Robot Hand-Eye Calibration using Structure-from-Motion *

Nicolas Andreff Radu Horaud

Bernard Espiau

INRIA Rhône-Alpes and GRAVIR-IMAG

655, av. de l'Europe, 38330 Montbonnot Saint Martin, France

email: firstname.lastname@inrialpes.fr

February 9, 2000

Abstract

The method we propose simplifies the practical procedure for hand-eye calibration. Indeed, no more calibration jig is needed and small calibration motions can be used.

Without calibration jig, camera motions are computed, up to an unknown scale factor, through structure-from-motion algorithms rather than pose estimation.

*This work was supported by the European Community through the Esprit-IV reactive LTR project number 26247 (VIGOR).

The unknown scale factor is then included in a linear formulation, parameterizing rotations with orthogonal matrices, which handles both large and small motions.

The algebraic analysis of the linear formulation determines whether calibration is partial or complete according to the motions nature.

Finally, in-depth experiments are conducted, with comparison to other methods.

1 Introduction

The background of this work is the guidance of a robot by visual servoing [12, 1]. In this framework, a basic issue is to determine the spatial relationship between a camera mounted onto a robot end-effector (Fig. 1) and the end-effector itself. This spatial relationship is a rigid transformation, a rotation and a translation, known as the hand-eye transformation. The determination of this transformation is called hand-eye calibration.

The goal of this paper is to discuss a technique allowing the hand-eye calibration to be performed in the working site. In practice, this requires that:

- No calibration jig will be allowed.

  A calibration jig is a very accurately manufactured 3D object holding targets as visual features. Mobile robots and space applications of robotics are typical examples where a calibration jig cannot be used. During their mission, such robots may nevertheless need to be calibrated again. However, as affordable on-board weight is limited, they

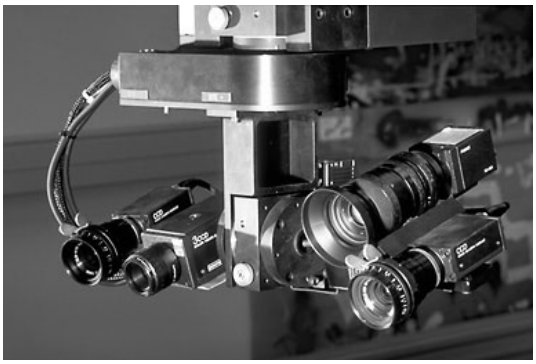

Figure 1: Some cameras mounted on our 5 DOF robot.

can not carry a calibration object and should use their surrounding environment instead. Thus, the availability of a hand-eye self-calibration method is mandatory.

- Special and/or large motions are difficult to achieve and hence should be avoided.

  Indeed, since the hand-eye system must be calibrated on-site, the amount of free robot workspace is limited and the motions have therefore to be of small amplitude. Therefore, the self-calibration method must be able to handle a large variety of motions, including small ones.

Hand-eye calibration was first studied a decade ago [30, 27]. It was shown that any solution to the problem requires to consider both euclidean end-effector motions and camera motions[1]. While the end-effector motions can be obtained from the encoders, the camera motions are to be computed from the images. It was also shown, both algebraically [30] and geometrically [4], that a sufficient condition to the uniqueness of the solution is the existence of two calibration motions with non-parallel rotation axes.

[1]Notice that this requirement may be implicit as in [24].

Several methods were proposed [30, 8, 15, 27, 5, 31] to solve for hand-eye calibration under the assumption that both end-effector and camera motions were known. They differ by the way they represent Euclidean motions, but all have two points in common: (i) rotation is represented by a minimal parameterization and (ii) all proposed methods use pose algorithms to estimate the camera motion relatively to the fixed calibration jig. Pose algorithms require the 3D Euclidean coordinates of the jig targets to be known together with their associated 2D projections onto each image.

Moreover, as the proposed methods use reduced representations of the rotation and since these are ill-defined when rotation angle is small, the calibration motions must be as large as possible: a rule for such a choice of large calibration motions is even given in [30].

Another approach is proposed by Wei *et al.* [32], who perform simultaneously hand-eye calibration and camera calibration without any calibration jig. However, this requires a complex non-linear minimization and the use of a restrictive class of calibration motions. Moreover, no algebraic analysis of the problem is given.

With regard to the existing approaches, we propose a different hand-eye self-calibration method which exploits two main ideas. The first idea is that a specific algebraic treatment is necessary to handle small rotations, since minimal parameterizations of rotations are not defined for small angles and are therefore ill-conditioned. The second idea is that camera motion can be computed from structure-from-motion algorithms rather than from pose algorithms, in order to avoid the use of the calibration jig. Our contributions can be summerized in the following. Firstly, hand-eye calibration is

reformulated in order to take into account the estimation of camera motions from structure-from-motion algorithms. Indeed, camera motions are thus obtained up to an unknown scale factor, which is introduced in the formulation. Secondly, a linear formulation, based on the representation of rotations by orthogonal matrices, is proposed which enables small calibration motions. Thirdly, an algebraic study of this linear solution is performed which shows that partial calibration can nevertheless be performed when the sufficient condition for the uniqueness of the solution is not fulfilled. Fourthly, in-depth experiments are conducted with comparison to other methods.

The remainder of this paper is organized as follows. Section 2 recalls the classical formulation of hand-eye calibration and the structure-from-motion paradigm. Section 3 gives contains the formulation of the linear hand-eye self-calibration method. Section 3.3 contains its algebraic analysis. Finally, Section 4 gives some experimental results and Section 5 concludes this work.

2 Background

In this section, after defining the notation used in this article, we briefly present the classical formulation of hand-eye calibration with a short description of three methods that will be used as references in the experimental section (Section 4). We then describe the estimation of camera motions, concluding in favor of Euclidean reconstruction rather than pose computation.

2.1 Notation

Matrices are represented by upper-case bold-face letters (e.g. $\mathbf{R}$) and vectors by lower-case bold-face letters (e.g. $\mathbf{t}$).

Rigid transformations (or, equivalently, Euclidean motions) are represented with homogeneous matrices of the form:

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

where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. This rigid transformation will be often referred to as the couple $(\mathbf{R}, \mathbf{t})$.

In the linear formulation of the problem, we will use the linear operator *vec* and the tensor product, also known as Kronecker product. The *vec* operator was introduced in [21] 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 Kronecker product [2, 3] is noted $\otimes$. 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:

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

2.2 Hand-eye problem formulation

We present here the classical approach [30, 4, 8, 15, 27, 5, 31] which states that, when the camera undergoes a motion $\mathbf{A} = (\mathbf{R}_a, \mathbf{t}_a)$ and that the cor-

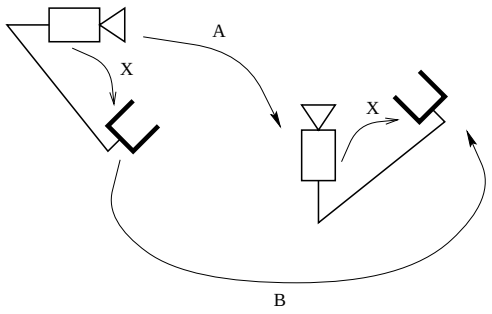

Figure 2: End-effector (represented here by a gripper) and camera motions are conjugated by the hand-eye transformation $\mathbf{X}$.

responding end-effector motion is $\mathbf{B} = (\mathbf{R}_b, \mathbf{t}_b)$, then they are conjugated by the hand-eye transformation $\mathbf{X} = (\mathbf{R}_x, \mathbf{t}_x)$ (Fig. 2). This yields the following homogeneous matrix equation:

$$\mathbf{AX} = \mathbf{XB} \tag{2}$$

where $\mathbf{A}$ is estimated, $\mathbf{B}$ is assumed to be known and $\mathbf{X}$ is the unknown.

Equation (2), applied to each motion i, splits into:

$$\mathbf{R}_{ai}\mathbf{R}_x = \mathbf{R}_x\mathbf{R}_{bi} \tag{3}$$

$$\mathbf{R}_{ai}\mathbf{t}_x + \mathbf{t}_{ai} = \mathbf{R}_x\mathbf{t}_{bi} + \mathbf{t}_x \tag{4}$$

In the method proposed in [30], the first equation is solved by least-square minimization of a linear system obtained by using the axis/angle representation of the rotations. Once $\mathbf{R}_x$ is known, the second equation is also solved with linear least squares techniques.

