Towards the Embedding of On-Line Hand-Eye Calibration into Visual Servoing

Nicolas Andreff

Towards the Embedding of On-Line Hand-Eye Calibration into Visual Servoing

Nicolas Andreff
INRIA Rhône-Alpes / IMAG – GRAVIR
655, avenue de l'Europe, 38330 Montbonnot, FRANCE

Abstract

This work is related to the visual servoing of a robot hand-mounted camera. The control law of this robotic system exhibits the necessity of determining the position/orientation of a camera reference frame with respect to the control reference frame. The hand-eye calibration consists in the determination of this transformation (i.e. a rotation and a translation).

We provide in this paper an on-line calibration method in two stages: first, an initial estimation is computed with two self-calibration movements, then the estimated hand-eye transformation is updated using the controlled motions of the robot. As the latter generate low amplitude rotations, the classical formulations are not efficient any more. The homogeneous matrix equation AX=XB appearing in usual approaches is therefore reformulated into a linear system and used as a basis of a Kalman filter which is to be run in parallel with the visual servoing.

Preliminary simulation results were obtained and show the good behaviour of the method.

1 Introduction

To control a robot by visual servoing, one has to know or to estimate several parameters ruling the control laws. In the case of a visual system rigidly mounted onto a robot arm, some of these parameters are related to the robot itself (*robot calibration*), some to the visual system (*camera calibration*) and some to the spatial relation between the robot and the visual system (*hand-eye calibration*).

More specifically, hand-eye calibration consists in determining the rigid transformation (i.e. a rotation and a translation) between a robot reference frame (usually, the end-effector frame) and the visual system reference frame (usually, in monocular systems, the camera frame).

The current approaches [1, 2, 6, 7, 9, 10, 11, 12] operate within a standard off-line calibration scheme. However, we would like to free ourselves from the human intervention required by such approaches. The goal of this paper is therefore to propose an on-line calibration method, i.e. without a-priori knowledge or any calibration object (*self-calibration*), and during the accomplishment of the robotic task.

From the viewpoint of a self-calibration procedure, the current solutions suffer from the drawback to need large motions with significant differences between each other, i.e. the rotation axis associated to each displacement should be different from the rotation axes of the other movements. Moreover, the higher accuracy in the estimation of the hand-eye transformation is demanded, the more motions are required ! The method we propose consists of a self-calibration stage where only two predefined movements are needed followed by an on-line updating stage using the controlled robot motions.

The previous approaches use either the axis/angle or the quaternion representations of a rotation to solve the well-known $\mathbf{AX} = \mathbf{XB}$ homogeneous matrix equation, and thus, need to avoid robot and camera motions with low amplitude rotation. (An exception is [9] where a method is presented which avoids the $\mathbf{AX} = \mathbf{XB}$ formulation, but requires a non-linear minimization performed with the Levenberg-Marquardt algorithm.) However, controlled robot motions will be of low amplitude and will only contain *small rotations*. In the axis/angle representation, this means that the robot motions will have a rotation angle close to $0°$, where the rotation axis can not be defined. It is therefore interesting to find a new formulation of the homogeneous matrix equation which will lead to the building of a Kalman filter.

The remainder of this paper is organized as follows: section 2 is devoted to the problem formulation, section 3 contains a complete set of simulation results and section 4 is the conclusion.

2 Problem formulation

2.1 Visual servoing and hand-eye calibration

Recall the simplest visual servoing control law under the task-function approach with exponential conver-

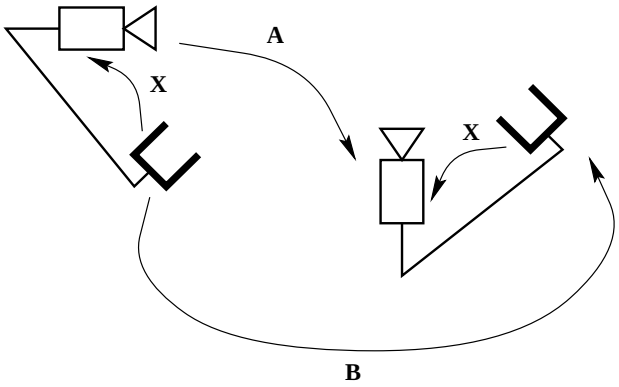

Figure 1: Two positions of the same hand-eye system linked by a camera displacement $\mathbf{A}$ associated to a robot motion $\mathbf{B}$ via the hand-eye transformation $\mathbf{X}$.

gence [4]:

$$\left\{ \begin{array}{rcl} \vec{\mathbf{T}}_c & = & -\lambda \vec{\mathbf{e}} \\ \vec{\mathbf{e}} & = & \mathbf{L}^{T^+}(\vec{\mathbf{s}} - \vec{\mathbf{s}}^*) \end{array} \right.$$

where $\vec{\mathbf{T}}_c$ is the kinematic screw applied to the robot (expressed in the end-effector frame), λ is a gain factor, $\vec{\mathbf{e}}$ is the task function (to be regulated to 0), $\mathbf{L}^T$ is the interaction matrix (expressed in the end-effector frame) and $\vec{\mathbf{s}}$ (resp. $\vec{\mathbf{s}}^*$) is the current (resp. desired) image primitive.

