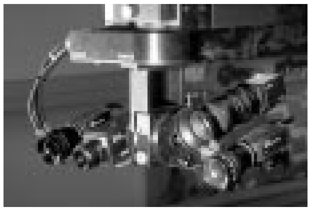

Figure 1. Some cameras mounted on our 5 DOF robot.

Notice that our method is less general than the one proposed in [21]. Indeed, the latter performs both the hand-eye calibration and the camera calibration. However, it is done to the cost of a complicated non-linear minimization whereas our method is purely linear and allows algebraic analysis.

There are two main contributions in this paper. The first one is a new formulation of hand-eye calibration. It states the problem as a linear system, thus allowing an analysis of what can be recovered according to the type and number of camera/robot motions. The second contribution is to derive from this new formulation the on-line hand-eye calibration process.

The remainder of this paper decomposes as follows. Section 2 recalls the classical formulation of hand-eye calibration and briefly describes the methods that will be used as references. Section 3 gives the new linear formulation and contains its algebraic analysis. The on-line hand-eye calibration is presented in Section 4. Finally, Section 5 gives some experimental results and

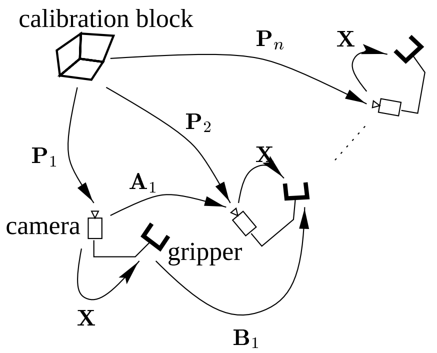

Figure 2. From images of a calibration object, the corresponding camera poses P_i can be computed, providing the rigid motions of the camera A_i. With the associated recorded robot motions B_i, the hand-eye transformation can be computed.

Section 6 concludes this work.

2. Classical hand-eye calibration

We present here the usual approach [19, 4, 8, 10, 16, 5, 20] which states that when the camera undergoes a rigid motion $\mathbf{A} = (\mathbf{R}_a, \mathbf{t}_a)$ and the corresponding robot motion is $\mathbf{B} = (\mathbf{R}_b, \mathbf{t}_b)$, then they are conjugated by the hand-eye transformation $\mathbf{X} = (\mathbf{R}_x, \mathbf{t}_x)$:

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

In the prior work, correspondences are established between the 3D points on the calibration object and their respective projection in each image i. From the exactly known intrinsic camera parameters, the pose (i.e. position and orientation) of the calibration object is then estimated ($\mathbf{P}_i = (\mathbf{R}_i, \mathbf{t}_i)$). The camera motion between image $i-1$ and image i is hence obtained by simple composition: $\mathbf{A}_i = \mathbf{P}_i \mathbf{P}_{i-1}^{-1}$ (see Figure 2). Finally, equation (1) is solved in several ways. We rapidly present the three methods that will be used as references.

The historical method [19] splits equation (1) applied to each motion i in two:

$$\begin{aligned} \mathbf{R}_{ai}\mathbf{R}_x &= \mathbf{R}_x\mathbf{R}_{bi} && (2) \\ \mathbf{R}_{ai}\mathbf{t}_x + \mathbf{t}_{ai} &= \mathbf{R}_x\mathbf{t}_{bi} + \mathbf{t}_x && (3) \end{aligned}$$

The first equation is solved by linear least square minimization of a system obtained by using the axis/angle representation of the rotations:

$$\begin{aligned} S(\mathbf{n}_{a_i} + \mathbf{n}_{b_i})\mathbf{n}_x &= \mathbf{n}_{a_i} - \mathbf{n}_{b_i},\ \forall i = 1..n && (4) \\ \theta_x &= 2\arctan(\|\mathbf{n}_x\|) && (5) \end{aligned}$$

where n is the number of motions; $\mathbf{n}_\bullet, \theta_\bullet$ are the axis and angle of the rotation $\mathbf{R}_\bullet$ and $S(\bullet)$ is the skew-symmetric matrix associated to a 3-vector. Once $\mathbf{R}_x$ is known, the second equation is also solved with linear least squares techniques.