To avoid this two-stage solution which propagates the error on the rotation estimation onto the translation, a non-linear minimization method based on the representation of the rotations with unit quaternions was proposed

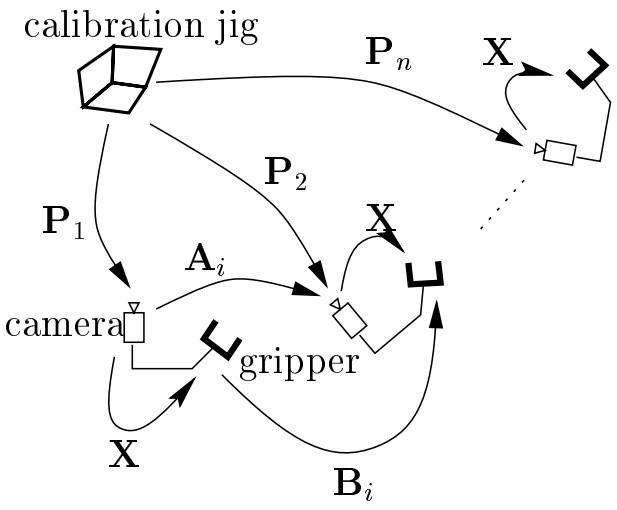

Figure 3: Hand-eye calibration from pose estimation

in [15]. Similarly, a method based on the unit dual quaternion representation of Euclidean motions was developed in [8] to solve simultaneously for hand-eye rotation and hand-eye translation.

2.3 Computing the camera motions

In the prior work [30, 4, 8, 15, 27, 5, 31], camera motions were computed considering images one at a time, as follows. First, 2D-to-3D correspondences were established between the 3D targets on the calibration jig and their 3D projections onto each image i. Then, from the 3D coordinates of the targets, their 2D projections and the intrinsic camera parameters, the pose (i.e. position and orientation) of the camera with respect to the calibration jig is estimated $\mathbf{P}_i = (\mathbf{R}_i, \mathbf{t}_i)$. Finally, the camera motion between image $i-1$ and image i $\mathbf{A}_i = (\mathbf{R}_{ai}, \mathbf{t}_{ai})$ is hence obtained by simple composition (Fig. 3):

$$\mathbf{A}_i = (\mathbf{R}_i \mathbf{R}_{i-1}^T, \mathbf{t}_i - \mathbf{R}_i \mathbf{R}_{i-1}^T \mathbf{t}_{i-1})$$

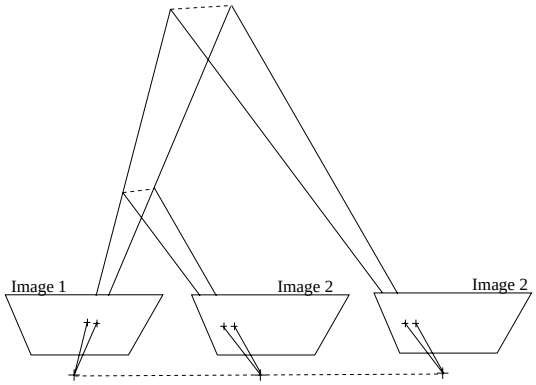

Figure 4: Scale factor ambiguity on the Euclidean reconstruction: which one of the two camera positions on the right and which one of the two sets of 3D points were used to generate the second image ?

Alternatively, one may simultaneously consider all the images that were collected during camera motion. Thus, one may use the multi-frame structure-from-motion paradigm (see [17] for a review). The advantage of structure-from-motion over pose algorithms is that the former does not require any knowledge about the observed 3D object. Indeed, structure-from-motion only relies on 2D-to-2D correspondences. These are more easily obtained since they depend on the image information only. There are two classes of (semi-)automatic methods to find them: a discrete approach, known as *matching* [13], and a continuous approach, known as *tracking* [14].

A relevant class of structure-from-motion methods is known as the Euclidean reconstruction [7, 28, 22, 29, 18, 6]. It assumes that the camera is calibrated (i.e. the camera intrinsic parameters are known). From this knowledge, one can reconstruct the structure of the scene and the motion of the camera up to an unknown scale factor (Fig. 4) using various methods (see

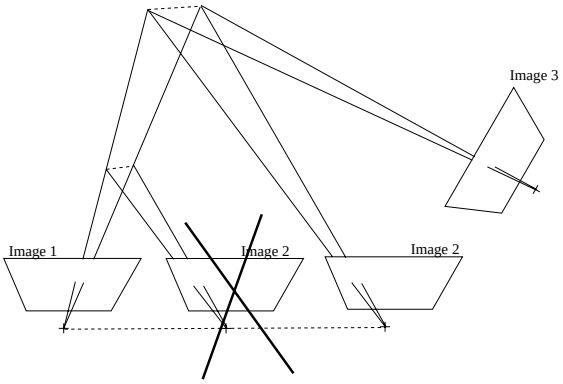

Figure 5: Once the scale factor is resolved between the first two images, the second camera position is uniquely defined with respect to the first one and consequently, the following camera positions are also uniquely defined.

below). This unknown scale factor is the global scale factor of the observed scene and is the same for all the camera motions in the sequence (Fig. 5). Therefore, the estimated camera motions are of the form:

$$\mathbf{A}_i(\lambda) = \begin{pmatrix} \mathbf{R}_{ai} & \lambda \mathbf{u}_{ai} \\ 0\ 0\ 0 & 1 \end{pmatrix} \tag{5}$$

where $\mathbf{R}_{ai}$ is the rotation of the camera between image $i-1$ and image i, λ is the unknown scale factor and $\mathbf{u}_{ai}$ is a vector, parallel to the camera translation $\mathbf{t}_{ai}$ and such that

$$\mathbf{t}_{ai} = \lambda \mathbf{u}_{ai} \tag{6}$$

Taking, without loss of generality, the first motion as a motion with non zero translation allows to arbitrarily choose $\mathbf{u}_{a1}$ as a unit vector. Hence, $\lambda = \|\mathbf{t}_{a1}\|$. Consequently, the $\mathbf{u}_{ai}$'s are related by: $\mathbf{u}_{ai} = \mathbf{t}_{ai}/\|\mathbf{u}_{a1}\|$ and λ can be interpreted as the unknown norm of the first translation.

In summary, camera rotations are completely recovered while camera translations are recovered up to a single unknown scale factor.

In practice, which structure-from-motion algorithm should we choose ? Affine camera models [22, 29, 18] yield simple linear solutions to the Euclidean reconstruction problem, based on matrix factorization. However, affine models are first-order approximations of the perspective model. Hence, only approximations of the Euclidean camera motions can be obtained. Besides, solutions exist [7, 28], based on the perspective model, that offer some Euclidean information on the camera motions, but are non linear. Our choice lies in fact between these two classes of methods: we propose a method for Euclidean reconstruction by successive affine approximations of the perspective model [6], which combines the simplicity of affine methods and the accuracy of non linear methods.

In summary, in order to estimate camera motions, structure-from-motion methods are more flexible than pose computation methods, since no 3D model is needed. The drawback of lowering this constraint is that camera motions are estimated up to an unknown scale factor which we must take into account in the hand-eye self-calibration method.

3 A new linear formulation

In this section, we first modify the formulation of hand-eye calibration in order to take into account the use of Euclidean reconstruction to compute camera motions. Then, we give a solution to this problem which handles small rotations.

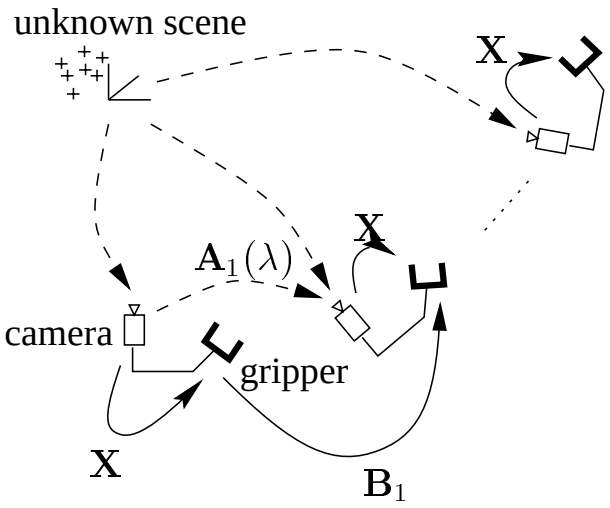

Figure 6: From images of an unknown scene and the knowledge of the intrinsic parameters of the camera, structure-from-motion algorithms estimate, up to an unknown scale factor λ, the camera motions $\mathbf{A}_i(\lambda)$.

3.1 Using structure-from-motion

For using structure-from-motion to estimate camera motions, we have to take into account the unknown scale factor λ. Indeed, the homogeneous equation (2) becomes (compare Fig. 3 and Fig. 6):

$$\mathbf{A}_i(\lambda)\mathbf{X} = \mathbf{X}\mathbf{B}_i \tag{7}$$

where $\mathbf{A}_i(\lambda)$ is the ith estimated camera motion. From (5) and (6), we thus obtain a set of two equations, similar to (3)–(4):

$$\mathbf{R}_{ai}\mathbf{R}_x = \mathbf{R}_x\mathbf{R}_{bi} \tag{8}$$

$$\mathbf{R}_{ai}\mathbf{t}_x + \lambda\mathbf{u}_{ai} = \mathbf{R}_x\mathbf{t}_{bi} + \mathbf{t}_x \tag{9}$$

where the unknowns are now $\mathbf{R}_x$, $\mathbf{t}_x$ and λ.

