Data-Selection for Hand-Eye Calibration A Vector Quantization Approach

Jochen Schmidt
Centre for Artificial Intelligence Research
Auckland University of Technology
Auckland, New Zealand

Heinrich Niemann
Lehrstuhl für Mustererkennung
Universität Erlangen-Nürnberg
91058 Erlangen, Germany

Abstract

This paper presents new vector quantization based methods for selecting well-suited data for hand-eye calibration from a given sequence of hand and eye movements. Data selection can improve the accuracy of classic hand-eye calibration, and make it possible in the first place in situations where the standard approach of manually selecting positions is inconvenient or even impossible, especially when using continuously recorded data. A variety of methods is proposed, which differ from each other in the dimensionality of the vector quantization compared to the degrees of freedom of the rotation representation, and how the rotation angle is incorporated. The performance of the proposed vector quantization based data selection methods is evaluated using data obtained from a manually moved optical tracking system (hand) and an endoscopic camera (eye).

1 Introduction

Hand-eye calibration, which has classically been used for calibrating the rigid transformation from the tip of a robot manipulator arm to a camera mounted on the arm, becomes more interesting for applications where similar problems arise, but which are not directly related to robotics. Examples include hand data provided by an optical tracking system instead of a robot, where the camera is mounted on an endoscope and moved manually as in Vogt (2006) or Schmidt et al. (2004), an inertial sensor mounted on a camera as in Aron et al. (2004), or even self-calibration of a rigid stereo camera system, where one camera can be treated as the "hand", and the other as the "eye" as in Luong and Faugeras (2001) and Schmidt (2006).

A problem that is common to all hand-eye calibration algorithms is that the quality of the result is highly dependent on the data used for computing the unknown transformation. The usual approach for solving this problem is to use robot movements that already take the restrictions on the data into account, which means that the movement has to be planned before recording. Suggestions how this can be achieved were already given in one of the original publications on hand-eye calibration by Tsai and Lenz (1989). In situations where planning such a well-suited movement is not possible (e. g., due to constraints on the available space) or cannot be controlled well (e. g., when using a hand-held camera), methods for data selection are required in order to get high-quality calibration results. In particular in situations where the camera records images at frame rates of 25 images per second while the camera is moving continuously, using the movements in temporal order is a bad choice.

Therefore, performing a data selection step before the actual hand-eye calibration is essential in these cases. We present a variety of new methods for automatic selection of well-suited data based on vector quantization. Using any of them before the actual hand-eye calibration makes calibration of continuously recorded data possible in the first place. And, as we will show in the experiments section, even when only a small number of planned poses are used as in the classic approach to acquiring data, running the proposed data selection algorithms can lead to an increase in accuracy. The algorithms presented here have been developed by Schmidt (2006). They have been partially revised, particularly making the automatic threshold computation (Sect. 3.5) more effective, and new experimental results will be presented in this article.

The problem of automatic data selection was already addressed in Schmidt et al. (2003), where it is applied in hand-eye calibration of an endoscopic surgery robot. This method first removes relative movements[1] with

[1] a relative movement is defined by the rotation and translation be-

small rotation angles using a fixed threshold. After this pre-selection step, *pairs* of relative movements are rated according to their suitability for hand-eye calibration. The goal is to use the best fraction of pairs for computing the hand-eye transformation. As a rating criterion it is proposed to use the scalar product between the rotation axes of two relative movements. A worst case estimate (if no movements are eliminated during pre-selection w. r. t. angle) of the time complexity of this approach is $O(N^4)$, N being the number of poses contained in the original input sequence. The method is more or less a brute-force approach; it cannot compete in computation time (or accuracy) with the vector quantization methods that we present in this article. Another problem is that always well-matching *pairs* of relative movements are selected, where one relative movement may be contained in multiple pairs. The pairs are afterwards used to form a linear system of equations for solving for the hand-eye transformation. Since each relative movement results in one equation, it may happen that one movement is used more than once, leading to two linearly dependent equations, one of them being redundant.

We have presented the first vector quantization based data selection method in Schmidt et al. (2004). This method is also described in the article at hand (Sect. 3.3). The pre-selection of movements using a fixed threshold has been adopted in Schmidt et al. (2004), while now an algorithm for computing a suitable threshold automatically is used. Schmidt et al. (2004) also contains an experimental comparison with the approach published in Schmidt et al. (2003); it concludes that the latter cannot compete with the vector quantization method at all, the calibration results are mostly worse than for the vector quantization based methods.

Zhang et al. (2005) and Shi et al. (2005) have also presented an iterative method for data selection in hand-eye calibration. Their method considers pairs of movements and rates their suitability for hand-eye calibration by comparing the angle between the two rotation axes, the rotation angles, and the norm of the translation vectors to thresholds. In contrast to that our vector quantization based approach does not only rate pairs, but takes into account all available movements on a global scale.

The remaining paper is structured as follows: A short introduction into hand-eye calibration will be given in Sect. 2, including a brief literature overview and explaining the critical factors that influence calibration accuracy. Section 3 described the new proposed data selection algorithms based on vector quantization. Experimental results, including a comparison to classic hand-eye calibration without data selection, are presented in Sect. 4. The paper concludes with Sect. 5. As the proposed algorithms make excessive use of different representations of 3D rotations, we have included a short introduction for readers not familiar with these representations in Appendix A.

tween two given poses; see also Sect. 3.1.

2 Hand-Eye Calibration

2.1 Overview

The hand-eye calibration problem can be formulated as follows: Given a robot manipulator arm and a camera mounted on that arm, compute the rigid transformation from arm to camera, also called *hand-eye transformation*. Knowledge of this transformation is necessary for computing the camera pose from the pose of the robot arm, which is usually provided by the robot itself, while the pose of the camera is unknown. Once the hand-eye transformation is known, the camera pose is easily computed from the pose information provided by the robot.