To avoid this two-stage resolution which spreads the error on the rotation estimation to translation, a non-linear minimization method based on the representation of the rotations with unit quaternions was proposed [10]. It minimizes the following cost function:

$$\begin{aligned} f(\mathsf{q}_x, \mathbf{t}) &= \lambda_1 \Sigma_{i=1}^n \|\mathbf{n}_{a_i} - \mathsf{q}_x * \mathbf{n}_{b_i} * \bar{\mathsf{q}}_x\|^2 \\ &+ \lambda_2 \Sigma_{i=1}^n \|\mathsf{q}_x * \mathbf{t}_{b_i} * \bar{\mathsf{q}}_x - (\mathbf{R}_{a_i} - \mathbf{I})\mathbf{t} - \mathbf{t}_{a_i}\|^2 \\ &+ \lambda(1 - \mathsf{q}_x^T \mathsf{q}_x)^2 && (6) \end{aligned}$$

where the new notations are: q_x is the quaternion associated to $\mathbf{R}_x$ and $\bar{\mathsf{q}}_x$ is its conjugate; λ is a Lagrange multiplier; λ_1 and λ_2 are user-chosen weights. The first term of this cost function provides an error on the hand-eye orientation in the quaternion space. The second term is minimized to obtain the hand-eye translation. Finally, the last term ensures that the solution q is a unit quaternion.

The last reference method is based on the unit dual quaternion representation of rigid motions [8]. Dual quaternions are formed by a couple $\tilde{q} = (\mathsf{q}, \mathsf{q}')$ of quaternions, where q is the quaternion associated to the rotation $\mathbf{R}$ and q', known as dual part, is also a quaternion and encodes the translation $\mathbf{t}$. Both q and q' have a scalar part q_0 (resp. q_0') and a vector part $\mathbf{q}$ (resp. $\mathbf{q}'$). To actually represent a rigid transformation a dual quaternion must be unitary: $\mathsf{q}^T\mathsf{q} = 1$ and $\mathsf{q}^T\bar{\mathsf{q}'} = 0$.

Note $\tilde{a}_i$ the unit dual quaternion associated to the ith camera motion $\mathbf{A}_i$, $\tilde{b}_i$ the unit dual quaternion associated to the ith robot motion $\mathbf{B}_i$ and $\tilde{q}_x$ unit dual quaternion associated to the hand-eye transformation. Then, (1) rewrites for each motion i as $\tilde{a}_i = \tilde{q}_x \tilde{b}_i \bar{\tilde{q}}_x$ and yields:

$$\begin{pmatrix} \mathbf{a}_i - \mathbf{b}_i & S(\mathbf{a}_i + \mathbf{b}_i) & 0_{3\times1} & 0_{3\times3} \\ \mathbf{a}_i' - \mathbf{b}_i' & S(\mathbf{a}_i' + \mathbf{b}_i') & \mathbf{a}_i - \mathbf{b}_i & S(\mathbf{a}_i + \mathbf{b}_i) \end{pmatrix} \begin{pmatrix} \mathsf{q}_x \\ \mathsf{q}_x' \end{pmatrix} = 0 \tag{7}$$

This system has a 2-dimensional kernel from which the solution is extracted by applying the unity constraint.

Note that whichever method is used, the hand-eye calibration problem intrinsically requires at least 2 motions, thereby 3 images, and these motions must have different rotation axes. This was shown algebraically [19] and geometrically [4].

3. A new linear formulation

3.1. Formulation

The classical formulations are based on the axis/angle representation of a rotation, either explicitly [10, 8] or implicitly [19]. However, such reduced representations are ill-defined when the angle of rotations tend to zero, i.e. in the case of small rotations of the hand-eye device.

Therefore, we propose in this section a new formulation which handles such cases. The new formulation is inspired by the resemblance of (1) with the Sylvester equation: $\mathbf{UV} + \mathbf{VW} = \mathbf{T}$. This matrix equation, which often occurs in system theory, is usually formulated as a linear system:

$$(\mathbf{U} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{W}) vec(\mathbf{V}) = vec(\mathbf{T}). \tag{8}$$

The operator *vec* was introduced in [12] and reorders (one line after the other) the coefficients of a $(m \times n)$ matrix $\mathbf{M}$ into the mn vector $vec(\mathbf{M}) = (M_{11}, \ldots, M_{1n}, M_{21}, \ldots, M_{mn})^T$.