The interaction matrix $\mathbf{L}^T$ is a function of the camera intrinsic parameters (namely, the scale factors α_u and α_v, and the angle θ between the image axes), a canonical interaction matrix expressed in the camera frame ($\mathbf{L}_c^T$, see [4] for details) and the hand-eye transformation ($\mathbf{X} = (\mathbf{R}_X, \vec{\mathbf{t}_X})$):

$$\mathbf{L}^T = \begin{pmatrix} -\alpha_u & \alpha_u cotg\theta \\ 0 & -\frac{\alpha_v}{sin\theta} \end{pmatrix} \cdot \mathbf{L}_c^T \cdot \begin{pmatrix} \mathbf{R}_X & -\mathbf{R}_X \mathbf{A}_s(-\mathbf{R}_X{}^T \vec{\mathbf{t}_X}) \\ 0 & \mathbf{R}_X \end{pmatrix}$$

where $\mathbf{A}_s(\bullet)$ denotes the skew matrix.

The intrinsic parameters are allowed to be only roughly determined since they do not have a strong influence on the stability of the system. However, hand-eye transformation has to be accurately estimated [3]. A way to obtain it is to consider a pair of robot and camera positions (see Figure 1).

As the robot performs the rigid motion $\mathbf{B} = (\mathbf{R}_B, \vec{\mathbf{t}_B})$, the camera undergoes a displacement determined by the rigid transformation $\mathbf{A} = (\mathbf{R}_A, \vec{\mathbf{t}_A})$. $\mathbf{A}$ and $\mathbf{B}$ are related by:

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

The latter equation is an homogeneous matrix equation and, thus, can be rewritten as one equation dealing with the rotational part and another with the translational part:

$$\mathbf{R}_X - \mathbf{R}_A \mathbf{R}_X \mathbf{R}_B{}^T = 0 \tag{2}$$

$$\mathbf{R}_X \vec{\mathbf{t}_B} + \vec{\mathbf{t}_X} - \mathbf{R}_A \vec{\mathbf{t}_X} = \vec{\mathbf{t}_A} \tag{3}$$

We assume for the rest of this paper that the robot motion is measurable and that the camera motion can be estimated.

Since controlled motions often contain small rotations, the purpose of the method proposed here is to deal with such ill-defined cases. To achieve this goal, the homogeneous matrix equation (1) can be reformulated as a linear system of the form:

$$\mathbf{C}\vec{\mathbf{x}} = \vec{\mathbf{y}}$$

where $\vec{\mathbf{x}}$ is a 12-vector containing the 9 coefficients of the rotation matrix $\mathbf{R}_X$ and the 3 coefficients of the translation $\vec{\mathbf{t}_X}$.

2.2 New formulation

Define $\vec{\mathbf{i}}$, $\vec{\mathbf{j}}$ and $\vec{\mathbf{k}}$, the row vectors of the rotation $\mathbf{R}_X$, $\mathbf{R}_X = \begin{pmatrix} \vec{\mathbf{i}}^T \\ \vec{\mathbf{j}}^T \\ \vec{\mathbf{k}}^T \end{pmatrix}$ and $\vec{\mathbf{x}} = \begin{pmatrix} \vec{\mathbf{i}}^T & \vec{\mathbf{j}}^T & \vec{\mathbf{k}}^T & \vec{\mathbf{t}_X}^T \end{pmatrix}^T$.