3.2 Linear formulation

We propose a new formulation which handles rotations of any kind. Its underlying idea is to embed the rotation part of the problem, intrinsically lying in $SO(3)$, in a larger space in order to deliberately free ourselves from the non-linear orthogonality constraint. This allows us to easily find a subspace of matrices verifying (2). Then, the application of the orthogonality constraint selects, in this subspace, the unique rotation which is solution to the problem. This general idea is very powerful here since, as we will see, the non-linear orthogonality constraint reduces to a linear norm constraint.

The new formulation is inspired by the similarity of (8) with the Sylvester equation: $\mathbf{UV} + \mathbf{VW} = \mathbf{T}$. This matrix equation, which often occurs in system theory [3], is usually formulated as a linear system [25, 16, 9]:

$$(\mathbf{U} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{W})vec(\mathbf{V}) = vec(\mathbf{T}) \tag{10}$$

One fundamental property of the Kronecker product is [3]:

$$vec(\mathbf{CDE}) = (\mathbf{C} \otimes \mathbf{E}^T)vec(\mathbf{D}) \tag{11}$$

where $\mathbf{C}$,$\mathbf{D}$,$\mathbf{E}$ are any matrices with adequate dimensions. Applying this relation to equation (8) yields:

$$(\mathbf{R}_{ai} \otimes \mathbf{R}_{bi})vec(\mathbf{R}_x) = vec(\mathbf{R}_x) \tag{12}$$

Introducing the notation $vec(\mathbf{R}_x)$ in equation (9), we obtain:

$$(\mathbf{I}_3 \otimes (\mathbf{t}_{bi}^T))vec(\mathbf{R}_x) + (\mathbf{I}_3 - \mathbf{R}_{ai})\mathbf{t}_x - \lambda \mathbf{u}_{ai} = 0 \tag{13}$$

We can then state the whole problem as a single homogeneous linear system:

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{ai} \otimes \mathbf{R}_{bi} & \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 1} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{bi}^T) & \mathbf{I}_3 - \mathbf{R}_{ai} & -\mathbf{u}_{ai} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{9\times 1} \\ \mathbf{0}_{3\times 1} \end{pmatrix} \tag{14}$$

The question is now: "What is the condition for this system to have a unique solution ?" and a subsequent one is: "What occurs when this condition is not fulfilled ?"

3.3 Algebraic analysis

From earlier work on hand-eye calibration [30, 4], we know that two motions with non-parallel rotation axes are sufficient to determine the hand-eye transformation. We will show in this section, that our new linear solution owns the same sufficient condition but also allows us to identify what can be obtained when such a sufficient condition is not fulfilled (the so-called *partial calibration*).

Hence, let us determine what can be obtained using various combinations of end-effector motions by successively considering: pure translations, pure rotations, planar motions (i.e. containing the same rotational axis and independent translations) and finally general motions. The results of this study are gathered up in Table 1. Notice that by inverting the roles of the end-effector and the camera, we obtain the same results for the recovery of the *eye-hand* transformation (i.e. the inverse of the hand-eye transformation).

Motion 1 Motion 2	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 $\mathbf{R}_B = \mathbf{I}$ $\mathbf{t}_B \neq 0$	$\mathbf{R}_x, \lambda$	$\mathbf{R}_x, \lambda$ $\mathbf{t}_x(\alpha)$	$\mathbf{R}_x, \lambda$ $\mathbf{t}_x(\alpha)$
Rotation $\mathbf{R}_B \neq \mathbf{I}$ $\mathbf{t}_B = 0$	$\mathbf{R}_x, \lambda$ $\mathbf{t}_x(\alpha)$	$\mathbf{R}_x, \mathbf{t}_x(\lambda)$ Decoupled solution	$\mathbf{R}_x, \mathbf{t}_x, \lambda$ General solution
General motion $\mathbf{R}_B \neq \mathbf{I}$ $\mathbf{t}_B \neq 0$	$\mathbf{R}_x, \lambda$ $\mathbf{t}_x(\alpha)$	$\mathbf{R}_x, \mathbf{t}_x, \lambda$ General solution	$\mathbf{R}_x, \mathbf{t}_x, \lambda$ General solution

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

3.3.1 Pure translations

Recall from equation (3), that when end-effector motions are pure translations (i.e. $\mathbf{R}_{bi} = \mathbf{I}_3$), then camera motions are pure translations too (i.e. $\mathbf{R}_{ai} = \mathbf{I}_3$). Hence, equation (4) becomes

$$\mathbf{t}_{ai} = \mathbf{R}_x \mathbf{t}_{bi} \tag{15}$$

Consequently, the amplitude of camera motion is the same as the amplitude of end-effector motion, which is not the case when rotations are involved. One can therefore keep control of the camera displacements and guarantee that a small end-effector motion will not generate an unexpected large camera motion. Concerning calibration, we have the following result:

Proposition 1 *Three independent pure translations yield a linear estimation of hand-eye rotation* $\mathbf{R}_x$ *and of the unknown scale factor* λ*. Hand-eye translation can not be observed.*

Proof: In the case of pure translations, the upper part of the system in (14) vanishes and its lower part simplifies into:

$$\left(\mathbf{I}_3 \otimes (\mathbf{t}_{bi}^T)\right) vec(\mathbf{R}_x) = \lambda \mathbf{u}_{ai} \tag{16}$$

This implies that hand-eye translation $\mathbf{t}_x$ can not be estimated. However, the nine coefficients of the hand-eye rotation $\mathbf{R}_x$ can be obtained as we show below. This was also demonstrated in [33] in the particular case where λ is known.

Let us assume temporarily that λ is known. If $\mathbf{t}_{bi} \neq 0$, then $\mathbf{I}_3 \otimes (\mathbf{t}_{bi}^T)$ has rank 3 since

$$\mathbf{I}_3 \otimes (\mathbf{t}_{bi}^T) = \begin{pmatrix} \mathbf{t}_{bi}^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{t}_{bi}^T & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{t}_{bi}^T \end{pmatrix} \tag{17}$$

Consequently, three linearly independent pure translations yield a full rank (9×9) system:

$$\underbrace{\begin{pmatrix} \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T) \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b3}^T) \end{pmatrix}}_{\mathbf{M}} vec(\mathbf{R}_x) = \lambda \begin{pmatrix} \mathbf{u}_{a1} \\ \mathbf{u}_{a2} \\ \mathbf{u}_{a3} \end{pmatrix} \tag{18}$$

of which the solution $\tilde{\mathbf{R}}_x$ is such that

$$vec(\tilde{\mathbf{R}}_x) = \lambda \mathbf{M}^{-1} \begin{pmatrix} \mathbf{u}_{a1} \\ \mathbf{u}_{a2} \\ \mathbf{u}_{a3} \end{pmatrix} \tag{19}$$

Since $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ [2], it is easy to verify that the analytic form of the inverse of $\mathbf{M}$ is:

$$\mathbf{M}^{-1} = \frac{1}{\Delta} \begin{pmatrix} \mathbf{I}_3 \otimes (\mathbf{t}_{b2} \times \mathbf{t}_{b3}) & \mathbf{I}_3 \otimes (\mathbf{t}_{b3} \times \mathbf{t}_{b1}) & \mathbf{I}_3 \otimes (\mathbf{t}_{b1} \times \mathbf{t}_{b2}) \end{pmatrix} \tag{20}$$

where $\times$ denotes the cross-product and $\Delta = \det(\mathbf{t}_{b1}, \mathbf{t}_{b2}, \mathbf{t}_{b3})$. This allows the rewriting of (19) in closed form:

$$vec(\tilde{\mathbf{R}}_x) = \frac{\lambda}{\Delta} (\mathbf{I}_3 \otimes (\mathbf{t}_{b2} \times \mathbf{t}_{b3})\mathbf{u}_{a1} + \mathbf{I}_3 \otimes (\mathbf{t}_{b3} \times \mathbf{t}_{b1})\mathbf{u}_{a2} + \mathbf{I}_3 \otimes (\mathbf{t}_{b1} \times \mathbf{t}_{b2})\mathbf{u}_{a3}) \tag{21}$$

Applying (11) yields

$$vec(\tilde{\mathbf{R}}_x) = \frac{\lambda}{\Delta} vec\left(\mathbf{u}_{a1}(\mathbf{t}_{b2} \times \mathbf{t}_{b3})^T + \mathbf{u}_{a2}(\mathbf{t}_{b3} \times \mathbf{t}_{b1})^T + \mathbf{u}_{a3}(\mathbf{t}_{b1} \times \mathbf{t}_{b2})^T\right) \tag{22}$$

and from the linearity of the *vec* operator, we finally obtain:

$$\tilde{\mathbf{R}}_x = \frac{\lambda}{\Delta}\left(\mathbf{u}_{a1}(\mathbf{t}_{b2} \times \mathbf{t}_{b3})^T + \mathbf{u}_{a2}(\mathbf{t}_{b3} \times \mathbf{t}_{b1})^T + \mathbf{u}_{a3}(\mathbf{t}_{b1} \times \mathbf{t}_{b2})^T\right) \tag{23}$$

Let us analyze this result and prove now that $\tilde{\mathbf{R}}_x$ is equal to $\mathbf{R}_x$, when measurements are exact. To do that, first recall that $\mathbf{R}_x\mathbf{t}_{bi} = \lambda\mathbf{u}_{ai}$. Hence,

$$\tilde{\mathbf{R}}_x = \frac{1}{\Delta}\mathbf{R}_x \underbrace{\left(\mathbf{t}_{b1}(\mathbf{t}_{b2} \times \mathbf{t}_{b3})^T + \mathbf{t}_{b2}(\mathbf{t}_{b3} \times \mathbf{t}_{b1})^T + \mathbf{t}_{b3}(\mathbf{t}_{b1} \times \mathbf{t}_{b2})^T\right)}_{\mathbf{N}} \tag{24}$$

Recalling that $\mathbf{R}_x$ is orthogonal and verifying that $\mathbf{N} = \Delta\mathbf{I}_3$, we obtain that $\tilde{\mathbf{R}}_x = \mathbf{R}_x$.