The $\otimes$ product is the Kronecker (or tensor) product [2, 3]. From two matrices $\mathbf{M}$ and $\mathbf{N}$ with respective dimensions $(m \times n)$ and $(o \times p)$, it defines the resulting $(mo \times np)$ matrix:

$$\mathbf{M} \otimes \mathbf{N} = \begin{pmatrix} M_{11}\mathbf{N} & \ldots & M_{1n}\mathbf{N} \\ \vdots & \ddots & \vdots \\ M_{m1}\mathbf{N} & \ldots & M_{mn}\mathbf{N} \end{pmatrix} \tag{9}$$

Using some properties of this product, equations (2)–(3) rewrites, for all motion i, as the homogeneous linear system:

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{a\,i} \otimes \mathbf{R}_{b\,i} & \mathbf{0}_{9\times 3} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b_i}^T) & \mathbf{I}_3 - \mathbf{R}_{a\,i} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{9\times 1} \\ \mathbf{t}_{a\,i} \end{pmatrix} \tag{10}$$

3.2. Algebraic analysis

From earlier work [19, 4], we know that 2 motions with non parallel rotation axes are necessary to determine the hand-eye transformation. Our new formulation allows to give an alternative proof of this result (see [1]) but also allows to define what can be obtained when such a necessary conditions is not fulfilled.

In this section, we determine what can be obtained using, first, only pure translations, second, two independent pure rotations of the robot, third, two independent planar motions (i.e. containing the same rotational part and two independent translations) and finally two independent general motions with non parallel rotation axes. The results of this study are summed up in Table 1.

Pure translations – In this case, the upper part of the system in (10) and its lower-left block vanish. Hence, it simplifies into:

$$\left(\mathbf{I}_3 \otimes (\mathbf{t}_{b_i}^T)\right) vec\mathbf{R}_x = \mathbf{t}_{a\,i} \tag{11}$$

which has rank 3 when $\mathbf{t}_{b\,i} \neq 0$ since

$$\mathbf{I}_3 \otimes (\mathbf{t}_{b_i}^T) = \begin{pmatrix} \mathbf{t}_{b_i}^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{t}_{b_i}^T & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{t}_{b_i}^T \end{pmatrix} \tag{12}$$

Consequently, three independent pure translations are enough to determine the nine coefficients of the hand-eye rotation $\mathbf{R}_x$ while the hand-eye translation $\mathbf{t}_x$ can not be obtained.

Pure rotations of the robot – In this case, $\mathbf{t}_{b\,i} = 0$ and the system in (10) is block-diagonal and decouples into:

$$\begin{aligned} (\mathbf{I}_9 - \mathbf{R}_{a\,i} \otimes \mathbf{R}_{b\,i})\, vec(\mathbf{R}_x) &= \mathbf{0}_{9\times 1} \quad (13) \\ (\mathbf{I}_3 - \mathbf{R}_{a\,i})\, \mathbf{t}_x &= \mathbf{t}_{a\,i} \quad (14) \end{aligned}$$

Consequently, two rotations with non parallel axes yield a unique solution for the hand-eye translation $\mathbf{t}_x$ since the second subsystem has then full rank.

Let us now study the first subsystem. One of the properties of the Kronecker product is that the eigenvalues of $\mathbf{M} \otimes \mathbf{N}$ are the product of the eigenvalues of $\mathbf{M}$ by those of $\mathbf{N}$. In our case, $\mathbf{R}_{a\,i}$ and $\mathbf{R}_{b\,i}$ have the same eigenvalues: $\{1, e^{i\theta_i}, e^{-i\theta_i}\}$ and thus the eigenvalues of $\mathbf{R}_{a\,i} \otimes \mathbf{R}_{b\,i}$ are: $\{1, 1, 1, e^{i\theta_i}, e^{i\theta_i}, e^{-i\theta_i}, e^{-i\theta_i}, e^{2i\theta_i}, e^{-2i\theta_i}\}$. Consequently, when the angle of rotation θ_i is not a multiple of π, then the matrix $\mathbf{I}_9 - \mathbf{R}_{a\,i} \otimes \mathbf{R}_{b\,i}$ has rank 6.