With these notations, equations (2) and (3) are rewritten as:

$$\underbrace{\begin{pmatrix} \mathbf{I}_9 - \mathbf{M} & \mathbf{0}_{9\times 3} \\ \mathbf{N} & \mathbf{I}_3 - \mathbf{R}_A \end{pmatrix}}_{\mathbf{C}} \vec{\mathbf{x}} = \underbrace{\begin{pmatrix} \vec{\mathbf{0}}_{9\times 1} \\ \vec{\mathbf{t}_A} \end{pmatrix}}_{\vec{\mathbf{y}}} \tag{4}$$

where

$$\mathbf{M} = \begin{pmatrix} R_{A_{(1,1)}}\mathbf{R}_B & R_{A_{(1,2)}}\mathbf{R}_B & R_{A_{(1,3)}}\mathbf{R}_B \\ R_{A_{(2,1)}}\mathbf{R}_B & R_{A_{(2,2)}}\mathbf{R}_B & R_{A_{(2,3)}}\mathbf{R}_B \\ R_{A_{(3,1)}}\mathbf{R}_B & R_{A_{(3,2)}}\mathbf{R}_B & R_{A_{(3,3)}}\mathbf{R}_B \end{pmatrix},$$

$$\mathbf{N} = \begin{pmatrix} \vec{\mathbf{t}_B}^T & \vec{\mathbf{0}}_{1\times 3} & \vec{\mathbf{0}}_{1\times 3} \\ \vec{\mathbf{0}}_{1\times 3} & \vec{\mathbf{t}_B}^T & \vec{\mathbf{0}}_{1\times 3} \\ \vec{\mathbf{0}}_{1\times 3} & \vec{\mathbf{0}}_{1\times 3} & \vec{\mathbf{t}_B}^T \end{pmatrix},$$

$\mathbf{I}_n$ is the $(n \times n)$ identity matrix and the notation $R_{A_{(p,q)}}$ means the coefficient located on the p^{th} row and the q^{th} column of matrix $\mathbf{R}_A$.

The previous expression is derived by expanding the matrix equations (2) and (3). This exhibits linear relations between the coefficients of vector $\vec{\mathbf{x}}$. Then, these relations just have to be collected into the linear system above.

This formulation does not ensure that the estimated row vectors $\vec{\mathbf{i}}$, $\vec{\mathbf{j}}$ and $\vec{\mathbf{k}}$ will fulfill the orthogonality constraint on the rotation matrix $\mathbf{R}_X$. However, it has the advantage of offering a linear expression of the problem, thus allowing the use of a Kalman filter. As for the orthogonality constraint, it will be discussed in section 2.5.

2.3 Kalman filter

A Kalman filter is built upon this model and the assumption that the rigid transformation $\mathbf{X}$ remains constant as:

$$\begin{cases} \vec{\mathbf{x}}_{k+1} &= \vec{\mathbf{x}}_k + \vec{\mathbf{v}}(k) \\ \vec{\mathbf{y}}(k) &= \mathbf{C}_k \vec{\mathbf{x}}_k + \vec{\mathbf{n}}(k) \end{cases} \quad (5)$$

where the subscript k denotes the value after the k^{th} robot/camera motion, $\vec{\mathbf{v}}$ represents the noise induced by modeling errors and $\vec{\mathbf{n}}$ represents the noise in the measurements. $\vec{\mathbf{x}}$ is defined in (4) and will be referred to as *state vector* in the following. $\vec{\mathbf{y}}$ is called the *measurement vector* and is also defined in (4).

A solution of this filter is given by:

$$\begin{aligned} \mathbf{P}(0) &= \mathbf{G}_0 \\ \mathbf{K}(k) &= \mathbf{P}(k)\mathbf{C}(k)^T \left[\mathbf{C}(k)\mathbf{P}(k)\mathbf{C}(k)^T + \mathbf{G}_2\right]^{-1} \\ \mathbf{P}(k+1) &= \mathbf{P}(k) + \mathbf{G}_1 \\ &- \mathbf{K}(k)\left[\mathbf{C}(k)\mathbf{P}(k)\mathbf{C}(k)^T + \mathbf{G}_2\right]\mathbf{K}(k)^T \\ \vec{\mathbf{x}}(k+1|k) &= \left[\mathbf{I}_{12} - \mathbf{K}(k)\mathbf{C}(k)\right]\vec{\mathbf{x}}(k|k-1) \\ &+ \mathbf{K}(k)\vec{\mathbf{y}}(k) \end{aligned}$$

where $\vec{\mathbf{x}}(k+1|k)$ is the estimate at time $k+1$ of the state-vector $\vec{\mathbf{x}}$, knowing all the required information at time k. $\mathbf{G}_0$ is an initial estimate for $\mathbf{P}$ (which is taken as $\mathbf{I}_{12}$), $\mathbf{G}_1$ is the covariance matrix associated to the noise $\vec{\mathbf{v}}$ in the state equation and $\mathbf{G}_2$ is the covariance matrix associated to the noise $\vec{\mathbf{n}}$ in the measurement equation.

