Robust Hand–Eye Calibration of an Endoscopic Surgery Robot Using Dual Quaternions

Jochen Schmidt, Florian Vogt*, and Heinrich Niemann

Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg
Martensstr. 3, 91058 Erlangen, Germany
{jschmidt,vogt,niemann}@informatik.uni-erlangen.de

Abstract. This paper presents an approach for applying a dual quaternion hand–eye calibration algorithm on an endoscopic surgery robot. Special focus is on robustness, since the error of position and orientation data provided by the robot can be large depending on the movement actually executed. Another inherent problem to all hand–eye calibration methods is that non–parallel rotation axes must be used; otherwise, the calibration will fail. Thus we propose a method for increasing the numerical stability by selecting an optimal set of relative movements from the recorded sequence. Experimental evaluation shows the error in the estimated transformation when using well–suited and ill–suited data. Additionally, we show how a RANSAC approach can be used for eliminating the erroneous robot data from the selected movements.

1 Introduction

In this paper we present an approach for the practical aspects in terms of robustness of hand–eye calibration using an endoscopic surgery robot. Especially, we address two problems: how to choose the data that is used in the calibration algorithm such that the numerical stability increases, and how to use a RANSAC approach for outlier detection and removal.

A common drawback of all hand–eye calibration algorithms, which is inherent to the problem itself, is that at least two motions are necessary where the rotations have non–parallel rotation axes. Consequently, numerical stability can be increased by selecting the data accordingly. Additionally, outlier detection and removal is essential, since the position and orientation data provided by the robot arm is unreliable when substantial changes in the direction of movement are executed. A RANSAC approach [6] is used for this purpose.

The application area is the reconstruction of high–quality medical light fields [12]. The hand–eye transformation has to be estimated every time when the camera head is mounted anew on the endoscope optics, which is done before each operation because it has to be sterilized. Therefore, an algorithm that works automatically and stably without human interaction is desirable.

A vast amount of literature is available on the topic of hand–eye calibration. The classical way is to solve for rotation first, and then for translation [9, 11]. In [7] an algorithm is proposed that solves for both simultaneously using nonlinear optimization, while Daniilidis [3, 4] is the first who presented a linear algorithm for simultaneous computation of the hand–eye parameters. This was the main reason why we chose this algorithm as a basis for our work.

* This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under grant SFB 603/TP B6. Only the authors are responsible for the content.

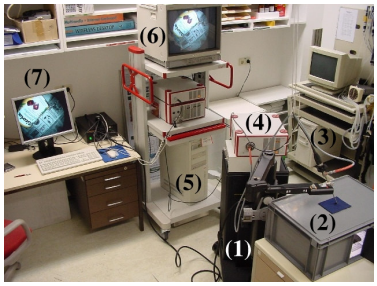

Fig. 1. Left: experimental setup. (1) AESOP 3000, (2) "patient", (3) camera head and endoscope, (4) light source, (5) computer, (6) video–endoscopic system (original image), (7) second monitor (computer image/light field). Right: original image as seen by the camera.

The paper is organized as follows: Sect. 2 describes the AESOP 3000 robot system as well as the method formerly used for estimating the unknown hand–eye transformation between robot plug and camera. In Sect. 3 we give a short introduction to hand–eye calibration methods, with special focus on the dual quaternion approach proposed by Daniilidis. How to make the hand–eye calibration robust enough for practical purposes is described in Sect. 4. Experimental results are given in Sect. 5.

2 The Robot System

We use the *Computer Motion Inc.* AESOP 3000 (cf. Fig. 1, left, no. (1)) endoscopic surgery robot. Images are grabbed directly from the endoscopic camera. The robot arm has seven degrees of freedom (one translational and six rotational), which are provided by the robot before and after each image is taken; the data is averaged for further processing. The complete experimental setup is shown in Fig. 1 (left). A calibration pattern is used to estimate the intrinsic camera parameters [10]. Radial and tangential lens–distortion coefficients are computed in order to undistort the (highly distorted) endoscopic images. Given these seven values, the position and orientation (pose) of the endoscope plug can be computed from the known kinematics, but not the pose of the tip of the endoscope.