This analysis proves that even if λ is unknown the column of $\tilde{\mathbf{R}}_x$, estimated from (23), are orthogonal to each other. Thus, only the unity constraint (i.e. $\det(\tilde{\mathbf{R}}_x) = 1$) remains to be verified by $\tilde{\mathbf{R}}_x$. From (23) again, the unity constraint immediately gives λ. Consequently, the hand-eye rotation can be recovered from three linearly independent translations. □

Proposition 2 *A minimum of 2 linearly independent pure translations are intrinsically enough to estimate the hand-eye rotation* $\mathbf{R}_x$ *and the unknown scale factor* λ.

Proof: The solution is not linear any more and comes in two steps.

1. Scale factor estimation

As $\mathbf{R}_x$ is orthogonal, it preserves the norm. Hence, for each pure translation i, we have:

$$\|\mathbf{R}_x \mathbf{t}_{bi}\| = \|\mathbf{t}_{bi}\|$$

Applying (15) and (6) on the left-hand side of this expression gives for all i:

$$\lambda \|\mathbf{u}_{a\,i}\| = \|\mathbf{t}_{bi}\|$$

where $\mathbf{u}_{a\,i}$ and $\mathbf{t}_{bi}$ are known.

2. Hand-eye rotation estimation

Remark that if $\mathbf{t}_{b1}$ and $\mathbf{t}_{b2}$ are two linearly independent vectors, then $\mathbf{t}_{b1} \times \mathbf{t}_{b2}$ is linearly independent from them. Moreover, one can prove that

$$\mathbf{R}_x(\mathbf{t}_{b1} \times \mathbf{t}_{b2}) = (\mathbf{R}_x \mathbf{t}_{b1}) \times (\mathbf{R}_x \mathbf{t}_{b2})$$

Therefore, we can form the following full-rank (9×9) system:

$$\begin{pmatrix} \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T) \\ \mathbf{I}_3 \otimes ((\mathbf{t}_{b1} \times \mathbf{t}_{b2})^T) \end{pmatrix} vec(\mathbf{R}_x) = \lambda \begin{pmatrix} \mathbf{u}_{a\,1} \\ \mathbf{u}_{a\,2} \\ \lambda(\mathbf{u}_{a\,1} \times \mathbf{u}_{a\,2}) \end{pmatrix} \tag{25}$$

Since λ is now known, $\mathbf{R}_x$ can be obtained by inverting this system and the orthogonality of the solution is guaranteed by the proof of Proposition 1.

□

3.3.2 Pure rotations

By "pure rotations", we mean motions of the end-effector such that $\mathbf{t}_{bi} = 0$. In practice, these motions can be realized by most of the robotic arms, since the latter are usually built in such a manner that their end-effector reference frame is centered on a wrist (i.e. the intersection of the last three revolute joint axes). For similar reasons, pan-tilt systems may also benefit from the subsequent analysis.

In such a case, we can state the following proposition

Proposition 3 *If the robot end-effector undergoes at least two pure rotations with non-parallel axes, then one can linearly estimate the hand-eye rotation* $\mathbf{R}_x$ *and the hand-eye translation up to the unknown scale factor* $\mathbf{t}_x/\lambda$. *These two estimations are decoupled.*

Notice that, in the case where camera motion is obtained through pose computation, λ is known and the hand-eye translation can thus be fully recovered, as does Li [19].

Proof: With pure rotations, the system in (14) is block-diagonal and decouples into:

$$(\mathbf{I}_9 - \mathbf{R}_{ai} \otimes \mathbf{R}_{bi})\, vec(\mathbf{R}_x) = \mathbf{0}_{9\times 1} \quad (26)$$

$$(\mathbf{I}_3 - \mathbf{R}_{ai})\, \mathbf{t}_x = \lambda \mathbf{u}_{ai} \quad (27)$$

With at least two rotations with non parallel axes, we form a system with equations similar to (27) which has then full rank and yields a 1-dimensional solution subspace:

$$\mathbf{t}_x = \lambda \mathbf{t}_{x0} \quad (28)$$

where $\mathbf{t}_{x0}$ is solution to the system:

$$(\mathbf{I}_3 - \mathbf{R}_{a\,i})\,\mathbf{t}_x = \mathbf{u}_{a\,i}, i = 1..n$$

Notice that the parameter of the subspace is the unknown scale factor. This is not surprising since pure rotations of the robot do not contain metric information.

Let us now study the first subsystem (26). 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 (9×9) matrix of (26) $\mathbf{I}_9 - \mathbf{R}_{a\,i} \otimes \mathbf{R}_{b\,i}$ has rank 6. Hence, the solution $\mathbf{R}_x$ lies in a 3-dimensional manifold. Using the two orthogonality constraints, the solution manifold dimension can only be reduced to 1, which confirms the need for two rotations.

In the case of two or more independent rotations, we can state the following lemma (see the proof in Appendix A):

Lemma 1 *If the robot end-effector undergoes at least 2 pure rotations of non parallel axes, then system (26) has rank 8, its null space $\mathcal{K}$ is 1-dimensional and the hand-eye rotation $\mathbf{R}_x$ is equal to:*

$$\mathbf{R}_x = \frac{sign(\det(\mathbf{V}))}{|\det(\mathbf{V})|^{\frac{1}{3}}}\mathbf{V} \tag{29}$$

where sign() returns the sign of its argument, $\mathbf{V} = vec^{-1}(\mathbf{v})$ and $\mathbf{v}$ is any vector of the null space $\mathcal{K}$.

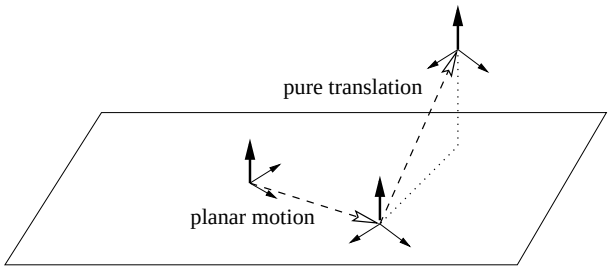

Figure 7: One planar motion with non-identity rotation and one non-zero pure translation which is not parallel to the rotation axis of the planar motion.

which completes the proof of Proposition 3. □

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

3.3.3 Planar motions

Some robots are restricted to move on a plane, such as car-like robots. In this case, all the robot and camera rotations have the same axis $\mathbf{n}_b$ (resp. $\mathbf{n}_a = \mathbf{R}_x\mathbf{n}_b$), which is orthogonal to the plane of motion. Then, we can demonstrate that

Lemma 2 *One planar motion with non-identity rotation and one non-zero pure translation (which is not parallel to the rotation axis of the planar motion, see Fig. 7) are intrinsically enough to recover the hand-eye rotation* $\mathbf{R}_x$ *and the unknown scale factor* λ*. The hand-eye translation can only be estimated up to an unknown height* α *along the normal to the camera plane of motion (Fig. 8).*

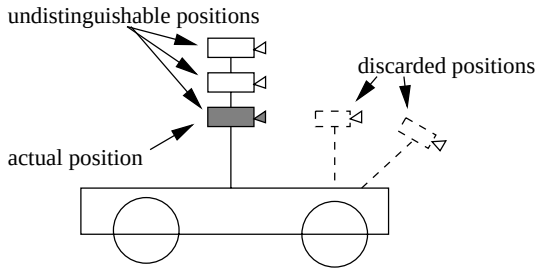

Figure 8: In the case of planar motions, one can not determine the altitude of a camera which is rigidly mounted onto the base.

Notice that this Lemma is not limited to the planar motion case, since the pure translation is not restricted to lie in the plane of motion.