An initial estimation is needed in this method. In order to cope with the desire of keeping the number of self-calibration movements low and still have a good precision for this initial estimation, the best thing to do is to use one of the classical methods with only 2 movements.

2.4 A model of noise

In the Kalman filter noises appear that we must model. The following assumptions will be made: $\vec{\mathbf{n}}$ and $\vec{\mathbf{v}}$ are uncorrelated, robot motion is free of noise (which is not too much of an assumption since the uncertainties in the robot motion are transferred into the unknown hand-eye transformation), and, finally, the camera motion estimation process is not required to fulfill the orthogonality constraint on the rotation matrix. The latter assumption allows one to use additive noise on each coefficient of the camera rotation matrix instead of additive noise on the rotation angles, thus simplifying the analysis.

As $\vec{\mathbf{n}}$ and $\vec{\mathbf{v}}$ are uncorrelated, it is straightforward to set $\vec{\mathbf{v}}$ as a Gaussian noise with a diagonal (12×12) covariance matrix $\mathbf{G}_1$. Then, we can express the noise $\vec{\mathbf{n}}$ in the measurement equation. To achieve this goal, first insert noise in the camera motion:

$$\begin{aligned} \mathbf{R}_A &= \tilde{\mathbf{R}}_A + \mathbf{\Xi}_a \\ \vec{\mathbf{t}}_A &= \tilde{\vec{\mathbf{t}}}_a + \vec{\mathbf{v}}_a \end{aligned}$$

where the notation $\tilde{\bullet}$ represents the estimated or measured value and where $\mathbf{\Xi}_a$ and $\vec{\mathbf{v}}_a$ are uncorrelated Gaussian noises and have respective covariance matrices $\mathbf{\Gamma}_1(9 \times 9)$ and $\mathbf{\Gamma}_2(3 \times 3)$. Then, feed equations (2) and (3) with the latter two equations and obtain:

$$\begin{aligned} \mathbf{R}_X - \tilde{\mathbf{R}}_A \mathbf{R}_X {\mathbf{R}_B}^T &= \mathbf{\Xi}_a \mathbf{R}_X {\mathbf{R}_B}^T \\ \mathbf{R}_X \vec{\mathbf{t}}_B + \vec{\mathbf{t}}_X - \tilde{\mathbf{R}}_A \vec{\mathbf{t}}_X &= \tilde{\vec{\mathbf{t}}}_a + \vec{\mathbf{v}}_a + \mathbf{\Xi}_a \vec{\mathbf{t}}_X \end{aligned}$$

For the purpose of analysis, assume, temporarily, that $\mathbf{R}_X$ and $\vec{\mathbf{t}}_X$ are known. Denote $\vec{\mathbf{v}}_a{}' = \vec{\mathbf{v}}_a + \mathbf{\Xi}_a \vec{\mathbf{t}}_X$ and $\vec{\xi}_a$ the vector form of $\mathbf{\Xi}_a \mathbf{R}_X {\mathbf{R}_B}^T$:

$$\forall (i,j) \in \{1,2,3\}^2, (\vec{\xi}_a)_{3(i-1)+j} = \left(\mathbf{\Xi}_a \mathbf{R}_X {\mathbf{R}_B}^T\right)_{ij}$$

Hence, $\vec{\mathbf{n}} = \left(\vec{\xi}_a{}^T, \vec{\mathbf{v}}_{\mathbf{a}}{}'^T\right)^T$.

The covariance matrix associated to $\vec{\mathbf{n}}$ according to the noise model is:

$$\mathbf{G}_2 = \begin{pmatrix} \mathbf{V}^T \mathbf{\Gamma}_1 \mathbf{V} & \mathbf{V}^T \mathbf{\Gamma}_1 \mathbf{W} \\ \mathbf{W}^T \mathbf{\Gamma}_1 \mathbf{V} & \mathbf{\Gamma}_2 + \mathbf{W}^T \mathbf{\Gamma}_1 \mathbf{W} \end{pmatrix}$$

where

$$\mathbf{V} = \begin{pmatrix} \mathbf{R}_X {\mathbf{R}_b}^T & 0 & 0 \\ 0 & \mathbf{R}_X {\mathbf{R}_b}^T & 0 \\ 0 & 0 & \mathbf{R}_X {\mathbf{R}_b}^T \end{pmatrix}$$

and

$$\mathbf{W} = \begin{pmatrix} \vec{\mathbf{t}}_X{}^T & 0 & 0 \\ 0 & \vec{\mathbf{t}}_X{}^T & 0 \\ 0 & 0 & \vec{\mathbf{t}}_X{}^T \end{pmatrix}.$$