The first hand-eye calibration methods were published by Shiu and Ahmad (1989), and Tsai and Lenz (1989); an early comparison of the methods available at that time was given in Wang (1992). The hand-eye calibration problem was formulated by Shiu and Ahmad (1989) as a matrix equation of the form

$$\boldsymbol{T}_{\mathrm{E}}\boldsymbol{T}_{\mathrm{HE}} = \boldsymbol{T}_{\mathrm{HE}}\boldsymbol{T}_{\mathrm{H}} \quad , \tag{1}$$

where $\boldsymbol{T}_{\mathrm{H}}$ is the robot arm (*hand*) movement, $\boldsymbol{T}_{\mathrm{E}}$ the camera (*eye*) movement, and $\boldsymbol{T}_{\mathrm{HE}}$ is the unknown hand-eye transformation, i. e., the transformation from gripper to camera[2]. Each matrix $\boldsymbol{T}_{\chi}$ is a rigid transformation in homogeneous form, i. e.,

$$\boldsymbol{T}_{\chi} = \begin{pmatrix} \boldsymbol{R}_{\chi} & \boldsymbol{t}_{\chi} \\ \boldsymbol{0}_3^{\mathrm{T}} & 1 \end{pmatrix}, \chi \in \{\mathrm{H}, \mathrm{E}, \mathrm{HE}\} \quad . \tag{2}$$

The transformations $\boldsymbol{T}_{\chi}$ consist of a 3×3 rotation matrix $\boldsymbol{R}_{\chi}$ and a 3D translation vector $\boldsymbol{t}_{\chi}$; $\boldsymbol{0}_3$ denotes the 3D null-vector.

The straightforward way for solving (1) is to split it into two separate equations, one that contains only rotation, and a second one that contains rotation and translation, which results in:

$$\boldsymbol{R}_{\mathrm{E}}\boldsymbol{R}_{\mathrm{HE}} = \boldsymbol{R}_{\mathrm{HE}}\boldsymbol{R}_{\mathrm{H}} \quad , \tag{3}$$

$$(\boldsymbol{R}_{\mathrm{E}} - \boldsymbol{I}_{3\times3})\boldsymbol{t}_{\mathrm{HE}} = \boldsymbol{R}_{\mathrm{HE}}\boldsymbol{t}_{\mathrm{H}} - \boldsymbol{t}_{\mathrm{E}} \quad , \tag{4}$$

where $\boldsymbol{I}_{3\times3}$ denotes the 3×3 identity matrix. Thus, the rotational part $\boldsymbol{R}_{\mathrm{HE}}$ of the hand-eye transformation can be determined first from (3), and, after inserting it into the second equation (4), the translational part $\boldsymbol{t}_{\mathrm{HE}}$ is computed. This is how hand-eye calibration is done by

[2] note that in some publications $\boldsymbol{T}_{\mathrm{HE}}$ is the transformation from camera to gripper.

Shiu and Ahmad (1989); Tsai and Lenz (1989); Chou and Kamel (1991); Wang (1992). Various parameterizations of rotation were applied (see Appendix A for an introduction): The original works of Shiu and Ahmad (1989), and Tsai and Lenz (1989) use the axis/angle representation; quaternions were used by Chou and Kamel (1991), and Horaud and Dornaika (1995). Dual quaternions were introduced by Daniilidis (1999, 2001).

In contrast to the former approaches, it was suggested by Chen (1991) that rotation and translation should be solved simultaneously, not separately. This approach is also followed by Horaud and Dornaika (1995), where a non-linear optimization of rotation and translation is performed. An online calibration method based on Kalman-filtering was proposed by Andreff (1997). Daniilidis (1999, 2001) introduced a hand-eye calibration algorithm based on dual quaternions that is also capable of handling rotation and translation simultaneously, but, in contrast to the former approaches, a linear solution is given. A method based on screw motions very similar to Daniilidis' dual quaternion algorithm can be found in Zhao and Liu (2006). Fassi and Legnani (2005) present a geometric interpretation of the original hand-eye calibration equation and analyze its properties.

Commonly, hand-eye calibration methods rely on the fact that the movement of the robot manipulator arm is provided by the robot, while the camera movement is computed using a calibration pattern and classic camera calibration methods, such as Tsai (1987); Zhang (1998, 2000). Andreff et al. (1999, 2001) and Schmidt et al. (2005) presented approaches that obtain the camera movement not by using a calibration pattern, but from point feature tracking and a structure-from-motion technique (see Faugeras and Luong (2001); Hartley and Zisserman (2003) for an introduction). In this case a scale factor has to be estimated additionally, thus making the problem very similar to the self-calibration of a rigid stereo-camera system (Luong and Faugeras, 1993, 2001; Faugeras and Luong, 2001; Dornaika and Chung, 2003; Schmidt, 2006).

2.2 Critical Factors

Regardless which algorithm is actually used, one important constraint is always valid for solving the general hand-eye calibration problem: At least two movements of the robot manipulator are necessary, where the axes of the rotations are non-parallel; this was shown by Tsai and Lenz (1989), and by Chen (1991). If the movement is not general enough, the hand-eye parameters can be recovered only partially; for details see, e. g., Andreff et al. (2001).

The critical factors and criteria for improving hand-eye calibration accuracy were already given by Tsai and Lenz (1989):

1. Maximize the angle between rotation axes of relative movements (influence on error in rotation, no translation recovery possible for parallel axes),
2. Maximize the rotation angle of relative movements (influence on error in rotation and translation),
3. Minimize the distance between the optical center of the camera and the calibration pattern (influence on error in translation),
4. Minimize the distance between the gripper coordinate system positions, i. e., small translational movement of the hand (influence on error in translation).

More details as well as an error analysis can be found in Tsai and Lenz (1989). If the movement of the robot gripper can be planned in advance, all items above may be controlled by the user. The usual way to fulfill the data requirements in robot hand-eye calibration is to use a calibration setup where the different positions of the gripper are chosen such that the data is well-suited for calibration. Such a setup is described, e. g., in Tsai and Lenz (1989).

In situations where planning movements is not possible or gripper movement is confined to certain areas, often not all criteria above can be controlled as desired. Particularly when hand-eye calibration is done for devices other than robot manipulator arms, controlled movements may be not possible at all or only with very low accuracy. An example of such an application that becomes more popular is the calibration of an optical tracking system (basically a camera) and an endoscope (optics and endoscopic camera). A so-called target, which consists of retroreflective markers, is mounted on the endoscope. Its 3D pose can then be determined by the optical tracking system. In this setup, the hand data is provided by the tracking system, the eye data by the endoscopic camera. Therefore, with a known hand-eye transformation, the movement of the endoscopic camera can be computed from the movement of the target. The endoscope is normally moved manually, and its exact pose cannot be controlled. For more details see Schmidt et al. (2004); Vogt (2006).