In the case of two or more independent rotations, the subsystem has rank 8 (the proof can be found in [1]). Let $\mathbf{v}$ be a vector of its kernel and $\mathbf{V} = vec^{-1}(\mathbf{v})$ the associated (3×3) matrix. This matrix $\mathbf{V}$ is invertible and its column are orthogonal (see [1]) but has not a unit determinant. Hence, the solution of the system is the normalized matrix based on $\mathbf{V}$:

$$\mathbf{R}_x = \frac{sign(\det(\mathbf{V}))}{|\det(\mathbf{V})|^{\frac{1}{3}}} \mathbf{V} \tag{15}$$

In practice, $\mathbf{v}$ can be determined using a Singular Value Decomposition (SVD) which is known to accurately estimate the kernel of a linear mapping.

In conclusion, in the case of two or more pure rotations of the robot, the hand-eye rotation and translation can be completely recovered and their numerical estimations are decoupled. Notice that by inverting the roles of the robot and the camera, one obtains the same results in the case of pure rotations of the camera for the recovery of the *eye-hand* transformation.

Planar motions – In this case, all the camera (resp. robot) rotations are equal: $\mathbf{R}_{ai} = \mathbf{R}_{a1}$ (resp. $\mathbf{R}_{bi} = \mathbf{R}_{b1}$) and have the same rotation axis $\mathbf{n}_a$ (resp. $\mathbf{n}_b$), which is orthogonal to the plane of motion. Considering two motions ($i = 1..2$), the system in (10) thus rewrites as:

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{a1} \otimes \mathbf{R}_{b1} & \mathbf{0}_{9\times 3} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) & \mathbf{I}_3 - \mathbf{R}_{a1} \\ \mathbf{I}_9 - \mathbf{R}_{a1} \otimes \mathbf{R}_{b1} & \mathbf{0}_{9\times 3} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T) & \mathbf{I}_3 - \mathbf{R}_{a1} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{9\times 1} \\ \mathbf{t}_{a1} \\ \mathbf{0}_{9\times 1} \\ \mathbf{t}_{a2} \end{pmatrix} \quad (16)$$

which is equivalent to:

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{a1} \otimes \mathbf{R}_{b1} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T - \mathbf{t}_{b1}^T) \end{pmatrix} vec(\mathbf{R}_x) = \begin{pmatrix} \mathbf{0}_{9\times 1} \\ \mathbf{t}_{a2} - ta_1 \end{pmatrix} \quad (17)$$

$$\begin{pmatrix} \mathbf{I}_3 - \mathbf{R}_{a1} \end{pmatrix} \mathbf{t}_x = \begin{pmatrix} \mathbf{t}_{a1} - \mathbf{R}_x \mathbf{t}_{b1} \end{pmatrix} \quad (18)$$

If the two motions are independent, then the first subsystem is of full rank and yield a unique solution for $\mathbf{R}_x$. On the opposite, the second subsystem is underconstrained. Hence, it admits as solution any vector of the form

$$\mathbf{t}_x(\alpha) = \mathbf{t}_\perp + \alpha \mathbf{n}_a \quad (19)$$

where α is any scalar value and $\mathbf{t}_\perp$ is a solution in the plane of motion. It is unique since the system has rank 2 and the plane of motion is 2-dimensional. In practice, this vector can be obtained by an SVD of matrix $\mathbf{I}_3 - \mathbf{R}_{a1}$ [14, §2.6].

The general case – In the case of two independent general motions with non-parallel axes, there exist a unique solution to the hand-eye calibration problem. Using our formulation, we could therefore solve the whole system in (10) with a linear least-square minimization technique. However, this solution is not independent of the physical unit used for the translation [21] and since it provides a solution which does not guarantee that the estimated $\mathbf{R}_x$ is an orthogonal matrix. Then, one has to perform an orthogonalization of the result but it is improbable to find the corresponding correction on the hand-eye translation estimation.

On the opposite, a two-step solution, as in [19], guarantees an orthogonal estimate of the hand-eye rotation. Indeed, the first step consists in determining the hand-eye rotation as in the case of pure rotations, which had this property. As for the second step, it consists in the estimation of the hand-eye translation knowing the estimated hand-eye rotation using a linear least-square minimization. Notice that, here, the solution is not decoupled any more as in the pure rotation case.