Up to now, the unknown transformation from plug to camera was estimated as follows: The distance from plug to endoscope–lens was measured by hand, while the orientation of the optics with respect to the plug was calculated in two steps. Since the camera head is not fixed at the endoscope optics but is mounted anew before each operation, the rotation between head and optics had to be computed. This was done by detecting a notch at the optics border (cf. Fig. 1, right). Usually a 30° optics is used, i. e. the angle had to be taken into account when computing the final transformation. Then the relative movement between two images using a calibration pattern was computed and the plug–angle was optimized such that the relative movement calculated by the kinematics equaled the real one. This method has some drawbacks: First of all, measuring by hand is arduous and inaccurate. Also, notch detection requires using additional low-level image processing methods instead of data already available and is only possible if an optics is used that actually has a notch, which is not the case for all endoscope optics. These drawbacks are eliminated by using a robust hand–eye calibration method as described in the following sections.

3 Hand–Eye Calibration

3.1 Overview

Given rigid displacements between the movements of a robot arm and the movements of a camera mounted on that arm, the unknown rigid transformation between arm and camera has to be computed, which is the same for all arm/camera movement pairs. This is known as hand–eye calibration. These circumstances are shown in the following commutative diagram; robot arm poses are denoted by $\boldsymbol{A}$, camera poses by $\boldsymbol{C}$ at two time steps i and k. The unknown hand–eye transformation is denoted by $\boldsymbol{R}_{\mathrm{HE}}$ and $\boldsymbol{t}_{\mathrm{HE}}$.

$$\begin{array}{ccc} \boldsymbol{A}_k & \xrightarrow{\boldsymbol{R}_{\mathrm{HE}},\boldsymbol{t}_{\mathrm{HE}}} & \boldsymbol{C}_k \\ {\scriptstyle \boldsymbol{R}_{\mathrm{A}ik},\boldsymbol{t}_{\mathrm{A}ik}}\uparrow & & \uparrow{\scriptstyle \boldsymbol{R}_{\mathrm{C}ik},\boldsymbol{t}_{\mathrm{C}ik}} \\ \boldsymbol{A}_i & \xrightarrow{\boldsymbol{R}_{\mathrm{HE}},\boldsymbol{t}_{\mathrm{HE}}} & \boldsymbol{C}_i \end{array} \tag{1}$$

The hand–eye parameters $\boldsymbol{R}_{\mathrm{HE}}$ and $\boldsymbol{t}_{\mathrm{HE}}$ can be recovered from the following equation induced by the commutativity of diagram (1):

$$\begin{pmatrix} \boldsymbol{R}_{\mathrm{HE}} & \boldsymbol{t}_{\mathrm{HE}} \\ \boldsymbol{0}_3^{\mathrm{T}} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_{\mathrm{A}ik} & \boldsymbol{t}_{\mathrm{A}ik} \\ \boldsymbol{0}_3^{\mathrm{T}} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_{\mathrm{C}ik} & \boldsymbol{t}_{\mathrm{C}ik} \\ \boldsymbol{0}_3^{\mathrm{T}} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_{\mathrm{HE}} & \boldsymbol{t}_{\mathrm{HE}} \\ \boldsymbol{0}_3^{\mathrm{T}} & 1 \end{pmatrix} \tag{2}$$

which can be decomposed into two separate equations:

$$\boldsymbol{R}_{\mathrm{HE}}\boldsymbol{R}_{\mathrm{A}ik} = \boldsymbol{R}_{\mathrm{C}ik}\boldsymbol{R}_{\mathrm{HE}} \tag{3}$$

$$(\boldsymbol{I}_3 - \boldsymbol{R}_{\mathrm{C}ik})\boldsymbol{t}_{\mathrm{HE}} = \boldsymbol{t}_{\mathrm{C}ik} - \boldsymbol{R}_{\mathrm{HE}}\boldsymbol{t}_{\mathrm{A}ik} \tag{4}$$

These are the well–known hand–eye equations, which were first published in [9, 11]. Numerous solutions for solving (3) and (4) have been proposed, e. g., [9, 11, 7, 3]. The classical way is to first solve (3) for $\boldsymbol{R}_{\mathrm{HE}}$, and then (4) for $\boldsymbol{t}_{\mathrm{HE}}$. In [7] an algorithm is proposed that solves for $\boldsymbol{R}_{\mathrm{HE}}$ and $\boldsymbol{t}_{\mathrm{HE}}$ simultaneously using nonlinear optimization, while Daniilidis [3, 4] is the first who presented a linear algorithm for simultaneous computation of the hand–eye parameters. Besides the well–founded theory and good performance, this was the main reason why we chose the dual quaternion algorithm for our application. Additionally, using this algorithm it is possible to show how numerical stability of hand–eye calibration in general is increased by selecting optimal relative movement pairs as described in Sect. 4.1. Therefore, we will now give a summary of the dual quaternion algorithm.