Another application is self-calibration of a stereo-camera system as discussed in Schmidt (2006). In this case, the rigid transformation between two cameras is estimated based on image information only, without using a calibration pattern. The cameras are either moved manually or implicitly by the head movement of a user in an Augmented Reality setting. Controlling criterion (3) in such a setup is usually not feasible: It would mean minimizing the distance of the cameras to the observed

scene, which would restrict the freedom of the user's movements considerably, rendering one of the main advantages of a self-calibration approach useless.

However, the main difference between these new application areas and classic hand-eye calibration is the way data is recorded. In a traditional setup, images of a calibration pattern are acquired at a comparatively small number of positions, say 20. In an Augmented Reality setting, where the user wearing a stereo camera is allowed to move freely, images are recorded at a rate of 25 or 30 frames per second. This results in an amount of data higher by one to two orders of magnitude, recorded at more or less arbitrary positions. Also, having a continuous movement where rotation and translation change only slightly between two consecutive frames, a trade-off between requirements (1) and (2) on one side and (4) on the other side has to be taken into account.

3 Data Selection

In this section various data selection algorithms will be presented that select a globally consistent set of relative movements that optimizes the non-parallelism criterion (1) as well as the rotation angle criterion (2) from Sect. 2.2. Criterion (1) is considered to be the most important one here, since no recovery of the hand-eye translation is possible if the rotation axes used are parallel.

3.1 Pre-Processing

The main purpose of the algorithms presented in this paper is increasing the hand-eye calibration accuracy when the hand positions were recorded continuously, and therefore consecutive poses differ only minimally. Given the criteria from Sect. 2.2, it can be expected that processing the data in temporal ordering is suboptimal; this is supported by the experimental results presented later on, hence using these readily available movements between consecutive poses cannot be recommended. Deciding which data, i. e., relative movements, should be used as input for the data selection method instead is an important step that has considerable influence on the performance of the algorithm. It has been proposed by Schmidt et al. (2003) to consider *all possible* relative movements between the recorded hand poses that are contained in the data and use these as input for the data selection algorithm. Given two robot gripper poses at time i and j consisting of rotation $\boldsymbol{R}_i$, $\boldsymbol{R}_j$ and translation $\boldsymbol{t}_i$, $\boldsymbol{t}_j$, the relative movement given by $\boldsymbol{R}_{ij}$, $\boldsymbol{t}_{ij}$ can be computed as follows:

$$\boldsymbol{R}_{ij} = \boldsymbol{R}_j^{\mathrm{T}} \boldsymbol{R}_i \tag{5}$$

for rotation and

$$\boldsymbol{t}_{ij} = \boldsymbol{R}_j^{\mathrm{T}} (\boldsymbol{t}_i - \boldsymbol{t}_j) = \boldsymbol{R}_{ij} \boldsymbol{R}_i^{\mathrm{T}} (\boldsymbol{t}_i - \boldsymbol{t}_j) \tag{6}$$

for translation.

For N poses, the total number of all relative movements is $N(N-1)/2$, i. e., the time complexity of the pre-processing step equals $O(N^2)$.

3.2 Vector Quantization – Overview

The basic idea of the following algorithms is: Given a set of N_{r} relative movements represented by their rotation axes, compute a new set of distinct axes consisting of N_{s} vectors, where $N_{\mathrm{s}} < N_{\mathrm{r}}$. This is achieved by running a clustering algorithm on the vectors representing axes, which computes a partitioning of the axes vectors.

A method which is suited very well for this task is *vector quantization* (see Linde et al., 1980). Note that although we propose to use vector quantization (and the LBG algorithm mentioned later), this should be seen as an example rather than a strict condition for data selection; in fact, *any* clustering algorithm can be applied.

In general, vector quantization works as follows: An input vector $\boldsymbol{x} \in \mathbb{R}^n$ is mapped to a vector of the so-called *codebook* $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{N_{\mathrm{s}}}\}$, which is a set of N_{s} n-dimensional vectors that define a partitioning of $\mathbb{R}^n$. Given a distance measure $d(\cdot,\cdot)$ on vectors in $\mathbb{R}^n$ (usually Euclidean distance), the input vectors are mapped as follows:

$$\begin{aligned} \boldsymbol{x} \mapsto \boldsymbol{c}_\kappa, \text{ where } d(\boldsymbol{x}, \boldsymbol{c}_\kappa) < d(\boldsymbol{x}, \boldsymbol{c}_i) \\ \forall i = 1, \ldots, N_{\mathrm{s}}, i \neq \kappa\,. \end{aligned} \tag{7}$$

Thus, the entries of the codebook $\mathcal{C}$ are the cluster centers in $\mathbb{R}^n$. For finding the entries of the codebook the well-known LBG algorithm (named after the authors Linde, Buzo, and Gray, see Linde et al. (1980)) is used, which is an iterative method that computes the codebook given the desired number of codebook entries. The complexity of the LBG algorithm for each iteration is $O(N_{\mathrm{r}} N_{\mathrm{s}})$, which equals $O(N^2 N_{\mathrm{s}})$. Note that the number of movements N_{r} is reduced further considerably by a pre-selection step removing movements with small rotation angles as described in the following section.

Various algorithms based on this idea are presented in the following. They differ in the input data used as well as in the dimensionality of vector quantization.

3.3 Three-Dimensional Vector Quantization of Normalized Rotation Axes

The vector quantization based data selection algorithm presented in this section requires 3D rotation axes $\boldsymbol{r}_i$ having norm one computed from relative movements as in-