Proof: Assume without loss of generality that the first motion is a pure translation ($\mathbf{R}_{a1} = \mathbf{R}_{b1} = \mathbf{I}_3$, $\mathbf{t}_{b1} \in \Re^3$) and the second is a planar motion with non-identity rotation such that its rotation axis $\mathbf{n}_b$ is not parallel to $\mathbf{t}_{b1}$ (Fig. 7). Then, the general system (14) rewrites as:

$$\begin{pmatrix} \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) & \mathbf{0}_{3\times 3} & -\mathbf{u}_{a1} \\ \mathbf{I}_9 - \mathbf{R}_{a2} \otimes \mathbf{R}_{b2} & \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 1} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T) & \mathbf{I}_3 - \mathbf{R}_{a2} & -\mathbf{u}_{a2} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \\ \lambda \end{pmatrix} = \mathbf{0}_{15\times 1} \qquad (30)$$

which is equivalent to the following two equations

$$\begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{a2} \otimes \mathbf{R}_{b2} & \mathbf{0}_{9\times 1} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) & -\mathbf{u}_{a1} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \lambda \end{pmatrix} = \mathbf{0}_{12\times 1} \qquad (31)$$

$$\begin{aligned} (\mathbf{I}_3 - \mathbf{R}_{a2})\mathbf{t}_x &= -\lambda \mathbf{u}_{a2} \\ &\quad - \left(\mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T)\right) vec(\mathbf{R}_x) \end{aligned} \qquad (32)$$

The solution comes in three steps:

1. Scale factor estimation

As in the proof of Proposition 2.

2. Hand-eye rotation estimation

   Recall that the camera axis of rotation $\mathbf{n}_a$ and the robot axis of rotation $\mathbf{n}_b$ are related by:

$$\mathbf{R}_x \mathbf{n}_b = \mathbf{n}_a$$

   which is similar to (15). Since $\mathbf{t}_{b1}$ and $\mathbf{n}_b$ are assumed to be non-parallel, they are linearly independent. Therefore, we obtain, as in the proof of Proposition 2, a full-rank (9×9) system where $\mathbf{R}_x$ is the only unknown:

$$\begin{pmatrix} \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) \\ \mathbf{I}_3 \otimes (\mathbf{n}_b^T) \\ \mathbf{I}_3 \otimes ((\mathbf{t}_{b1} \times \mathbf{n}_b)^T) \end{pmatrix} vec(\mathbf{R}_x) = \begin{pmatrix} \lambda \mathbf{u}_{a1} \\ \mathbf{n}_a \\ \lambda(\mathbf{u}_{a1} \times \mathbf{n}_a) \end{pmatrix} \tag{33}$$

3. Hand-eye translation estimation

   We can insert the estimated $\mathbf{R}_x$ and λ into (32) and obtain a system, where only $\mathbf{t}_x$ is unknown. This system is always under-constrained. Hence, it admits as solution any vector of the form

$$\mathbf{t}_x(\alpha) = \mathbf{t}_\perp + \alpha \mathbf{n}_a \tag{34}$$

   where α is any scalar value and $\mathbf{t}_\perp$ is a solution in the plane of the camera motion. The latter vector is unique since $\mathbf{I}_3 - \mathbf{R}_{a1}$ has rank 2 and the plane of motion is 2-dimensional. In practice, $\mathbf{t}_\perp$ can be obtained by an SVD of $\mathbf{I}_3 - \mathbf{R}_{a1}$ [23, §2.6].

□

The previous Lemma serves as a basis to the case of planar motions as:

Proposition 4 *Two planar motions allow the estimation of the hand-eye rotation* $\mathbf{R}_x$ *and the unknown scale factor* λ *if one the following three sets of conditions is fulfilled:*

- *the two motions are linearly independent pure translations*
- *one of the two motions is a non-zero pure translation*
- *the two motions contain a non-identity rotation and*

$$(\mathbf{I}_3 - \mathbf{R}_{b2})\mathbf{t}_{b1} - (\mathbf{I}_3 - \mathbf{R}_{b1})\mathbf{t}_{b2} \neq 0$$

In the last two cases, the hand-eye translation can only be estimated up to an unknown height α *along the normal to the camera plane of motion (Fig. 8).*

Proof: The first set of conditions falls back into the pure translation case and Proposition 2 apply. The second set of conditions is contained in Lemma 2.

Let us now show that the last set of conditions can be brought back to the second one. To do that, consider the system which is built upon the two planar motions:

$$\begin{array}{l} L_1 \rightarrow \\ L_2 \rightarrow \\ L_3 \rightarrow \\ L_4 \rightarrow \end{array} \begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_{a1} \otimes \mathbf{R}_{b1} & \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 1} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) & \mathbf{I}_3 - \mathbf{R}_{a1} & -\mathbf{u}_{a1} \\ \mathbf{I}_9 - \mathbf{R}_{a2} \otimes \mathbf{R}_{b2} & \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 1} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T) & \mathbf{I}_3 - \mathbf{R}_{a2} & -\mathbf{u}_{a2} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \\ \lambda \end{pmatrix} = \mathbf{0}_{15\times 1} \qquad (35)$$

The block line L_1 and the third one L_3 of this system are equivalent since both motions have the same rotation axis. Hence, we can discard the first one and obtain:

$$\begin{array}{l} L_1' \rightarrow \\ L_2' \rightarrow \\ L_3' \rightarrow \end{array} \begin{pmatrix} \mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T) & \mathbf{I}_3 - \mathbf{R}_{a1} & -\mathbf{u}_{a1} \\ \mathbf{I}_9 - \mathbf{R}_{a2} \otimes \mathbf{R}_{b2} & \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 1} \\ \mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T) & \mathbf{I}_3 - \mathbf{R}_{a2} & -\mathbf{u}_{a2} \end{pmatrix} \begin{pmatrix} vec(\mathbf{R}_x) \\ \mathbf{t}_x \\ \lambda \end{pmatrix} = \mathbf{0}_{15\times 1} \qquad (36)$$

Consider now the linear combination $(\mathbf{I}_3 - \mathbf{R}_{a2})L_1' - (\mathbf{I}_3 - \mathbf{R}_{a1})L_3'$ which gives:

$$\big[(\mathbf{I}_3 - \mathbf{R}_{a2})\big(\mathbf{I}_3 \otimes (\mathbf{t}_{b1}^T)\big) - (\mathbf{I}_3 - \mathbf{R}_{a1})\big(\mathbf{I}_3 \otimes (\mathbf{t}_{b2}^T)\big)\big] vec(\mathbf{R}_x) \qquad (37)$$

$$+\big[(\mathbf{I}_3 - \mathbf{R}_{a2})(\mathbf{I}_3 - \mathbf{R}_{a1}) - (\mathbf{I}_3 - \mathbf{R}_{a1})(\mathbf{I}_3 - \mathbf{R}_{a2})\big]\mathbf{t}_x \qquad (38)$$

$$-\lambda\big[(\mathbf{I}_3 - \mathbf{R}_{a2})\mathbf{u}_{a1} - (\mathbf{I}_3 - \mathbf{R}_{a1})\mathbf{u}_{a2}\big] \;=\; 0 \qquad (39)$$

As $\mathbf{R}_{a1}$ and $\mathbf{R}_{a2}$ have the same rotation axis, they commute and, hence, $(\mathbf{I}_3 - \mathbf{R}_{a2})(\mathbf{I}_3 - \mathbf{R}_{a1}) - (\mathbf{I}_3 - \mathbf{R}_{a1})(\mathbf{I}_3 - \mathbf{R}_{a2}) = 0$. Therefore, the term on line (38) is null. As for the term on line (39), let us denote it as $\mathbf{u}_{a1}'$.

Let us now consider the first term (37) and show that it can be rewritten under the form $\mathbf{R}_x \mathbf{t}_{b1}^{'T}$. To do that, recall that $(\mathbf{I}_3 \otimes \mathbf{t}_{bi}^T) vec(\mathbf{R}_x) = \mathbf{R}_x \mathbf{t}_{bi}$. Hence, the first term equals:

$$(\mathbf{I}_3 - \mathbf{R}_{a2})\mathbf{R}_x\mathbf{t}_{b1} - (\mathbf{I}_3 - \mathbf{R}_{a1})\mathbf{R}_x\mathbf{t}_{b2}$$

Using $\mathbf{R}_{ai}\mathbf{R}_x = \mathbf{R}_x\mathbf{R}_{bi}$, we then obtain:

$$\mathbf{R}_x\big(\underbrace{(\mathbf{I}_3 - \mathbf{R}_{b2})\mathbf{t}_{b1} - (\mathbf{I}_3 - \mathbf{R}_{b1})\mathbf{t}_{b2}}_{\mathbf{t}_{b1}'}\big)$$

Consequently, L_1' is equivalent to:

$$\mathbf{R}_x\mathbf{t}_{b1}' = \lambda\mathbf{u}_{a1}'$$

where we recognize the pure translation case. Hence, system (35) rewrites under the same form as in (30) of Lemma 2. Therefore, a solution exists if the virtual robot pure translation $\mathbf{t}_{b_1}'$ is not parallel to $\mathbf{n}_b$. As both $\mathbf{t}_{b1}$ and $\mathbf{t}_{b2}$ are orthogonal to $\mathbf{n}_b$, this condition reduces to a non zero condition on $\mathbf{t}_{b_1}'$, which is expressed as:

$$(\mathbf{I}_3 - \mathbf{R}_{b2})\mathbf{t}_{b1} - (\mathbf{I}_3 - \mathbf{R}_{b1})\mathbf{t}_{b2} \neq 0$$

□

In conclusion, we exhibited sufficient conditions to obtain, from two planar motions, the hand-eye rotation and the hand-eye translation, up to a component perpendicular to the camera plane of motion. In the case of a car, this unknown component can be interpreted as a height with respect to the base of the car (Fig. 8).