3.2 Dual Quaternions: A Unified Representation of Rotation and Translation

Quaternions Quaternions are a commonly used representation for rotations in 3–D, hence we will not go into much detail here; for details see, e. g., [5, 8].

A quaternion $\boldsymbol{h}$ is defined as $\boldsymbol{h} = w + x\mathrm{i} + y\mathrm{j} + z\mathrm{k}$ with $w, x, y, z \in \mathbb{R}$, where w is the real part and x, y, z are the imaginary parts. For the imaginary units, the following equation holds: $\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1$. Often a quaternion is written as a 4–D vector $\boldsymbol{h} = (w, x, y, z)$ or $\boldsymbol{h} = (w, \boldsymbol{v})$, where $\boldsymbol{v}$ is a 3–vector containing the imaginary parts. Just as the multiplication of two unit complex numbers defines a rotation in 2–D, a multiplication of two unit quaternions yields a rotation in 3–D. Let $\boldsymbol{p}$ be a 3–D point

to be rotated, $\boldsymbol{a}$ a rotation axis with $|\boldsymbol{a}| = 1$, and θ the angle of rotation around this axis. Define the following two quaternions:

$$\boldsymbol{h} = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2} \cdot \boldsymbol{a}\right), \quad \boldsymbol{p}' = (0, \boldsymbol{p}) \quad . \tag{5}$$

Then

$$\boldsymbol{p}'_{\text{rot}} = \boldsymbol{h}\boldsymbol{p}'\overline{\boldsymbol{h}} \quad , \tag{6}$$

where $\boldsymbol{p}'_{\text{rot}}$ is the rotated point and $\overline{\boldsymbol{h}}$ is the conjugate of $\boldsymbol{h}$.

Dual Quaternions As quaternions are a representation for 3–D rotations, dual quaternions treat rotations *and* translations in a unified way.
Dual Numbers Dual numbers were proposed by Clifford in the 19$^{\text{th}}$ century [2]. They are defined by $\hat{z} = a + \varepsilon b$, where $\varepsilon^2 = 0$. When using vectors for a and b instead of real numbers, the result is a dual vector.
Dual Quaternions A dual quaternion $\hat{\boldsymbol{h}}$ is defined as a quaternion, where the real and imaginary parts are dual numbers instead of real ones, or equivalently as a dual vector where the dual and the non–dual part are quaternions: $\hat{\boldsymbol{h}} = \boldsymbol{h} + \varepsilon\boldsymbol{h}'$. Just as unit quaternions represent rotations, unit dual quaternions contain rotation and translation [3]. In the dual quaternion representation of $\boldsymbol{R}$ and $\boldsymbol{t}$, the non–dual part $\boldsymbol{h}$ is defined as in (5), and the dual part as

$$\boldsymbol{h}' = \left(-\frac{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{a}}{2}\sin\frac{\theta}{2}, \frac{1}{2}\left(\boldsymbol{t} \times (\sin\frac{\theta}{2} \cdot \boldsymbol{a}) + \cos\frac{\theta}{2} \cdot \boldsymbol{t}\right)\right) \quad . \tag{7}$$

Using the dual quaternion representation, the hand–eye calibration formulas (3) and (4) and can be written in a concise way, very similar to (6):

$$\hat{\boldsymbol{h}}_c = \hat{\boldsymbol{h}}\hat{\boldsymbol{h}}_r\overline{\hat{\boldsymbol{h}}} \quad , \tag{8}$$

where $\hat{\boldsymbol{h}}_r$ encodes the movement of the robot arm and $\hat{\boldsymbol{h}}_c$ the movement of the camera.

3.3 Algorithm

This section gives an overview over a linear algorithm for hand–eye calibration using dual quaternions presented by Daniilidis in [3, 4]. Starting from (8) he derives a linear system of equations which has to be solved for $\boldsymbol{h}$ and $\boldsymbol{h}'$:

$$\boldsymbol{A}\begin{pmatrix}\boldsymbol{h}\\ \boldsymbol{h}'\end{pmatrix} = \boldsymbol{0} \quad . \tag{9}$$

Since each movement pair $(\hat{\boldsymbol{h}}_r, \hat{\boldsymbol{h}}_c)$ results in 6 equations, for n motions $\boldsymbol{A}$ is a $6n \times 8$ matrix having rank 6.