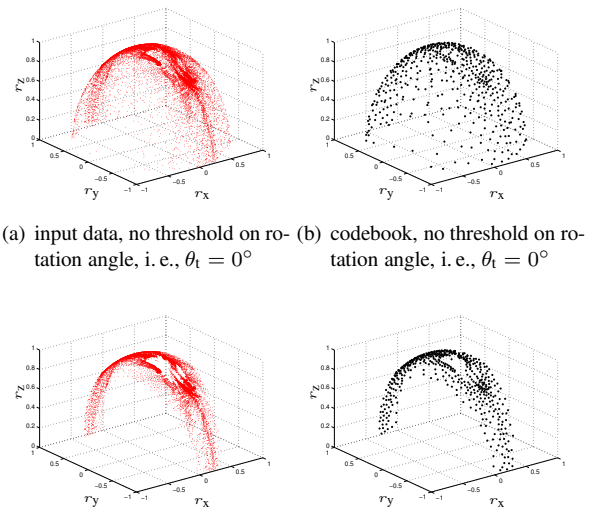
Input: N_r relative movements consisting of rotation and translation $\boldsymbol{R}_i$, $\boldsymbol{t}_i$ (cf. Sect. 3.1), θ_t = threshold for pre-selection according to rotation angle, N_s = number of desired relative movements after data selection is complete (codebook size). Output: N_s relative movements consisting of rotation and translation $\boldsymbol{R}_\kappa$, $\boldsymbol{t}_\kappa$.		
FOR each relative movement i		
	Compute axis $\boldsymbol{r}_i$ (norm one) and angle θ_i from $\boldsymbol{R}_i$	
	IF	$\lvert\theta_i\rvert < \theta_\mathrm{t}$ OR ($\lvert\theta_i\rvert > 180° - \theta_\mathrm{t}$ AND $\lvert\theta_i\rvert < 180° + \theta_\mathrm{t}$) OR $\lvert\theta_i\rvert > 360° - \theta$
	THEN	Rotation angle too small: remove movement i from data set
	ELSE	Resolve ambiguities (restrict to a single hemisphere)
Compute codebook $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{N_\mathrm{s}}\}$ of size N_s using the remaining $\boldsymbol{r}_i$ as training vectors		
FOR each remaining axis $\boldsymbol{r}_i$		
	Classify $\boldsymbol{r}_i$ to one of the partitions represented by codebook vector $\boldsymbol{c}_\kappa$: $\boldsymbol{r}_i \rightarrow \boldsymbol{r}_{i,\kappa}$	
	Compute the distance $d(\boldsymbol{r}_{i,\kappa}, \boldsymbol{c}_\kappa)$	
FOR each codebook entry $\boldsymbol{c}_\kappa$		
	Determine $\boldsymbol{r}_\kappa = \boldsymbol{r}_{j,\kappa}$, where $d(\boldsymbol{r}_{j,\kappa}, \boldsymbol{c}_\kappa) < d(\boldsymbol{r}_{i,\kappa}, \boldsymbol{c}_\kappa)\ \forall i, j$ of partition κ, $i \neq j$	
	Select the relative movement $\boldsymbol{R}_\kappa$, $\boldsymbol{t}_\kappa$ that corresponds to $\boldsymbol{r}_\kappa$ as one of the resulting movements	

Figure 1: Structure chart for data selection using a 3D vector quantization of normalized rotation axes.

put data. This algorithm was first published in Schmidt et al. (2004).

Before selecting the movements according to their non-parallelism, a pre-selection is done according to their rotation angle, because for angles close to zero the rotation axis is not well-defined (see Appendix A.2). For an angle of 180° singularities in hand-eye calibration arise (Shiu and Ahmad, 1989; Daniilidis, 1999) These are exactly the cases where the rotation matrix has multiple real Eigen-values. This step removes these movements, thus optimizing criterion (2). Movements are discarded that have rotation angles greater than a given threshold θ_t and less than $180° - \theta_\mathrm{t}$ or higher than $180° + \theta_\mathrm{t}$ and less than $360° - \theta_\mathrm{t}$. The second interval is due to the fact that a rotation about an axis $\boldsymbol{r}$ by an angle θ is the same as a rotation about the axis $-\boldsymbol{r}$ by the angle $360° - \theta$. After the pre-selection step, only the rotation axes (normalized to one) are used for further processing. The complete algorithm is shown in Fig. 1.

After pre-selection according to rotation angle and normalization of the rotation axes to one, the ambiguity in the axis/angle representation is resolved to make sure that similar rotation axes are actually close to each other in 3D. Since all normalized rotation axes $\boldsymbol{r}_i = \begin{pmatrix} r_{i\mathrm{x}} & r_{i\mathrm{y}} & r_{i\mathrm{z}} \end{pmatrix}^\mathrm{T}$ lie on a sphere in 3D space, this can be achieved by restricting the axes to one hemisphere. Here, w. l. o. g. the hemisphere with non-negative $r_{i\mathrm{z}}$-coordinate was chosen. If this coordinate of an axis $\boldsymbol{r}_i$ is negative, the axis $\boldsymbol{r}_i$ is substituted by $-\boldsymbol{r}_i$. Rotation axes having a zero $r_{i\mathrm{z}}$-coordinate have to be handled separately by checking the $r_{i\mathrm{y}}$- and $r_{i\mathrm{x}}$-coordinates.



(a) input data, no threshold on rotation angle, i. e., $\theta_\mathrm{t} = 0°$

(b) codebook, no threshold on rotation angle, i. e., $\theta_\mathrm{t} = 0°$

(c) input data, threshold of $\theta_\mathrm{t} = 15°$

(d) codebook, threshold of $\theta_\mathrm{t} = 15°$

Figure 2: Examples of vector quantization result in 3D based on real data: The input data (rotation axes of norm one) in (a), (c) are shown as small dots, the codebook are plotted in (b), (d) as bold dots. Because of the normalization of the axes to one, all vectors lie on a sphere.

The next step is training the vector quantizer, i. e.,

computation of the codebook vectors $\mathcal{C}$, which results in a clustering of the rotation axes. Note that since all axes have norm one, the vectors are not uniformly distributed in space, but lie on the surface of the unit sphere. An example obtained from real data is depicted in Fig. 2. For the data set shown in Fig. 2(a) no pre-selection of the data with respect to small rotation angles was done, i. e., all relative movements were used as input for vector quantization. The resulting codebook is plotted in Fig. 2(b). Figure 2(c) shows the same data set, where relative movements having a rotation angle smaller than $15°$ have been removed, and the generated codebook is depicted in Fig. 2(d).