As $\mathbf{R}_X$ and $\vec{\mathbf{t}}_X$ are unknown and $\mathbf{R}_B$ is time-varying, they are approximated by: $\mathbf{R}_X \approx \mathbf{I}_3$, $\vec{\mathbf{t}}_X \approx 0$ and $\mathbf{R}_B \approx \mathbf{I}_3$. Under this approximation, the covariance matrix $\mathbf{G}_2$ is chosen as: $\mathbf{G}_2 = \begin{pmatrix} \Gamma_1 & 0 \\ 0 & \Gamma_2 \end{pmatrix}$

2.5 From Kalman state vector to rigid transformation

After each iteration (i.e. robot/camera motion), the Kalman Filter above determines a new state vector $\vec{\mathbf{x}}_k$. This state vector contains, as seen earlier, the current estimation of the 3 row vectors ($\vec{\mathbf{i}}$, $\vec{\mathbf{j}}$ and $\vec{\mathbf{k}}$) of the rotation matrix $\mathbf{R}_X$ and the current estimation of the translation vector $\vec{\mathbf{t}}_X$.

However, it is not granted that the estimated row vectors satisfy the necessary condition of orthonormality. As the raw estimates can not be used as an approximation of a rotation, an orthogonalization has to

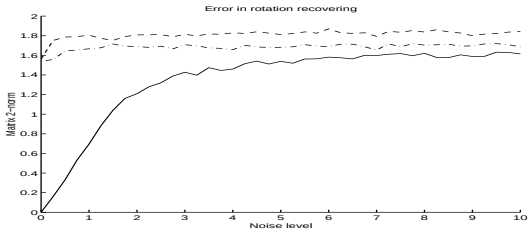

Figure 2: Error in the recovering of a rotation matrix according to several methods: SVD decomposition (dashed line), QR decomposition (dash-dotted line) and minimization (solid line).

be made. In order to keep the optimal convergence of the Kalman Filter, this operation has to be applied outside the filter loop.

The orthonormalization step consists in finding a rotation matrix $\mathbf{R}$ which fits the best the row vectors $\vec{\mathbf{i}}, \vec{\mathbf{j}}, \vec{\mathbf{k}}$. One can use the Gram-Schmidt orthogonalization method but it is not numerically robust. One should therefore use either an SVD or a QR decomposition [8] or the minimization method presented in [5], which is the most robust (see Figure 2).

The underlying idea of the latter method is to find the rotation matrix $\mathbf{R}$ which satisfies:

$$\mathbf{R} \cdot \begin{pmatrix} \vec{\mathbf{i}}^T \\ \vec{\mathbf{j}}^T \\ \vec{\mathbf{k}}^T \end{pmatrix}^T = \mathbf{I}_{3\times 3}$$

This problem leads to a minimization problem in the quaternion space of the form:

$$\min_{\mathbf{q}}(\mathbf{q}^T \mathbf{B} \mathbf{q})$$

3 Simulation results

Now that the orthonormality constraint is guaranted, the axis/angle or quaternion formalisms can be used to quantify the simulation results. In the remaining, we will consider: relative errors in translation, computed as $\|\vec{\mathbf{t}_X} - \tilde{\vec{\mathbf{t}_X}}\|/\|\vec{\mathbf{t}_X}\|$, errors in rotation expressed in the quaternion representation and computed as $\|\mathbf{q}_X - \tilde{\mathbf{q}_X}\|$.

3.1 Sensitivity to measurement noise

In order to study the sensitivity of the method to noise in the measurements, 50 simulations were run with several noise levels. Noise in the rotation part and noise in the translation part have been studied separately.

Robot motions are defined by choosing Roll-Pitch-Yaw angles according to Gaussian laws of standard deviation equal to $2\pi/100$ and translation coefficients according to Gaussian laws of standard deviation equal to 1 length unit.

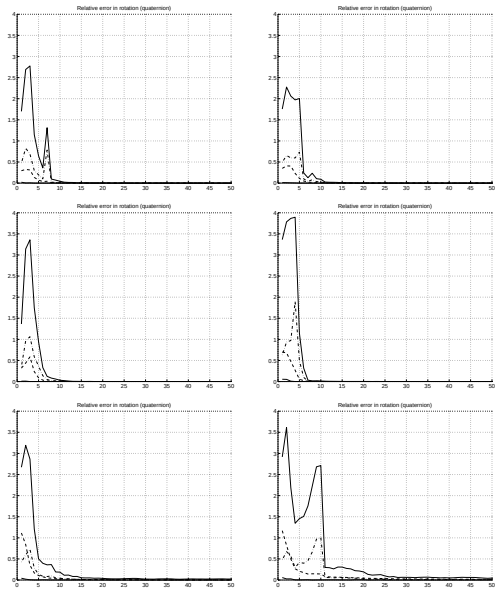

Figure 3: Evolution of the relative error in the rotation (expressed as a quaternion) with various levels of noise in the rotation: 0, 0.01, 0.05, 0.1, 0.5 and 1. Maximal and minimal errors are plotted with solid line, mean value errors with dashed line and RMS errors with dash-dotted line.