Note that at least two motions of the robot arm/camera *with different rotation axes* are necessary for reconstructing the rigid hand–eye transformation. This is a general result [1, 11], i. e. it is not specific to the dual quaternion algorithm.

Solving (9) using Singular Value Decomposition (SVD) with $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$ results in two zero singular values, or nearly zero singular values in the case of noisy data. The two–dimensional solution space is spanned by the column vectors $\boldsymbol{v}_7$ and $\boldsymbol{v}_8$ of $\boldsymbol{V}$ that

correspond to the zero singular values, i. e. $\left(\boldsymbol{h}\ \boldsymbol{h}'\right)^{\mathrm{T}} = \lambda_1 \boldsymbol{v}_7 + \lambda_2 \boldsymbol{v}_8$. The two remaining unknowns λ_1 and λ_2 can be computed by using the additional constraint that $\hat{\boldsymbol{h}}$ is a unit dual quaternion. The recovery of the actual hand–eye transformation, i. e. $\boldsymbol{R}_{\mathrm{HE}}$ and $\boldsymbol{t}_{\mathrm{HE}}$, from $\hat{\boldsymbol{h}}$ is easy: The rotation matrix can be computed directly from the non–dual part $\boldsymbol{h}$ of $\hat{\boldsymbol{h}}$ (cf. e. g., [8]). The translation vector is given by $\boldsymbol{t}_{\mathrm{HE}} = 2\boldsymbol{h}'\overline{\boldsymbol{h}}$ (cf. [3]).

4 Hand-Eye Calibration of an Endoscopic Surgery Robot

In this section we are going to describe how the numerical stability of hand–eye calibration can be increased by selecting robot/camera movement pairs in an optimal way. Additionally, elimination of outliers using a RANSAC approach is presented.

4.1 Selection of Movement Pairs for Increased Numerical Stability

For reconstructing the rigid transformation from robot arm to camera using hand–eye calibration, at least two motions with *different* rotation axes are necessary (cf. Sect. 3.3).

As probably in most applications, in endoscopic surgery, robot arm movements are usually continuous, which means that translation and rotation of neighboring frames are similar and the rotation axes are not very different. Hence it is usually suboptimal to process the arm/camera positions in their temporal order. It is much better to select the data such that relative movements are used for calibration that actually fulfill the requirement above. As an optimality criterion we propose to use the scalar product between the rotation axes of two camera movements. Let $\boldsymbol{a}_{ij}$ and $\boldsymbol{a}_{kl}$ be the normalized rotation axes of two relative movements from frame i to j and from k to l, respectively. Then

$$s_{ij,kl} = |\boldsymbol{a}_{ij}^{\mathrm{T}} \boldsymbol{a}_{kl}| \tag{10}$$

gives a value of one for parallel rotation axes and zero for orthogonal axes, where the latter are the ones that are suited best for hand–eye calibration. Note that camera and not robot arm data should be used at this point, since the camera was calibrated accurately using a calibration pattern, while the data provided by the robot is still corrupted by outliers and hence unreliable. Also, it is important for practical purposes, that the rotation axes of relative movements are well–defined. For small rotations, the axis changes its direction considerably, and two axes may be almost orthogonal even if the movement pair is ill–suited for hand–eye calibration. Therefore, we recommend a pre–selection of those relative movements, where the rotation matrix differs from identity. In our implementation, we use the rotation angle θ for pre–selection (cf. Sect. 5). An additional benefit of this step is that the amount of data for the following pairwise rating decreases and thus computation time as well.

Selection of the best pairs increases the numerical stability of the hand–eye calibration, which can be easily seen if we examine at the condition number of matrix $\boldsymbol{A}$ in (9). This matrix is of rank six if non–parallel axes are used, and of rank five if parallel axes are used. Now consider the ratio of the largest singular–value of $\boldsymbol{A}$ and the sixth singular–value: The system is ill–conditioned in the case of parallel axes (the condition number becomes infinite in the worst case), and the condition gets better (i. e. the sixth singular value is much greater than zero) if the axes are non-parallel.