	Translation $\mathbf{R}_B = \mathbf{I}$ $\mathbf{t}_B \neq 0$	Rotation $\mathbf{R}_B \neq \mathbf{I}$ $\mathbf{t}_B = 0$	General motion $\mathbf{R}_B \neq \mathbf{I}$ $\mathbf{t}_B \neq 0$
Translation $\mathbf{R}_B = \mathbf{I}$ $\mathbf{t}_B \neq 0$	$\mathbf{R}_X$ (only if a third independent translation is available)	$\mathbf{R}_X$ $\mathbf{t}_X(\alpha)$	$\mathbf{R}_X$ $\mathbf{t}_X(\alpha)$
Rotation $\mathbf{R}_B \neq \mathbf{I}$ $\mathbf{t}_B = 0$	$\mathbf{R}_X$ $\mathbf{t}_X(\alpha)$	$\mathbf{R}_X, \mathbf{t}_X$ Decoupled solution	$\mathbf{R}_X, \mathbf{t}_X$ General two-step solution
General motion $\mathbf{R}_B \neq \mathbf{I}$ $\mathbf{t}_B \neq 0$	$\mathbf{R}_X$ $\mathbf{t}_X(\alpha)$	$\mathbf{R}_X, \mathbf{t}_X$ General two-step solution	$\mathbf{R}_X, \mathbf{t}_X$ General two-step solution

Table 1. *Summary of the results for two independent motions.*

4. On-line hand-eye calibration

In this section, we extend the new linear formulation in order to deal with an unknown scene. As the scene is now unknown, pose estimation has to be replaced by a structure from motion algorithm.

In the case of a single moving calibrated camera we address here, the relevant algorithm is Euclidean 3D reconstruction [7, 17, 13, 18, 11, 6]. Provided that the intrinsic camera parameters are roughly determined and do not vary during the recording of an image sequence, Euclidean 3D reconstruction estimates from this sequence the structure of the scene, assumed static, and the relative motions between each of the camera positions. However, the size of the scene can not be recovered and therefore, the camera translations are estimated up to a scale factor. Since this unknown scale factor is often unique, we will only treat this case, even though our method can be easily extended to the case where several unknown scale factors are to be determined.

Assuming a single unknown scale factor, only the scaled camera translations $\mathbf{u}_{ai} = \mathbf{t}_{ai}/\|\mathbf{t}_{a1}\|$ are hence determined while the camera rotations $\mathbf{R}_{ai}$ are fully recovered. Therefore, instead of the camera motions $\mathbf{A}_i$ given through pose estimation, only the scaled motions $\tilde{\mathbf{A}}_i = (\mathbf{R}_{ai}, \mathbf{u}_{ai})$ are obtainable. The problem of on-line hand-eye calibration is then illustrated by Figure 3.

The link between $\mathbf{A}_i$ and $\tilde{\mathbf{A}}_i$ is thus given through

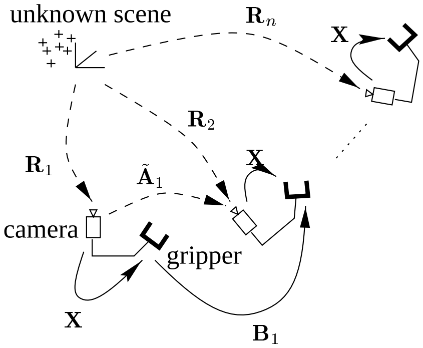

Figure 3. From images of an unknown scene and the knowledge of the intrinsic parameters of the camera, Euclidean 3D reconstruction computes, up to an unknown scale factor, the relative positions of the camera R_i with respect to its initial location R_1. From them, one can only compute the camera motions $\tilde{\mathbf{A}}_i$, up to the unknown scale factor, but still retrieve the hand-eye transformation.

the unknown scale factor λ:

$$\mathbf{A}_i = (\mathbf{R}_{a\,i}, \lambda \mathbf{u}_{a\,i}) \tag{20}$$

With this relation, the linear formulation (10) is changed into:

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{a\,i} \otimes \mathbf{R}_{b\,i} & \mathbf{0}_{9\times3} & \mathbf{0}_{3\times1} \\ \mathbf{I}_3 \otimes \mathbf{t}_{b\,i}^{\;T} & \mathbf{I}_3 - \mathbf{R}_{a\,i} & -\mathbf{u}_{a\,i} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \\ \lambda \end{pmatrix} = \mathbf{0}_{12\times1} \tag{21}$$

This system is solved in two steps as in the new linear formulation to guarantee the orthogonality of the estimation of $\mathbf{R}_x$.