3.3.4 The general case

In the case of two independent general motions with non-parallel axes, there exists a unique solution to the hand-eye calibration problem. We obtain the same result for our hand-eye self-calibration problem:

Proposition 5 *If the robot end-effector undergoes two independent general motions with non-parallel axes, then the hand-eye transformation* $(\mathbf{R}_x, \mathbf{t}_x)$ *can be fully recovered, as well as the Euclidean reconstruction unknown scale factor* λ.

Using our formulation, one possibility to solve the whole system in (14) is to find its null space, which is a subspace of $\Re^{13}$. The latter subspace must

be 1-dimensional and only depend on λ, according to the sufficient condition for hand-eye calibration. Hence, the solution to hand-eye self-calibration is a 13×1 vector to be found in a 1-dimensional subspace. It can therefore be extracted from this null space by applying the unity constraint to the first 9 coefficients representing the hand-eye rotation, as seen in the pure translation case.

However, Wei *et al* [32] remarked, in the case where camera motions are obtained through pose computation, that the accuracy of the simultaneous estimation of hand-eye rotation and translation is not independent of the physical unit used for the translation. By analogy with this remark, solving directly for the whole system may yield the same dependence. In addition, such a solution does not guarantee that the estimated $\mathbf{R}_x$ is an orthogonal matrix. Then, one has to perform a correction of the result by applying the orthogonality constraint. However, this correction is non-linear in essence and it is hence improbable to find the corresponding correction on the hand-eye translation estimation.

On the opposite, a two-step solution, as in [30], guarantees an orthogonal estimate of the hand-eye rotation. Indeed, the first step consists of the linear estimation of the hand-eye rotation as in the case of pure rotations (26), which had this property:

$$\begin{pmatrix} \mathbf{I}_3 - \mathbf{R}_{a1} \otimes \mathbf{R}_{b1} \\ \mathbf{I}_3 - \mathbf{R}_{a2} \otimes \mathbf{R}_{b2} \end{pmatrix} vec(\mathbf{R}_x) = 0 \tag{40}$$

As for the second step, it exploits the remaining lines in (14):

$$\begin{pmatrix} \mathbf{I}_3 - \mathbf{R}_{a1} & -\mathbf{u}_{a1} \\ \mathbf{I}_3 - \mathbf{R}_{a2} & -\mathbf{u}_{a2} \end{pmatrix} \begin{pmatrix} \mathbf{t}_x \\ \lambda \end{pmatrix} = \begin{pmatrix} -\mathbf{R}_x \mathbf{t}_{b1} \\ -\mathbf{R}_x \mathbf{t}_{b2} \end{pmatrix} \tag{41}$$

We thus have a unique linear solution to the hand-eye translation and the scale factor.

4 Experiments

In this section, we will first choose a distance to measure the errors between rigid transformations since their group $SE(3)$ does not hold an intrinsic metric [20]. Second, we will show some simulation results to test the robustness to noise of our method, compared to the reference methods. Finally, we will give experimental results in real conditions. Notice that more experimental results can be found in [1].

In this section, we numbered the methods we compared as follows: axis/angle method [30] (**M1**), dual quaternion method [8] (**M2**), non-linear minimization [15] (**M3**), our linear formulation adapted to the case where camera motions are obtained through pose computation (**M4**), and self-calibration (**M5**).

4.1 Error measurement

To measure the errors in translation, we chose the usual relative error in $\Re^3$: $\|\hat{\mathbf{t}} - \mathbf{t}\|/\|\mathbf{t}\|$, where the '^' notation represents the estimated value.

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 $\|\hat{\mathsf{q}} - \mathsf{q}\| = 2 - 2\cos\frac{\alpha}{2}$. It is thus strictly increasing from 0 to 4 as α goes from 0 to 2π. Moreover, this metric avoids the singularity in $\alpha = \pi$ appearing when using geodesics [26, p.35].

4.2 Simulations

We first performed simulations to gain some insight of the numerical behavior of our linear method (**M4**) with comparison to the reference methods (**M1**–**M3**). We thus tested the robustness of the methods to noise and their accuracy with respect to the number of calibration motions in use.

4.2.1 Simulation procedure

For each simulation series and for each value of the parameter of interest (noise, number of motions), we followed the same methodology. First, we defined a hand-eye transformation by random choice of the Roll-Pitch-Yaw angles of its rotation matrix as well as of the coefficients of its translation vector, according to Gaussian laws. Second, we similarly chose a sequence of robot motions and defined, from it and the hand-eye transformation, the corresponding camera motion sequence. Third, we added noise to the camera motions (see below). Finally, we performed hand-eye calibration with the various methods and compared their results to the initial hand-eye transformation.

4.2.2 Inserting noise

We added noise to the camera translations $\mathbf{t}_{A_i}$ by defining $\tilde{\mathbf{t}}_{A_i} = \mathbf{t}_{A_i} + \nu \|\mathbf{t}_{A_i}\| \mathbf{n}$ where ν is a scalar and $\mathbf{n}$ is a Gaussian 3-vector with zero mean and unit variance (white noise). As for the camera rotations, we added noise to their Roll-Pitch-Yaw angles as $\tilde{\alpha} = (1 + \nu r)\alpha$ where α is any of these angles, ν is the same as for the translation and r is a 1-dimensional white-noise. Hence, ν defines a signal-to-noise ratio.

4.2.3 Robustness to noise

We tested for the value of ν, making it vary from 0 to 20% in two simulation series. In the first one, we made 100 different choices of hand-eye transformations and motion sequences for each noise level. These sequences contained only two motions, with maximal amplitude of 1 m in translation and 180 deg in rotation. Fig. 9 gathers the calibration errors. It shows that Tsai and Lenz's method (**M1**) and ours (**M4**) obtain the highest accuracy in rotation. For translations, they are very powerful as long as the noise level is low but are less accurate than the dual quaternion method (**M2**) or the non linear minimization method (**M3**) when the noise level increases.

In a second simulation series, we almost repeated the first one, just reducing the amplitude of the calibration motions to 2 cm in translation and 10 deg in rotation. The results (Fig. 10) show that our linear formulation is less sensitive to this reduction than the other methods.

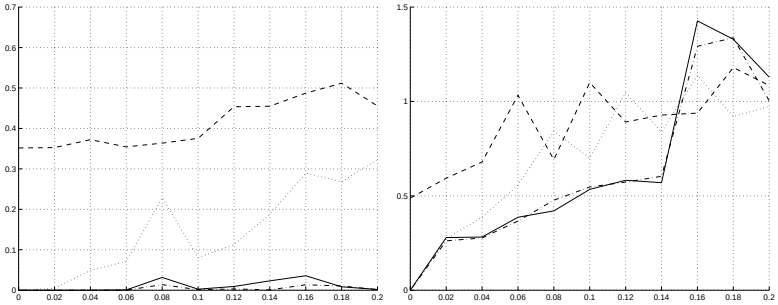

Figure 9: Rotation (left) and translation (right) relative calibration errors with respect to noise level: **M1** (—), **M2** ($\cdots$), **M3** (- -), **M4** (– ·).

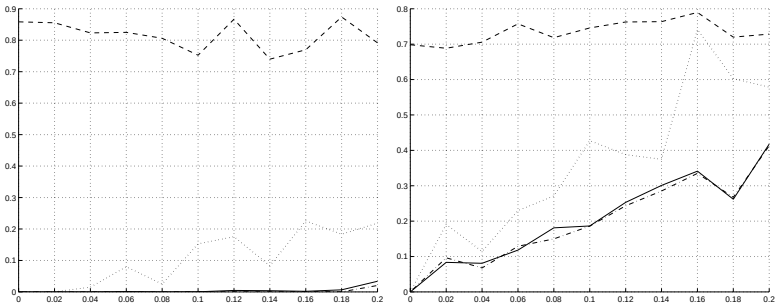

Figure 10: Calibration errors with respect to noise level using small motions (Same conventions as in Fig. 9)

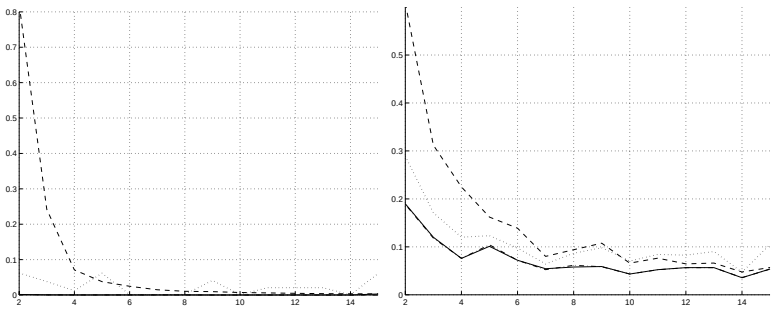

Figure 11: Calibration errors with respect to the number of calibration motions using small motions (Same conventions as in Fig. 9)

4.2.4 Influence of motion number

In this experiment, we kept the noise level constant ($\nu = 0.01$) and generated sequences of varying length, i.e. from 2 to 15 calibration motions. Their amplitude was chosen to be small (1 cm in translation and 10 deg in rotation). For each sequence length, we proceeded to 100 random choices of hand-eye transformations and calibration motions. The results (Fig. 11) show here again a higher accuracy for our linear formulation.