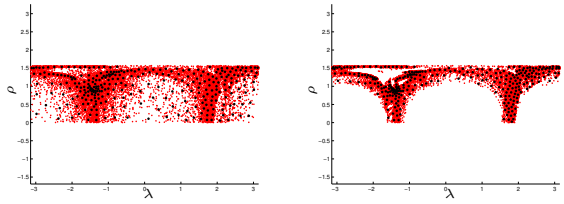
In many algorithms that apply vector quantization, the codebook vectors can be directly used for further processing; note that this is not the case for data selection as described here. Codebook vectors are computed as the center of gravity (i. e., mean values) of all input vectors belonging to a certain partition. Therefore, a codebook vector usually does not coincide with an element of the input vector set, which means that it cannot be related to an actual relative movement. Additional steps have to be taken in order to obtain a single rotation axis (and the associated relative movement) per partition: Firstly, each rotation axis $\boldsymbol{r}_i$ has to be classified to one of the partitions defined by the codebook vectors. The classified axes are denoted by $\boldsymbol{r}_{i,\kappa}$. Secondly, for each rotation axis $\boldsymbol{r}_{i,\kappa}$ of a partition κ, the distance to the codebook vector $\boldsymbol{c}_\kappa$ representing that partition is computed; the selected axis is the one where the distance to the codebook vector $d(\boldsymbol{r}_{i,\kappa}, \boldsymbol{c}_\kappa)$ is smallest. The relative movements belonging to the rotation axes selected this way can now be used for hand-eye calibration.

3.4 Two-Dimensional Vector Quantization of Normalized Rotation Axes

The data selection algorithm presented before uses normalized rotation axes as input, where the rotation angle is not encoded in the axis but handled separately, an axis being a 3D vector with only two degrees of freedom. Hence, the dimensionality of the vector quantization can be reduced from three to two by using an appropriate parameterization of the axes. An obvious choice for this task are polar coordinates. Given a rotation axis $\boldsymbol{r}$, the polar coordinates λ, ρ of $\boldsymbol{r}$ are computed as:

$$\lambda = \arctan \frac{r_\mathrm{y}}{r_\mathrm{x}}, \quad \rho = \arcsin r_\mathrm{z} \,. \tag{8}$$

The data selection algorithm using polar coordinates is similar to the one before (shown in Fig. 1). The main difference is that an additional computation step preceding the codebook generation has to be introduced that converts the normalized rotation axes $\boldsymbol{r}_i$ to polar coordinates using (8). This reduces the dimensionality of the following vector quantization, as only 2D vectors that contain the angles λ_i and ρ_i are used.



(a) No threshold on rotation angle, i. e., $\theta_\mathrm{t} = 0°$ (b) Threshold of $\theta_\mathrm{t} = 15°$

Figure 3: Examples of vector quantization result using polar coordinates based on real data: The input data, i. e., normalized rotation axes converted to polar coordinates, are represented by the small dots, the codebook vectors are plotted as bold dots. The angles λ and ρ are given in rad.

An example of a vector quantization result using polar coordinates is shown in Fig. 3. The same data set as in Fig. 2 was used.

3.5 Automatic Computation of Rotation Angle Thresholds

A drawback of the two data selection algorithms presented in Sects. 3.3 and 3.4 is that a threshold θ_t for the rotation angle has to be set manually. There are mainly two ways to improve the algorithm:

1. Use a threshold, but compute it automatically from the available data.

2. Do not use a threshold at all, i. e., all data are used for vector quantization; the rotation angle is taken into account implicitly by an appropriate parameterization.

The former is discussed here; algorithms using the second option are shown in Sects. 3.6 and 3.7.

In the data selection algorithms presented before a single threshold was used, which is applied to the lower ($0°$, $360°$) and upper ($180°$) bounds of the rotation angle interval symmetrically. The best-suited movements have rotation angles located in the center of the two intervals at $90°$ and $270°$. The algorithm for automatic threshold computation is different in two ways: Two separate thresholds for the upper and lower bound are calculated, and these are not necessarily symmetric. The structure chart for automatic threshold computation is shown in Fig. 4.

The algorithm requires the desired remaining fraction of movements to be specified as an input parameter. This is a clear advantage over using a threshold for the angle

Input: N_r relative movements consisting of rotation and translation $\boldsymbol{R}_i$, $\boldsymbol{t}_i$ (cf. Sect. 3.1), $\delta =$ desired remaining fraction (0 to 1) of relative movements after pre-selection. Output: $\theta_\mathrm{u}, \theta_\mathrm{l} =$ upper and lower threshold for pre-selection according to rotation angle.				
FOR each relative movement i				
	Compute rotation angle θ_i from $\boldsymbol{R}_i$			
	Take absolute angle, i. e., $\theta_i := \lvert\theta_i\rvert$			
	IF	$\theta_i > 180°$		
	THEN	Normalize angle to interval $0°$ to $180°$: $\theta_i := 360° - \theta_i$		
	Store angle θ_i in a list $\mathcal{L}$ (indexed from $0, \ldots, N_\mathrm{r} - 1$)			
Sort list $\mathcal{L}$ (ascending)				
$\alpha_\mathrm{l} := \mathcal{L}(0)$, $\alpha_\mathrm{u} := \mathcal{L}(N_\mathrm{r} - 1)$				
IF	$\alpha_\mathrm{l} > 90°$			
THEN	$\theta_\mathrm{l} := 0°$, $\theta_\mathrm{u} := 180° - \mathcal{L}((N_\mathrm{r} - 1) - \mathrm{Round}((1-\delta)\cdot(N_\mathrm{r}+1)))$			
ELSE	IF	$\alpha_\mathrm{u} < 90°$		
	THEN	$\theta_\mathrm{l} := \mathcal{L}(\mathrm{Round}((1-\delta)\cdot(N_\mathrm{r}-1)))$, $\theta_\mathrm{u} := 0°$		
	ELSE	IF	$\alpha_\mathrm{l} \leq 90°$ AND $\alpha_\mathrm{u} \geq 90°$	
		THEN	Find index i, where $\mathcal{L}(i) \geq 90°$ and $\mathcal{L}(j) < 90°\ \forall j = 0, \ldots, i-1$	
			Compute fractions ξ_l, ξ_u of list elements left and right of i (cf. (9))	
			Compute fractions ρ_s, ρ_b to be removed on left and right (cf. (10))	
			IF	$\xi_\mathrm{l} \geq \xi_\mathrm{u}$
			THEN	$\theta_\mathrm{l} := \mathcal{L}(\mathrm{Round}(\rho_\mathrm{b}(N_\mathrm{r}-1)))$
				$\theta_\mathrm{u} := 180° - \mathcal{L}((N_\mathrm{r}-1) - \mathrm{Round}(\rho_\mathrm{s}(N_\mathrm{r}+1))$
			ELSE	$\theta_\mathrm{l} := \mathcal{L}(\mathrm{Round}(\rho_\mathrm{s}(N_\mathrm{r}-1)))$
				$\theta_\mathrm{u} := 180° - \mathcal{L}((N_\mathrm{r}-1) - \mathrm{Round}(\rho_\mathrm{b}(N_\mathrm{r}+1))$