5. Experimentation

Error measurement To measure the errors in translation, we chose the usual relative error: $\|\hat{\mathbf{t}} - \mathbf{t}\|/\|\mathbf{t}\|$, where the '$\hat{\ }$' notation represents the estimated value.

As for the errors in orientation, no canonical measure is defined. We chose the quaternion norm used in [8]: $\|\hat{\mathsf{q}} - \mathsf{q}\|$ for its simplicity and its direct relation to α, the angle of the residual rotation between these two orientations. Indeed, if $\hat{\mathsf{q}}$ and q are unitary, then $\|\hat{\mathsf{q}} - \mathsf{q}\| = 2 - 2\cos\frac{\alpha}{2}$.

As no ground-truth value is available for comparison, we compared, for each motion i, $\mathbf{A}_i\mathbf{X}$ and $\mathbf{X}\mathbf{B}_i$. We then aggregated all these errors into RMS errors.

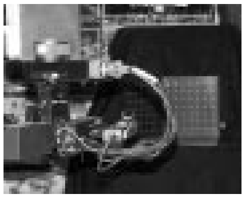

Figure 4. In Experiment 1, the camera observes a calibration grid.

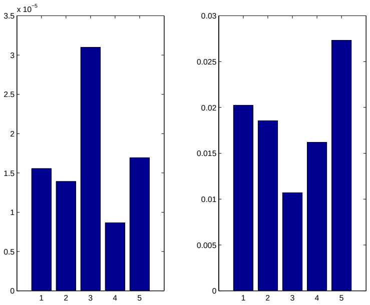

Figure 5. RMS errors in rotation (left) and translation (right) with 33 images of a calibration grid for each method (see text).

Experiment 1 To evaluate the correctness of the solution obtained by on-line calibration, we had to compare it with those obtained by classical calibration methods with the same data.

Hence, we took images of our calibration grid (Figure 4) and performed hand-eye calibration with the axis/angle method [19] (**M1**), the dual quaternion method [8] (**M2**) and the non-linear minimization [10] (**M3**). We also performed classical hand-eye calibration by replacing the first stage of (**M1**) by the SVD based solution (**M4**). Finally, using the same points, extracted from the calibration grid, but not its 3D model, we applied the on-line hand-eye calibration method (**M5**). The Euclidean 3D reconstruction method we used is the one proposed in [6].

The results obtained for a trajectory of 33 positions are given in Figure 5. These positions were chosen as far as possible from each other according to the advice given in [19]. It can be seen that (**M4**) gives the smallest error in rotation due to the efficiency of the SVD and thus obtains also a reduced error in translation. As for (**M5**), it gives larger errors, as expected since the 3D model is not used. However, the degradation is rather small.

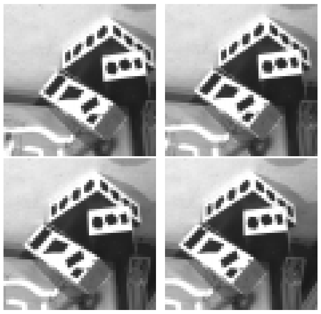

Figure 6. A sequence of 4 images used for on-line hand-eye calibration in Experiment 2.

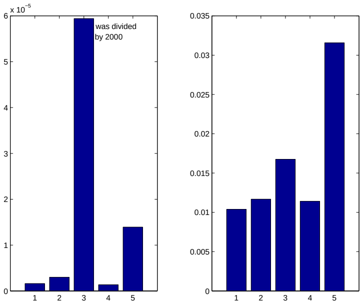

Figure 7. RMS errors in rotation (left) and translation (right) with 4 images (see text).

Experiment 2 In a second experiment, we tested (**M5**) with more realistic images. Four positions were defined where the images shown in Figure 6 were taken. In the first image, points were extracted and then tracked during the motion between each position of the camera. Then, on-line calibration was performed upon the tracked points.

In a goal of comparison, the blocks were replaced by the calibration grid and the robot was moved anew to the four predefined positions. Then, hand-eye calibration was performed with the images taken there.