Hand-eye transformations to be recovered are also chosen randomly with Roll-Pitch-Yaw angles between 0 and 2π and translation coefficients have standard deviation equal to 100 length unit.

Sensitivity to noise in the rotation part. Here, the noise in the translation measurements is kept constant (i.e. a measurement error in the camera translation $\vec{\mathbf{v}_a}$ is randomly chosen with 0.1 length unit standard deviation, which is the value expected by the Kalman filter) and 6 cases were studied. They all correspond to a different choice of standard deviation while randomly choosing the measurement error in the camera rotation $\mathbf{\Xi}_a$. The chosen values for the standard deviations are respectively: 0, 0.01, 0.05, 0.1, 0.5, 1, while the Kalman filter is tuned for a 0.1 standard deviation (i.e. the correlation matrix is chosen diagonal with non-zero coefficents equal to 0.01). (Recall that this noise is added to a rotation matrix, which therefore has unit row vectors, i.e. its coefficients vary most of the time between -1 and 1.)

Figures 3 and 4 show the results of these simulations.

Sensitivity to noise in the translation part. A Gaussian noise is added to each coefficient of the camera translation. Fifty simulations were run for 6 levels of noise (i.e. standard deviation equal to 0, 0.01, 0.05, 0.1, 0.5 and 1).

These values are to be compared with the camera

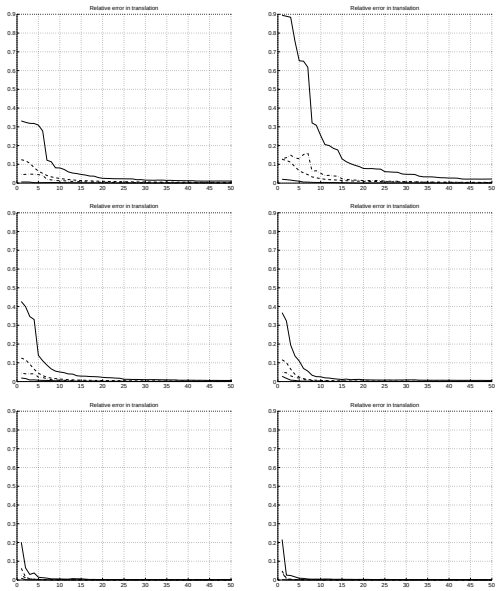

Figure 4: Evolution of the relative error in the translation vector with various levels of noise in the rotation: 0, 0.01, 0.05, 0.1, 0.5 and 1. Maximal and minimal errors are plotted with solid line, mean value errors with dashed line and RMS errors with dash-dotted line.

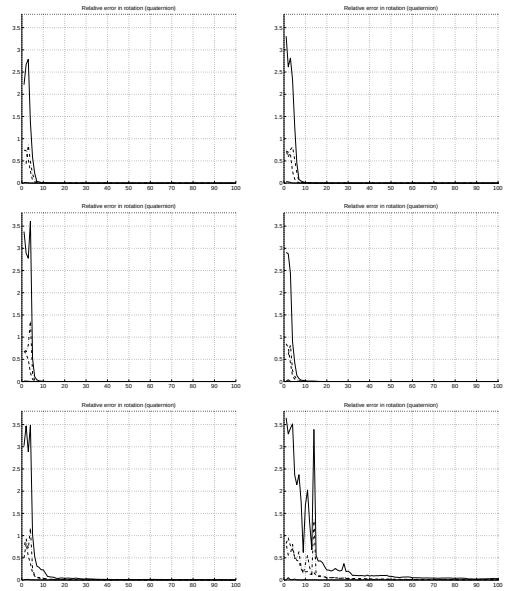

Figure 5: Evolution of the relative error in the rotation (expressed as a quaternion) with various levels of noise in the translation: 0, 0.01, 0.05, 0.1, 0.5 and 1. Maximal and minimal errors are plotted with solid line, mean value errors with dashed line and RMS errors with dash-dotted line.

translations. However, it is not easy since they depend on the robot motions and the hand-eye transformation. Nevertheless, we can have an idea of the amplitude of the camera translations. Recalling (3) and approximating $\mathbf{R}_A$ by the identity matrix in it leads to $\vec{\mathbf{t}_A} \approx \mathbf{R}_X \vec{\mathbf{t}_B}$, thus considering that camera and robot translations have almost the same amplitudes. In the present case, they have standard deviation equal to 1.