Figure 4: This structure chart shows an algorithm for computing upper and lower rotation angle thresholds θ_u, θ_l automatically.

directly: When an explicit angle threshold is provided, more often than not it will be chosen either too high or too low, as we have no a-priori knowledge on the data used for calibration. If the threshold is too high, there may be no movement left after applying the threshold, which makes calibration impossible. If chosen too low, possibly lots of movements are processed further that are actually not very good and thus distort the calibration result. Since the rotation angles are highly dependent on the recorded image sequence, a general recommendation for choosing a threshold on the angle is not possible. In contrast, specifying a percentage of movements that are to remain after pre-selection allows choosing the best ones while it is guaranteed that a sufficient number of movements is left for calibration.

First, the rotation angle θ_i is computed for each relative movement; the rotation axes are irrelevant for threshold determination. Since only the amount of rotation is of interest we take the absolute values of θ_i. Then, all angles are normalized to the interval $0°$ to $180°$ and stored in a list $\mathcal{L}$ which can be accessed by an index ranging from 0 to $N_\mathrm{r} - 1$. After sorting the list in ascending order, the smallest and largest rotation angles α_l and α_u that are contained in the recorded image sequence can be found in the entries $\mathcal{L}(0)$ and $\mathcal{L}(N_\mathrm{r} - 1)$.

Recall that after normalization the best-suited movements have a rotation angle around $90°$. Three cases have to be distinguished. If the smallest rotation angle α_l is greater than $90°$, no movements are contained in the sequence that have to be removed at the lower bound. Therefore, the lower threshold θ_l is set to $0°$. All movements have to be removed at the upper bound, and the upper threshold is defined by the list entry having index $(N_\mathrm{r} - 1) - \mathrm{Round}((1-\delta)\cdot(N_\mathrm{r}+1))$.

A similar situation, with reversed roles of thresholds, arises if the largest angle α_u is smaller than $90°$.

In this case all movements have to be removed at the lower bound, and the upper threshold θ_u is set to $0°$, while the lower one is given by the list entry at index $\mathrm{Round}((1-\delta)\cdot(N_\mathrm{r}-1))$.

The case where angles below $90°$ exist as well as angles above $90°$ is slightly more complex, as both thresholds have to be chosen asymmetrically depending on the number of movements with rotation angles greater and smaller than $90°$. Therefore, the first step is to identify the position i of the first angle in the list that is equal or larger than $90°$, i. e., the "center" of the list with respect to the best angle contained in the sequence. Now the fraction of angles left and right of the identified position i can be computed as:

$$\xi_\mathrm{l} = \frac{i}{N_\mathrm{r}-1}\,, \quad \xi_\mathrm{u} = 1-\xi_\mathrm{l} \quad , \tag{9}$$

where ξ_l is the fraction of movements in the lower part and ξ_u in the upper part with respect to $90°$. When deciding about which movements are to be removed, it is important to keep as many close to $90°$ as possible, i. e., in the majority of cases most movements will be deleted in the part of the list containing the greater amount of movements, and only a small fraction (if any) in the part containing the smaller amount of movements. We compute two factors ρ_s and ρ_b that determine the fraction of movements to be deleted in the smaller (ρ_s) and the bigger (ρ_b) one of the two parts of the list. These factors are given by:

$$\begin{aligned}\rho_\mathrm{s} &= \max\left(\frac{1}{2}((1-\delta)-|\xi_\mathrm{l}-\xi_\mathrm{u}|),0\right) \quad ,\\ \rho_\mathrm{b} &= \min((1-\delta),\max(\xi_\mathrm{l},\xi_\mathrm{u})-\min(\xi_\mathrm{l},\xi_\mathrm{u}))+\rho_\mathrm{s}.\end{aligned} \tag{10}$$

The first addend in the equation for ρ_b removes as many movements as possible from the bigger part, i. e., either all that have been required or an amount that makes it the same size as the originally smaller one. The remaining movements can then be removed symmetrically from both parts, the amount being defined by ρ_s.

If most movements are located in the lower part of the list, the thresholds θ_l, θ_u are then defined by the list elements having index $\mathrm{Round}(\rho_\mathrm{b}(N_\mathrm{r}-1))$ and $(N_\mathrm{r}-1)-\mathrm{Round}(\rho_\mathrm{s}(N_\mathrm{r}+1))$, respectively. In case that most movements are located in the upper part of the list, ρ_s and ρ_b change roles when computing the indices of the list elements.

Strictly speaking checking the first two cases, where all movements are located on one side of $90°$, is not necessary, because both are covered by (10). Nevertheless they have been included in the structure chart, as we believe the concept is easier to understand this way.

3.6 Vector Quantization Using Axis/Angle Representation

This section presents a data selection algorithm that does not need thresholds on rotation angles as no pre-selection of relative movements is done, i. e., all available movements are used. A structure chart is shown in Fig. 5.

Instead of treating rotation axis and angle separately, the axis/angle representation is used as described in Appendix A.2, where the angle θ is encoded as the norm of the axis vector $\boldsymbol{r}$ in a 3D vector $\boldsymbol{\omega}$ having three degrees of freedom.

We start with a set of N_r relative movements, now represented by their rotation axes with angles encoded in $\boldsymbol{\omega}_i$. The result is a set of N_s vectors, $N_\mathrm{s} < N_\mathrm{r}$, where the corresponding selected movements are a trade-off between criterion (1) and (2) as defined in Sect. 2.2: Movements having small rotation angles will be found in the resulting data set if their rotation axes fit well to the remaining data.

At the beginning all $\boldsymbol{\omega}_i$ are normalized such that angles are in the range $0°$ to $180°$. This is different from the normalization in the previous data selection methods, where the sign of the axes elements is used. Generally, there are always these two options: either the sign of an axis is controlled or the rotation angle, never both. Depending on the application, either option may have advantages and disadvantages.