If the goal is an optimal data set consisting of m movement pairs, the problem can be formulated as follows: Let $\mathcal{C}$ denote the set of all movement pairs possible; find a subset $\mathcal{M}$ of $\mathcal{C}$ such that the following criterion is minimized:

$$s_{\mathcal{M}} = \sum_{(ij,kl)\in\mathcal{M}} s_{ij,kl} \quad , \tag{11}$$

where $s_{\mathcal{M}}$ denotes the rating of $\mathcal{M}$. Better pairs in the sense of low absolute ratings $s_{\mathcal{M}}$ lead to lower condition numbers and thus higher numerical stability. In general, this means that $s_{ij,kl}$ has to be computed for all possible movement pairs (but of course not necessarily be stored) in order to get the optimal subset $\mathcal{M}$. Our experiments showed that this is actually possible if the number of frames is not too high; e. g., even for 100 frames and a pre–selection as described above, computation of the optimal subset takes only a few seconds on a state–of–the art PC. For n frames, the algorithm is of complexity $O(n^4)$: The total number of all relative movements is $n(n+1)/2$; if m movements are left after pre–selection, the total number of pairs is $m(m+1)/2$. For the worst case, i. e. $m = n(n+1)/2$, this results in $m(m+1)/2 = (n^4 + 2n^3 + 3n^2 + 2n)/8 = O(n^4)$.

4.2 Eliminating Outliers

The AESOP 3000 robot provides pose information, which is usually accurate (cf. Fig. 2). Nevertheless, experiments showed that in some cases, especially when the direction of movement changes substantially, pose information is very unreliable (note the peaks in Fig. 2). This is only a local problem, since the experiments also showed that the pose–error does not sum up during the movement of the arm. Thus it is necessary to detect the positions where those changes occur, so that they can be removed from the data used for hand–eye calibration. Possible methods for outlier elimination are:

- Remove the positions where the changes in direction of translation of the robot arm are very high.
- Follow an iterative approach and use all data in the first iteration; for the second iteration remove those positions that have very high errors when comparing the hand–eye transformed robot poses with the data from the calibrated endoscope.
- Apply a RANSAC [6] approach; use the same error measure as in the item above.

Since the most promising approach is the last one, we are going to describe now how to apply RANSAC for outlier removal:

1. Choose m random samples from the movement pairs selected as described in Sect. 4.1, where each sample consists of $e = 2$ movements, the minimum number required for hand–eye calibration.
2. For each sample i: Compute $\boldsymbol{R}_{\mathrm{HE}i}$, $\boldsymbol{t}_{\mathrm{HE}i}$.
3. Apply the hand–eye transformation to the robot arm poses and compute the error between the transformed arm poses and the calibrated camera poses for each relative movement. Determine the number of consistent pairs.
4. Keep the largest set of consistent pairs.
5. After all samples are evaluated: Re–compute $\boldsymbol{R}_{\mathrm{HE}i}$ and $\boldsymbol{t}_{\mathrm{HE}i}$ using *all* consistent pairs of the largest set.

Table 1. Mean error per frame for old method and hand–eye calibration, once with best movement pairs, once for pairs in temporal order. For the Euler angles, the error is given in degrees, for the rotation in norm of the difference quaternion, and for the translation in mm.

Method/Sequence	Euler x	Euler y	Euler z	Quaternion	Translation
ALF1, old method	0.289	0.279	0.246	0.00477	0.675
ALF1, hand–eye, best	0.219	0.386	0.245	0.00495	0.897
ALF1, hand–eye, temporal order	0.941	0.729	0.675	0.0135	10.7
ALF2, old method	0.259	0.264	0.352	0.00495	0.910
ALF2, hand–eye, best	0.218	0.272	0.288	0.00433	1.15
ALF2, hand–eye, temporal order	0.502	0.834	1.50	0.0161	10.85

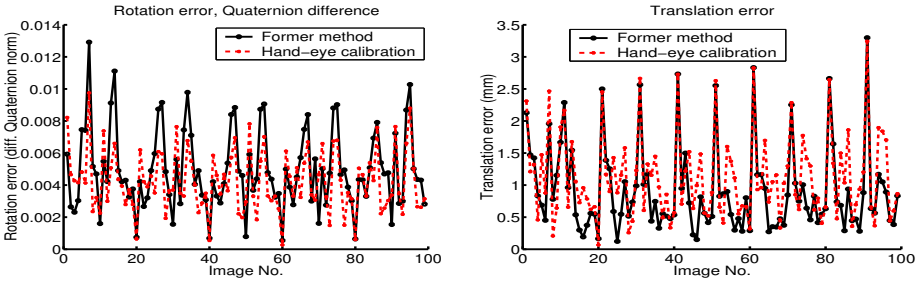

Fig. 2. Sequence *ALF2*. Left: Error in rotation measured in norm of the difference quaternion. Right: Error in translation measured in norm of the difference vector.