Figures 5 and 6 show the results of these simulations.

3.2 Sensitivity to initial estimate

This section deals with sensitivity to initial estimates. In order to study it, the reference hand-eye transformation was disturbed by adding white noise to its Roll-Pitch-Yaw angles and to its translation coefficients. The standard deviation of these disturbances are defined as a percentage of the standard deviation used in the determination of the hand-eye transformation.

Here again, fifty simulations were run for each of the following error percentages: 0%, 5%, 10%, 50%, 75% and 100%. The results can be found in Figures 7 and 8.

3.3 Realistic simulations

The previous section shows the good numerical behaviour of the proposed method, but it still remains to study more realistic cases. Aiming to this, one should choose a realistic initial estimation of the hand-eye transformation then use it with realistic robot motions.

Computing an initial estimation. As discussed above, this method can not completely free itself from the classical approaches since it needs an initial estimation of the hand-eye transformation, but it requires a smaller number of calibration movements. Here, two methods are compared: the 'historical' one [11], based on the axis/angle representation of the rotation and [2] which is based on a dual quaternion representation of the rotation. The accurate non-linear method [7] is left apart since it uses a Levenberg-Marquardt minimization, thus taking too much time for an accuracy increase out of the present needs.

Tests have been run to determine which of the two methods should be used. They were done with various noise levels and, for each noise level, 500 simulations were realized. Each simulation consisted in choosing randomly a hand-eye transformation, computing a disturbed camera motion from it and two predefined self-calibration moves, recovering the hand-eye transformation from robot and camera motions with both methods and, finally, comparing the results (see Figure 9).

Realistic robot motions. In this experiment, a hand-eye transformation is chosen randomly. Then, realistic robot motions are chosen: according to the control law, the components of the kinematic screw decrease exponentially from their initial values; there-

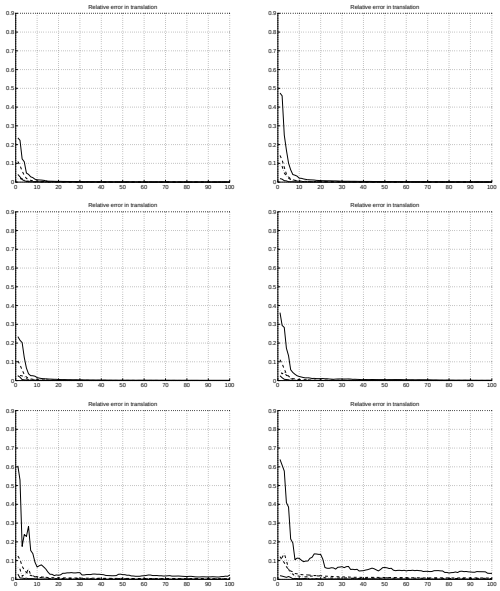

Figure 6: Evolution of the relative error in the translation vector with various levels of noise in the translation: 0, 0.01, 0.05, 0.1, 0.5 and 1. Maximal and minimal errors are plotted with solid line, mean value errors with dashed line and RMS errors with dash-dotted line.

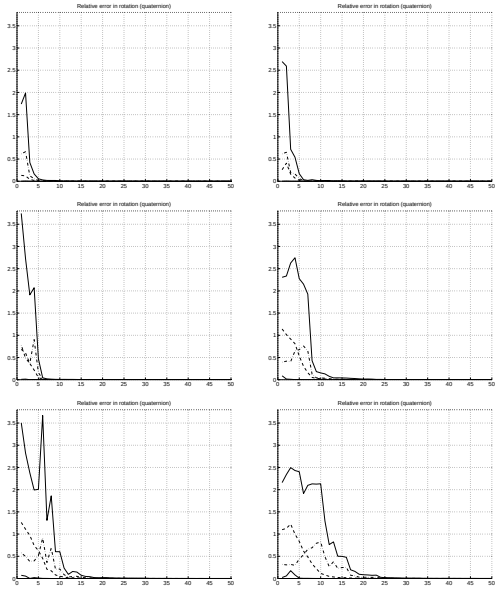

Figure 7: Evolution of the relative error in the rotation (expressed as a quaternion) with various error percentages in the initial estimates: 0%, 5%, 10%, 50%, 75% and 100%. Maximal and minimal errors are plotted with solid line, mean value errors with dashed line and RMS errors with dash-dotted line.