4.3 Experiments on real data

When dealing with real data, no ground-truth value is available for comparison. Therefore, we compared, for each motion i, $\mathbf{A}_i\mathbf{X}$ and $\mathbf{X}\mathbf{B}_i$. We then gathered all these errors into RMS errors.

4.3.1 Experiment 1

To evaluate the correctness of the solution obtained by hand-eye self-calibration, we had to compare it with those obtained by classical calibration methods

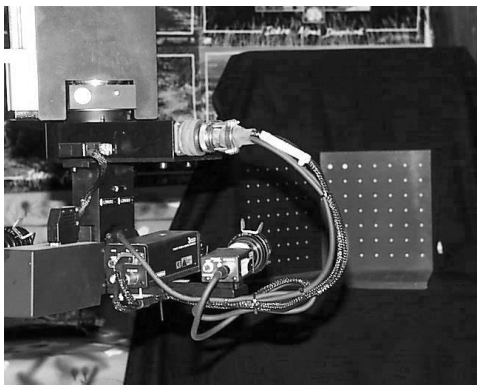

Figure 12: In Experiment 1, the camera observes a calibration grid.

with the same data.

Hence, we took images of our calibration grid (Fig. 12) and performed hand-eye calibration with the axis/angle method [30] (**M1**), the dual quaternion method [8] (**M2**), the non-linear minimization [15] (**M3**) and the linear formulation (**M4**). Finally, using the same points, extracted from the images of the calibration grid, but not their 3D model, we applied the hand-eye self-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 Fig. 13. These positions were chosen as far as possible from each other according to the advice given in [30]. It can be seen that (**M4**) gives the smallest error in rotation due to the numerical 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 and can be explained by an approximative estimation of the intrinsic parameters.

From this long sequence, we used a RANSAC-like method to compute a

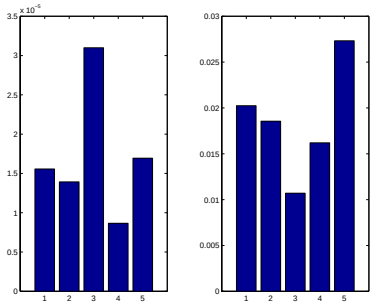

Figure 13: RMS errors in rotation (left) and translation (right) with 33 images of a calibration grid for each method (see text).

Method	Rotation error	Translation error
M1	$1.10.10^{-5}$	0.018
M2	$1.61.10^{-5}$	0.096
M3	$9.77.10^{-5}$	0.149
M4	$0.06.10^{-5}$	0.023
M5	$1.99.10^{-5}$	0.322

Table 2: Comparison with a robust estimation of the hand-eye transformation

robust estimation of the hand-eye transformation (see [1]). Then, we compared the results obtained above to this robust estimation. We gather the errors in Table 2. It confirms that the linear method is numerically very efficient, especially as far as rotation is concerned. Moreover, the self-calibration method yields a lower accuracy, which nevertheless remains acceptable in the context of visual servoing [11].

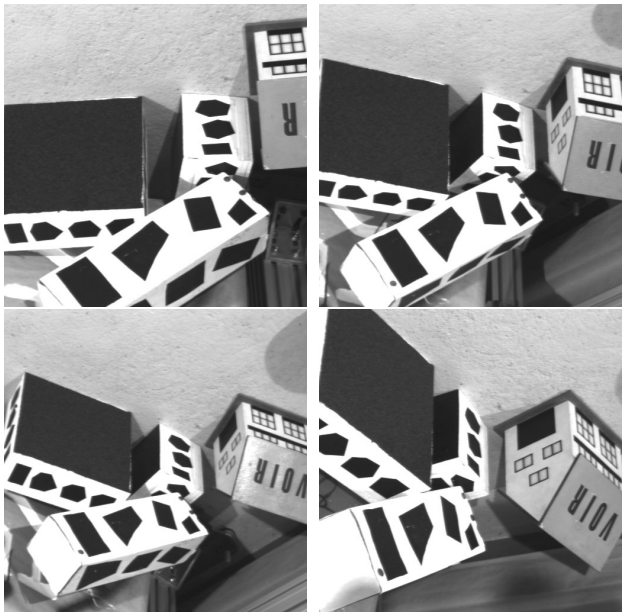

Figure 14: A sequence of 4 images used for hand-eye self-calibration in Experiment 2.

4.3.2 Experiment 2

In a second experiment, we tested (**M5**) with more realistic images. Four positions were defined where the images shown in Fig. 14 were taken. In the first image, points were extracted and then tracked during the motion between each position of the camera. Then, hand-eye self-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 Fig. 15. They show an awful behavior of the non linear minimization method, probably due to the small

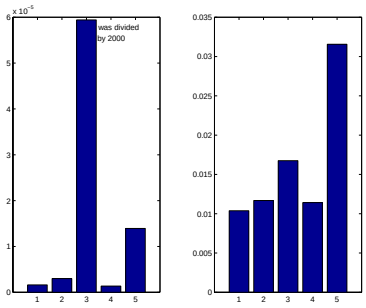

Figure 15: RMS errors in rotation (left) and translation (right) with 4 images (see text).

number of data. They also show a slightly higher degradation of the performance of (**M5**) compared to the others. Nevertheless, it remains in an acceptable ratio since the relative error in translation is close to 3%.

To balance the lack of ground-truth, we also compared the results obtained in this experiment to the robust estimation described in Experiment 1 (Table 3). This comparison confirms the accuracy of both the linear method and the self-calibration scheme.

Method	Rotation error	Translation error
M1	$2.7.10^{-5}$	0.18
M2	$2.8.10^{-5}$	0.22
M3	1.82	1.01
M4	$2.3.10^{-5}$	0.17
M5	$2.8.10^{-4}$	0.20

Table 3: Comparison with a robust estimation of the hand-eye transformation

5 Conclusion

We proposed a hand-eye self-calibration method which reduces the human supervision compared with classical calibration methods. The cost of releasing the human constraint is a small degradation of the numerical accuracy. However, the obtained precision is good enough in the context of visual servoing.

This method is based on the structure-from-motion paradigm, rather than pose estimation, to compute the camera motions and its derivation includes a new linear formulation of hand-eye calibration. The linearity of the formulation allows a simple algebraic analysis. Thus, 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. Moreover, the linear formulation provides improved numerical accuracy even in the case where the camera/robot rotations have small amplitude.

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 Euclidean motions of a stereo pair [10]. 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 hand-eye (self-)calibration that yield the higher numerical accuracy ?"

A Proof of Lemma 1

A.1 Preliminary results

Preliminary result 1 *Given two similar rotation matrices* $\mathbf{R}$ *and* $\mathbf{R}'$ *(i.e. there exists a rotation matrix* $\mathbf{R}_x$ *such that* $\mathbf{R}' = \mathbf{R}_x\mathbf{R}{\mathbf{R}_x}^T$*) then*

1) if $\mathbf{v}$ *is an eigenvector of* $\mathbf{R} \otimes \mathbf{R}'$*, then* $(\mathbf{I} \otimes {\mathbf{R}_x}^T)\mathbf{v}$ *is an eigenvector of* $\mathbf{R} \otimes \mathbf{R}$ *for the same eigenvalue;*

2) if $\mathbf{x}$ *is an eigenvector of* $\mathbf{R} \otimes \mathbf{R}$*, then* $(\mathbf{I} \otimes \mathbf{R}_x)\mathbf{x}$ *is an eigenvector of* $\mathbf{R} \otimes \mathbf{R}'$ *for the same eigenvalue.*

Proof: 1) Let $\mathbf{v}$ be an eigenvector of $\mathbf{R} \otimes \mathbf{R}'$ with eigenvalue λ. Then, $(\mathbf{R} \otimes \mathbf{R}')\mathbf{v} = \lambda\mathbf{v}$. Replacing $\mathbf{R}'$ by $\mathbf{R}_x\mathbf{R}{\mathbf{R}_x}^T$ in this relation gives:

$$(\mathbf{R} \otimes \mathbf{R}_x\mathbf{R}{\mathbf{R}_x}^T)\mathbf{v} = \lambda\mathbf{v}$$

From $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$[2], we obtain:

$$(\mathbf{I} \otimes \mathbf{R}_x)(\mathbf{R} \otimes \mathbf{R})(\mathbf{I} \otimes {\mathbf{R}_x}^T)\mathbf{v} = \lambda\mathbf{v}$$

As $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$[2], we derive the following relation:

$$(\mathbf{I} \otimes {\mathbf{R}_x}^T)^{-1}(\mathbf{R} \otimes \mathbf{R})(\mathbf{I} \otimes {\mathbf{R}_x}^T)\mathbf{v} = \lambda\mathbf{v}$$

Hence, $(\mathbf{R} \otimes \mathbf{R})(\mathbf{I} \otimes {\mathbf{R}_x}^T)\mathbf{v} = \lambda(\mathbf{I} \otimes {\mathbf{R}_x}^T)\mathbf{v}$.