The probability P that in at least one sample *all* e elements are inliers is given by $P = 1-(1-(1-\epsilon)^e)^m$, where ϵ is the estimated outlier–rate. If this equation is solved for m we get the minimum number of samples that should be chosen. For $P = 0.99$, $e = 2$, and an estimated outlier–rate of $\epsilon = 20\%$ we get $m \approx 4.51$, i. e. at least 5 samples should be used.

5 Experiments

For evaluation of accuracy and robustness of the dual quaternion hand–eye calibration algorithm with the extensions for movement pair selection and outlier elimination, we present here the results on endoscopic image sequences with different movement paths. Instead of a real patient, a box with a hole for the endoscope that is inlaid with newspaper and printed OP–images of the abdomen was used. An example image (without calibration pattern) is shown in Fig. 1 (right).

Table 1 shows the errors for the two sequences *ALF1* (55 frames) and *ALF2* (100 frames) for the former method used, for the best movement pairs using the hand–eye method, and for the hand–eye algorithm where the relative movements were used in temporal order. The error per frame was computed between the actual endoscope poses (from the calibration pattern) and the poses computed by applying the hand-eye transformation to the robot arm data. Comparison of the errors of the best pairs and pairs in temporal order shows impressively the influence of pair selection: When using the temporal order, the hand–eye algorithm fails; when using the best pairs, the error is comparable to the former method described in Sect. 2, but without its drawbacks, in particular completely automatically. Before the selection of pairs rated using (10), we pre–selected those relative movements, where the rotation angle θ was between 10°

and 170° for sequence *ALF1*, and between 15° and 165° for *ALF2*. For *ALF1* we used 30% (absolute: 36531) of all possible pairs left after pre–selection, for *ALF2* 10% (absolute: 1720). Figure 2 depicts plots of the relative error in frame–to–frame movement for the *ALF2* sequence in rotation (measured in norm of the difference quaternion) and translation for the former non–automatic method described in Sect. 2 and for the robust hand–eye calibration method. Noticeable are the distinct peaks: These are exactly the frames where the robot position data is very erroneous, which is the case if the movement direction changes considerably. Remember also, that no nonlinear refinement was used yet, which would result in even better performance.

6 Conclusion

We presented an approach for selecting the relative robot/camera movements such that hand–eye calibration can be performed in a numerically stable way. Outlier removal is very important in our application as well, which is the use of an endoscopic surgery robot, since the robot position data is unreliable when movements such as substantial direction changes are executed. We showed how to use RANSAC to accomplish this goal. Although we applied a dual quaternion hand–eye calibration algorithm, these problems are not specific to it, but inherent to the hand–eye calibration problem itself. We showed experimentally the benefit of movement pair selection compared to the straightforward approach of using relative movements in temporal order.

References

1. H. Chen. A Screw Motion Approach to Uniqueness Analysis of Head–Eye Geometry. In *Proc. of CVPR*, pages 145–151, Maui, Hawaii, June 1991.
2. William Clifford. Preliminary Sketch of Bi–quaternions. *Proceedings of the London Mathematical Society*, 4:381–395, 1873.
3. K. Daniilidis. Hand-Eye Calibration Using Dual Quaternions. *International Journal of Robotics Research*, 18:286–298, 1999.
4. K. Daniilidis. Using the Algebra of Dual Quaternions for Motion Alignment. In G. Sommer, editor, *Geometric Computing with Clifford Algebras*, chapter 20, pages 489–500. Springer-Verlag, 2001.
5. Oliver Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
6. M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–385, 1981.
7. R. Horaud and F. Dornaika. Hand-Eye Calibration. *International Journal of Robotics Research*, 14(3):195–210, 1995.
8. J. Schmidt and H. Niemann. Using Quaternions for Parametrizing 3–D Rotations in Unconstrained Nonlinear Optimization. In *Vision, Modeling, and Visualization 2001*, pages 399–406, Stuttgart, Germany, November 2001. AKA/IOS Press, Berlin, Amsterdam.
9. Y. 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.
10. R. Y. Tsai. A Versatile Camera Calibration Technique for High–Accuracy 3D Machine Vision Metrology Using Off–the–Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, RA–3(4):323–344, August 1987.
11. 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, June 1989.
12. F. Vogt, S. Krüger, D. Paulus, H. Niemann, W. Hohenberger, and C. H. Schick. Endoskopische Lichtfelder mit einem kameraführenden Roboter. In *7. Workshop Bildverarbeitung für die Medizin*, pages 418–422, Erlangen, March 2003. Springer-Verlag.