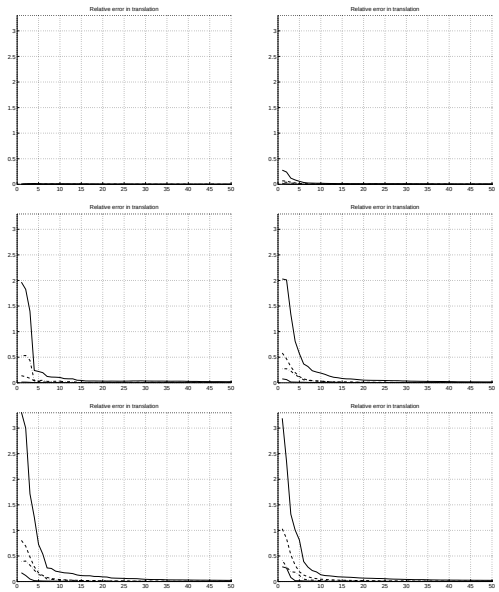

Figure 8: Evolution of the relative error in the translation vector with various error percentages in the initial estimates: 0%, 5%, 10%, 50%, 75% and 100%. Maximal and minimal errors are plotted with solid line, mean value errors with dashed line and RMS errors with dash-dotted line.

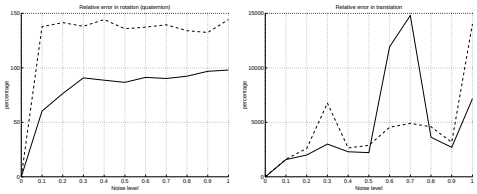

Figure 9: Comparison of two classical approaches in the case of two calibration moves with various noise levels: [11] (dashed line) and [2] (solid line). Errors in rotation (left) and in translation (right) are plotted.

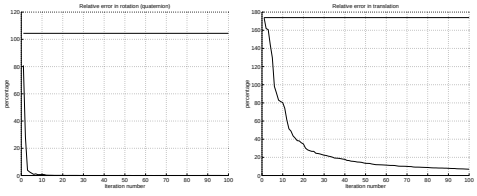

Figure 10: Simulation with realistic robot motions: error in rotation (left) and in translation (right). The horizontal lines represent the initial errors after 2 calibration movements.

fore, an initial kinematic screw is chosen randomly as the starting point of a simulated convergent robot motion. As these robot motions are determined, the camera motions can be computed. Noise is then added to them and the obtained values are fed with the robot motions into the filter.

The whole process is preceded by 2 self-calibration movements which determine, according to [2] (since previous tests shows its higher robustness), an initial estimate of the hand-eye transformation.

The results of this experiment are shown in Figure 10.

4 Conclusion

This was the first attempt to solve for the hand-eye calibration problem in the context of a hand-mounted camera. The linear formulation which has been proposed permits an on-line improvement of an initial estimation. The latter results from the use of one of the current approaches in a self-calibration scheme where the robot/camera set-up is only given two predefined movements.

The simulation results are promising, showing the interest of such an approach. The model of noise should however be modified to cope with noise on the Roll-Pitch-Yaw angles rather than noise on each coefficient of the rotation matrix. Then, it will remain to experiment the method on a real robot, first with predefined robot motions then in a whole visual servoing application.

References

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

[2] K. Daniilidis and E. Bayro-Corrochano. The dual quaternion approach to hand-eye calibration. In *Proc. IAPR International Conference on Pattern Recognition*, pages 318–322, 1996.

[3] B. Espiau. Effect of Camera Calibration Errors on Visual Servoing in Robotics. In *Third International Symposium on Experimental Robotics*, October 1993.

[4] B. Espiau, F. Chaumette, and P. Rives. A New Approach To Visual Servoing in Robotics. *IEEE Trans. on Robotics and Automation*, 8(3), June 1992.

[5] R. Horaud, S. Christy, F. Dornaika, and B. Lamiroy. Object pose: Links between paraperspective and perspective. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 426–433, Cambridge, Mass., June 1995. IEEE Computer Society Press.

[6] R. Horaud, S. Christy, and R. Mohr. Euclidean reconstruction and affine camera calibration using controlled robot motion. In *Proceedings IEEE/RSJ International Conference of Intelligent Robots and Systems*, Grenoble, France, September 1997.

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

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

[9] S. Rémy, M. Dhome, J.M. Lavest, and N. Daucher. Hand-eye calibration. Submitted to IROS'97, 1997.

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

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

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