In contrast to the methods presented before, where movements with rotation angles close to zero are scattered, they will be concentrated near the origin of the coordinate system now. Therefore, the pre-selection using a rotation angle threshold can be omitted, as the vector quantization step will only select a few movements having unsuitable rotation angles, while the majority will be alright. This feature is the main advantage of the algorithm. Also, 3D vector quantization can be used in a straight forward fashion, as the 3D axis/angle vectors $\boldsymbol{\omega}_i$ have three degrees of freedom.

The basic principle of the rest of the algorithm remains the same as before, except that vector quantization is now done on the vector where rotation axis and angle are encoded.

3.7 Vector Quantization Using Quaternions

As in the previous section a data selection algorithm is presented here that does not remove movements having small rotation angles and therefore has no need for thresholds. Now, the quaternion representation of 3D rotations is used (see Appendix A.3).

After the quaternion representation has been computed for each relative movement, the ambiguity in the quaternion representation has to be resolved. Since the quater-

Input: N_r relative movements consisting of rotation and translation $\boldsymbol{R}_i$, $\boldsymbol{t}_i$ (cf. Sect. 3.1), N_s = number of desired relative movements after data selection is complete. Output: N_s relative movements consisting of rotation and translation $\boldsymbol{R}_\kappa$, $\boldsymbol{t}_\kappa$.		
FOR each relative movement i		
	Compute axis/angle $\boldsymbol{\omega}_i$ from $\boldsymbol{R}_i$; angle $\theta_i = \lvert\boldsymbol{\omega}_i\rvert$	
	IF	$\theta_i > 180°$
	THEN substitute $\boldsymbol{\omega}_i$ by a rotation about the negative axis and angle $360° - \theta_i$	
Compute codebook $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{N_\mathrm{s}}\}$ of size N_s using the $\boldsymbol{\omega}_i$ as training vectors		
FOR each vector $\boldsymbol{\omega}_i$		
	Classify $\boldsymbol{\omega}_i$ to one of the partitions represented by codebook vector $\boldsymbol{c}_\kappa$: $\boldsymbol{\omega}_i \rightarrow \boldsymbol{\omega}_{i,\kappa}$	
	Compute the distance $d(\boldsymbol{\omega}_{i,\kappa}, \boldsymbol{c}_\kappa)$	
FOR each codebook entry $\boldsymbol{c}_\kappa$		
	Determine $\boldsymbol{\omega}_\kappa = \boldsymbol{\omega}_{j,\kappa}$, where $d(\boldsymbol{\omega}_{j,\kappa}, \boldsymbol{c}_\kappa) < d(\boldsymbol{\omega}_{i,\kappa}, \boldsymbol{c}_\kappa)$ $\forall i, j$ of partition κ, $i \neq j$	
	Select the relative movement $\boldsymbol{R}_\kappa$, $\boldsymbol{t}_\kappa$ that corresponds to $\boldsymbol{\omega}_\kappa$ as one of the resulting movements	

Figure 5: Structure chart for data selection using axis/angle representation.

nions $\boldsymbol{q}_i$ and $-\boldsymbol{q}_i$ represent the same rotation, we restrict the quaternions to the hyper-hemisphere with positive real part. This can be done similarly to the axis ambiguity resolution discussed before in Sect. 3.3, with the difference that a quaternion consists of four elements instead of three. The remaining part of the algorithm is similar to the data selection methods discussed before. A main advantage of quaternions compared to using the rotation axis is that the quaternion is well-defined for arbitrary rotation angles. While the rotation axis is undefined for a rotation angle of zero, i. e., for movements where the rotation matrix equals $\boldsymbol{I}_{3\times3}$, the corresponding quaternion is defined and equals 1. The main disadvantage of using quaternions is that these consist of four elements with only three degrees of freedom; therefore, a 4D vector quantization has to be used instead of a three-dimensional one. Of course, as before when rotation axes were discussed, polar angles representing quaternions could be used, as these have norm one and thus lie on a hypersphere.

4 Experimental Results

This section presents an experimental evaluation of the vector quantization based data selection algorithms. It starts with an introduction to the metrics used for residual error computation in Sect. 4.1. In Sect. 4.2 the data sets used are described, followed by the experimental results in Sects. 4.3 to 4.6. In particular we will look into the following topics: How does the codebook size used for vector quantization influence the calibration error (Sect. 4.3), how does the pre-selection threshold on the rotation angle affect the result (Sect. 4.4), and which one of the presented data selection methods performs best (Sect. 4.5)? Finally, Sect. 4.6 shows how the data selection performs compared to manually selecting well-suited poses during data acquisition (although this is not the main application area of the proposed methods). The actual hand-eye calibration was done using the linear dual quaternion algorithm by Daniilidis (1999, 2001).

4.1 Residual Error Metrics

For experimental evaluation we need error metrics for rotation and translation that measure the accuracy of hand-eye calibration. Commonly, the error in translation is given as a relative error, while for rotation an absolute error metric is used (e. g., Horaud and Dornaika, 1995; Daniilidis, 1999; Andreff et al., 2001). In this paper, absolute and relative errors will be shown for both, rotation and translation. The absolute residual error for translation is given by

$$\epsilon_{\boldsymbol{t}\mathrm{abs}} = \frac{1}{N}\sum_{i=1}^{N} \lVert\widehat{\boldsymbol{t}}_i - \boldsymbol{t}_i\rVert \quad , \tag{11}$$

and the relative residual error by

$$\epsilon_{\boldsymbol{t}\text{rel}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\widehat{\boldsymbol{t}}_i - \boldsymbol{t}_i\|}{\|\boldsymbol{t}_i\|} \quad , \tag{12}$$

where N is the number of translation vectors used for error computation (see below), $\boldsymbol{t}_i$ is the true translation vector, and $\widehat{\boldsymbol{t}}_i$ is the vector estimated by hand-eye calibration.