The results of this experiment are given in Figure 7. They show an awful behavior of the non linear minimization method, probably due to the small number of data. They also show a higher degradation of the performance of (**M5**) compared to the others, but still in an acceptable ratio.

6. Conclusion

We proposed a on-line hand-eye calibration method which allows to reduce the human supervision compared with classical calibration methods. Its derivation includes a new linear formulation of classical hand-eye calibration which is linear and numerically efficient. From its algebraic analysis, we determined the parts of the hand-eye transformation that can be obtained from a reduced number of motions which does not allow a complete calibration.

However, one difficulty with the Euclidean 3D reconstruction with a moving camera is to be able to find reliable point correspondences between images. The method proposed in [6] solves this problem by tracking points along the motion. However, it requires that the points are tracked from the beginning until the end of the robot trajectory. This is a hard constraint since, in practice, one hardly obtains enough points after a long trajectory.

Stereo-vision may offer the answer to this problem since it was shown that Euclidean reconstruction can be performed, without any prior knowledge, from two rigid motions of a stereo pair [9]. This is fully in coherence with our constraints. Moreover, this kind of method releases the constraint on the presence of points along the whole sequence of images.

Finally, there is a pending question which was never answered: “What are the motions for (on-line) hand-eye calibration that yield the higher numerical accuracy ?”

Acknoledgments

We acknowledge support from the European Community through the Esprit-IV reactive LTR project number 26247.

References

[1] N. Andreff, B. Espiau, and R. Horaud. Une méthode d’auto-étalonnage pince-caméra. Research Report 3507, INRIA, October 1998.

[2] R. Bellman. *Introduction to matrix analysis*. McGraw-Hill, 1960.

[3] J. W. Brewer. Kronecker products and matrix calculus in system theory. *Theoretical Comput. Sci.*, CAS-25(9):772–781, 1978.

[4] H. Chen. A screw motion approach to uniqueness analysis of head-eye geometry. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA*, pages 145–151, June 1991.

[5] J. Chou and M. Kamel. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *The International Journal of Robotics Research*, 10(3):240–254, June 1991.

[6] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, November 1996.

[7] N. Cui, J. Weng, and P. Cohen. Recursive-batch estimation of motion and structure from monocular image sequences. *Computer Vision and Image Understanding*, 59(2):154–170, March 1994.

[8] K. Daniilidis and E. Bayro-Corrochano. The dual quaternion approach to hand-eye calibration. In *Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria*, pages 318–322, 1996.

[9] F. Devernay and O. Faugeras. From projective to euclidean reconstruction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, pages 264–269, June 1996.

[10] R. Horaud and F. Dornaika. Hand-eye calibration. *The International Journal of Robotics Research*, 14(3):195–210, June 1995.

[11] J. Koenderink and A. van Doorn. Affine structure from motion. *Journal of the Optical Society of America A*, 8(2):377–385, 1991.

[12] H. Neudecker. A note on Kronecker matrix product and matrix equation systems. *SIAM J. Appl. Math.*, 17(3):603–606, 1969.

[13] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In J. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, pages 97–108. Springer-Verlag, May 1994.

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[15] S. Rémy, M. Dhome, J. Lavest, and N. Daucher. Hand-eye calibration. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1057–1065, 1997.

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

[17] C. Taylor, D. Kriegman, and P. Anandan. Structure and motion in two dimensions from multiple images: A least squares approach. In *Proceedings of the* IEEE *Workshop on Visual Motion, Princeton, New Jersey*, pages 242–248. IEEE Computer Society Press, October 1991.

[18] C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. In *Proceedings of the* IEEE *Workshop on Visual Motion, Princeton, New Jersey*, pages 21–28, Los Alamitos, California, USA, October 1991. IEEE Computer Society Press.

[19] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. IEEE *Transactions on Robotics and Automation*, 5(3):345–358, 1989.

[20] C. Wang. Extrinsic calibration of a robot sensor mounted on a robot. IEEE *Transactions on Robotics and Automation*, 8(2):161–175, April 1992.

[21] G. Wei, K. Arbter, and G. Hirzinger. Active self-calibration of robotic eyes and hand-eye relationships with model identification. IEEE *Transactions on Robotics and Automation*, 14(1):158–166, 1998.