2) Let $\mathbf{x}$ be an eigenvector of $\mathbf{R} \otimes \mathbf{R}$ with eigenvalue α. Then,

$$(\mathbf{R} \otimes \mathbf{R})\mathbf{x} = \alpha\mathbf{x}$$

As $(\mathbf{I} \otimes \mathbf{R}_x{}^T)(\mathbf{I} \otimes \mathbf{R}_x) = \mathbf{I}$, we can insert it on both sides:

$$(\mathbf{R} \otimes \mathbf{R})(\mathbf{I} \otimes \mathbf{R}_x{}^T)(\mathbf{I} \otimes \mathbf{R}_x)\mathbf{x} = \alpha(\mathbf{I} \otimes \mathbf{R}_x{}^T)(\mathbf{I} \otimes \mathbf{R}_x)\mathbf{x}$$

which rewrites as:

$$(\mathbf{I} \otimes \mathbf{R}_x)(\mathbf{R} \otimes \mathbf{R})(\mathbf{I} \otimes \mathbf{R}_x{}^T)(\mathbf{I} \otimes \mathbf{R}_x)\mathbf{x} = \alpha(\mathbf{I} \otimes \mathbf{R}_x)\mathbf{x}$$

Hence,

$$(\mathbf{R} \otimes \mathbf{R}')(\mathbf{I} \otimes \mathbf{R}_x)\mathbf{x} = \alpha(\mathbf{I} \otimes \mathbf{R}_x)\mathbf{x}.$$

□

Preliminary result 2 *Let* $\mathbf{R}_1$ *and* $\mathbf{R}_2$ *be 2 rotation matrices with non parallel axes. Let* $\mathbf{R}$ *be another rotation matrix. Then,*

$$\left.\begin{array}{l} \mathbf{R}_1 \otimes \mathbf{R}_1 \; vec(\mathbf{R}) = vec(\mathbf{R}) \\ \mathbf{R}_2 \otimes \mathbf{R}_2 \; vec(\mathbf{R}) = vec(\mathbf{R}) \end{array}\right\} \Rightarrow \mathbf{R} = \mathbf{I}_3$$

Proof: The previous system is equivalent to

$$\mathbf{R}_1\mathbf{R} = \mathbf{R}\mathbf{R}_1$$
$$\mathbf{R}_2\mathbf{R} = \mathbf{R}\mathbf{R}_2$$

If $\mathbf{R}$ satisfies the first equation, then either $\mathbf{R}$ is the identity or it has the same rotation axis as $\mathbf{R}_1$. Similarly, it is either the identity or has the same rotation axis as $\mathbf{R}_2$. As $\mathbf{R}_1$ and $\mathbf{R}_2$ have different rotation axes, it must be the identity. □

Preliminary result 3 *Let* $\mathbf{R}_1$ *and* $\mathbf{R}_2$ *be two rotation matrices with non parallel rotation axes. Let* $\mathbf{M} \neq 0$ *be a matrix such that*

$$\mathbf{R}_1 \otimes \mathbf{R}_1 \; vec(\mathbf{M}) = vec(\mathbf{M})$$
$$\mathbf{R}_2 \otimes \mathbf{R}_2 \; vec(\mathbf{M}) = vec(\mathbf{M})$$

Then,

$$\exists \lambda \neq 0, \mathbf{M} = \lambda \mathbf{I}_3$$

Proof: To write

$$\mathbf{R}_1 \otimes \mathbf{R}_1 \; vec(\mathbf{M}) = vec(\mathbf{M})$$

is equivalent to say that $\mathbf{R}_1$ and $\mathbf{M}$ commute. Therefore, $\mathbf{M}$ is of the form $\lambda\mathbf{R}$ where $\lambda \neq 0$ and $\mathbf{R}$ is a rotation matrix which commutes with $\mathbf{R}_1$. This can be easily seen by replacing $\mathbf{M}$ by its SVD.

Thus, $\mathbf{M} = \lambda\mathbf{R}$ where $\mathbf{R}$ is such that:

$$\mathbf{R}_1\mathbf{R} = \mathbf{R}\mathbf{R}_1$$
$$\mathbf{R}_2\mathbf{R} = \mathbf{R}\mathbf{R}_2$$

From Preliminary result 2, we obtain $\mathbf{R} = \mathbf{I}_3$ and $\mathbf{M} = \lambda\mathbf{I}_3$. □

A.2 Proof of Lemma 1

System (26) is equivalent to

$$\mathbf{R}_{A_1} \otimes \mathbf{R}_{B_1}\mathbf{v} = \mathbf{v}$$
$$\mathbf{R}_{A_2} \otimes \mathbf{R}_{B_2}\mathbf{v} = \mathbf{v}$$

Under the assumption that the camera motions and the robot motions are rigidly linked by a constant hand-eye transformation $(\mathbf{R}_x, \mathbf{t}_x)$ and from Preliminary result 1, this system becomes:

$$\begin{aligned} \mathbf{R}_{B_1} \otimes \mathbf{R}_{B_1}\mathbf{v}' &= \mathbf{v}' \\ \mathbf{R}_{B_2} \otimes \mathbf{R}_{B_2}\mathbf{v}' &= \mathbf{v}' \end{aligned}$$

where $\mathbf{v}' = (\mathbf{I} \otimes {\mathbf{R}_x}^T)\mathbf{v}$. Applying the result of Preliminary result 3, we obtain that $vec^{-1}(\mathbf{v}') = \lambda \mathbf{I}_3$. Using the definition of $\mathbf{v}'$ and the properties of the Kronecker product, we end up in:

$$vec^{-1}(\mathbf{v}){\mathbf{R}_x}^T = \lambda \mathbf{I}_3$$

where $\mathbf{V} = vec^{-1}(\mathbf{v})$. Hence,

$$\mathbf{V} = \lambda \mathbf{R}_x$$

Consequently, the matrix $\mathbf{V}$ extracted from the null space of (26) is proportional to the hand-eye rotation. The coefficient λ is obtained from the orthogonality constraint: $det(\mathbf{R}_x) = 1$. The latter becomes $det(\mathbf{V}) = \lambda^3$ which finally gives:

$$\lambda = sgn(det(\mathbf{V}))\ |det(\mathbf{V})|^{1/3}$$

□

References

[1] N. Andreff. *Asservissement visuel à partir de droites et auto-étalonnage pince-caméra.* Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, November 1999.

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

[3] J. W. Brewer. Kronecker products and matrix calculus in system theory. *"IEEE Trans. on Circuits and Systems"*, CAS-25(9):772–781, 1978.

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

[5] 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.

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

[7] N. Cui, J.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 *"Proc. IAPR International Conference on Pattern Recognition"*, pages 318–322, 1996.

[9] A. S. Deif, N. P. Seif, and S. A. Hussein. Sylvester's equation : accuracy and computational stability. *Journal of Computational and Applied Mathematics*, 61:1–11, 1995.

[10] F. Devernay and O. Faugeras. From projective to euclidean reconstruction. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 264–269, June 1996.

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

[12] 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.

[13] A.W. Gruen. Adaptative least squares correlation: a powerful image matching technique. *S. Afr. Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3):175–187, 1985.

[14] G. Hager and K. Toyama. X vision: A portable substrate for real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, 1998.

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

[16] D. Y. Hu and L. Reichel. Krylov-subspace methods for the Sylvester equation. *Linear Algebra and its Applications*, 172:283–313, 1992.

[17] J. Inigo Thomas and J. Oliensis. Dealing with noise in multiframe structure from motion. *Computer Vision and Image Understanding*, 76(2):109–124, November 1999.

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

[19] M. Li. Kinematic calibration of an active head-eye system. *IEEE Trans. on Robotics and Automation*, 14(1):153–158, February 1998.

[20] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

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

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

[23] 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, 1992.

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

[25] F. Rotella and P. Borne. Explicit solution of Sylvester and Lyapunov equations. *Mathematics and Computers in Simulation*, 31:271–281, 1989.

[26] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control : The Task Function Approach* . Clarendon Press, Oxford, 1991.

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

[28] C.T. Taylor, D.J. Kriegman, and P. Anandan. Structure and motion in two dimensions from multiple images: A least squares approach. In *Proc. IEEE Workshop on Visual Motion*, pages 242–248. IEEE Computer Society Press, October 1991.

[29] C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. In *Proc. IEEE Workshop on Visual Motion*, pages 21–28. IEEE Computer Society Press, October 1991.

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

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

[32] G-Q Wei, K Arbter, and G Hirzinger. Active self-calibration of robotic eyes and hand-eye relationships with model identification. *"IEEE Trans. on Robotics and Automation"*, 14(1):158–165, 1998.

[33] H. Zhuang. A note on using only position equations for robotic hand/eye calibration. *"IEEE Trans. on Systems, Man and Cybernetics"*, 28(3):426–427, June 1998.