Different metrics for errors in rotation are used in literature. While in Horaud and Dornaika (1995) the norm of the difference between two rotation matrices is given, this work follows (Daniilidis, 1999; Andreff et al., 2001) and uses the norm of quaternion differences instead for relative residual errors, given by:

$$\epsilon_{\boldsymbol{R}\text{rel}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\widehat{\boldsymbol{q}}_i - \boldsymbol{q}_i\|}{\|1 - \boldsymbol{q}_i\|} \quad . \tag{13}$$

The norm of quaternion differences is obviously connected to the rotation angle as well as to the angle between the two rotation axes:

$$\|\widehat{\boldsymbol{q}}_i - \boldsymbol{q}_i\|^2 = 2 - 2(\cos\widehat{\theta}_i\cos\theta_i + \widehat{\boldsymbol{r}}_i^{\mathrm{T}}\boldsymbol{r}_i\sin\widehat{\theta}_i\sin\theta_i) \quad , \tag{14}$$

where $\widehat{\theta}_i, \theta_i$ are the rotation angles and $\widehat{\boldsymbol{r}}_i, \boldsymbol{r}_i$ the rotation axes corresponding to the quaternions $\widehat{\boldsymbol{q}}_i$ and $\boldsymbol{q}_i$, respectively.

For rotations about the same axis, but by different angles, this metric has the property that it is directly connected to the residual rotation angle, as (14) can be simplified to:

$$\|\widehat{\boldsymbol{q}}_i - \boldsymbol{q}_i\| = \sqrt{2 - 2\cos\frac{\widehat{\theta}_i - \theta_i}{2}} \quad , \tag{15}$$

The absolute residuals can either be given using quaternions as well, or in degrees based on the axis/angle representation of the residual rotation matrix $\boldsymbol{R}_{\text{res}\,i}$, which is given by:

$$\boldsymbol{R}_{\text{res}\,i} = \widehat{\boldsymbol{R}}_i^{\mathrm{T}}\boldsymbol{R}_i \quad . \tag{16}$$

A simplified absolute rotational residual error can now be defined by the rotation angle $\theta_{\text{res}\,i}$, which can be computed from one of the complex Eigen-values of $\boldsymbol{R}_{\text{res}\,i}$:

$$\epsilon'_{\boldsymbol{R}\text{abs}} = \frac{1}{N}\sum_{i=1}^{N}|\theta_{\text{res}\,i}| \quad . \tag{17}$$

The advantage of using this metric instead of the quaternion based one is simply that an absolute residual given in degrees makes it easier for the reader to judge whether the error is high or low.

Figure 6: Left: Optical tracking system smARTtrack1 in the laboratory. Right: smARTtrack1 in the operating room during an endoscopic surgery. Images by courtesy of F. Vogt.

The metrics presented above are used for computing a prediction error, which is the residual between the predicted eye position computed from hand data and the estimated hand-eye transformation, and the real (calibrated) eye pose. In order to give an overall residual error, a set of relative movements (we use 100) is selected randomly from the complete set of all possible relative movements (cf. Sect. 3.1). Note that again it is of disadvantage to use relative movements between subsequent positions, because the movements will usually be small, which results in large relative errors and thus does not reflect the actual quality of the estimated hand-eye transformation. The results shown in this section's tables have been obtained by iterating the above process 100 times and averaging the resulting residual errors. The reason why the randomly chosen movements are selected from the complete data set rather than the set obtained after vector quantization is that the latter would not result in a valid residual error that can be used to describe the actual calibration accuracy, because these data have been used for parameter estimation. As the estimation process optimizes the hand-eye parameters on the provided data, the residual error would always be minimal.

To summarize: The residual errors in translation shown in the following were computed using (11) and (12). For relative residual errors in rotation, (13) was used. The absolute rotational errors show the average rotation angle in degrees, which was computed using (17). All results are given with an accuracy of three valid digits.

4.2 Description of Data Sets

Instead of a robot as in classic hand-eye calibration, we have used an optical tracking system in our experiments. The infrared optical tracking system smARTtrack1 by Advanced Realtime GmbH (shown in Fig. 6) provides pose data of a so-called target (the *hand*) that is fixed to an endoscope. It is a typical optical tracking system consisting of two infrared cameras and the target, which

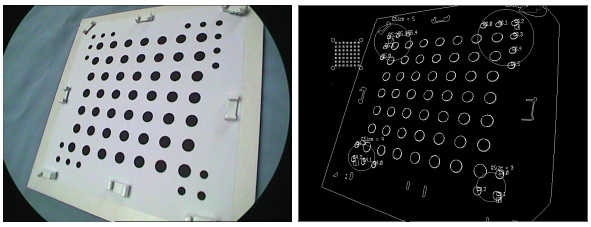


Figure 7: Left: Asymmetric 7 × 7 calibration pattern with marked corners. Right: Processed image of calibration pattern after ellipse-fitting.

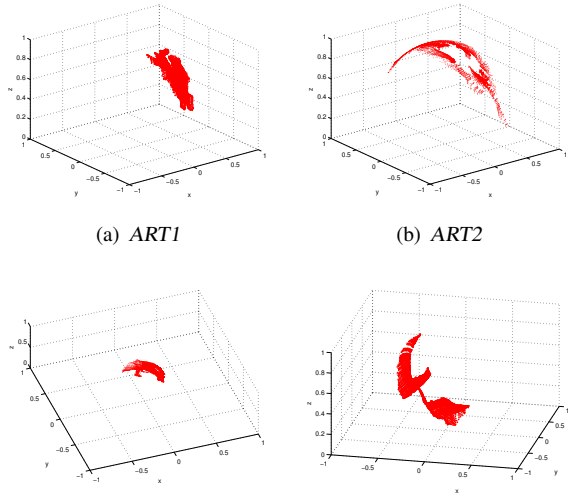
is built from markers that can easily be identified in the images captured by the cameras. Spheres with a retro-reflective surface are used, and marker identification is simplified by active illumination with infrared light. The 3D position of each visible marker is calculated by the tracking system; knowledge of the geometry of the target then allows to calculate its pose. The accuracy of the pose is 0.19 mm in x- and y-direction, 0.36 mm in z-direction, and 0.14° for rotation.

A CCD camera is mounted rigidly on the endoscope, which is moved manually. The objective of hand-eye calibration is to determine the unknown transformation from the target-pose provided by the optical tracking system to 3D camera coordinates.

The camera (*eye*) poses are computed using a calibration pattern and standard camera calibration techniques (Zhang, 1998, 2000). We use an asymmetric pattern as shown in Fig. 7 (left), which consists of 49 circular calibration points arranged in a 7 × 7 pattern. The corners of the pattern are coded using smaller dots, to allow for resolving ambiguities when the camera is moved. We extract contours in the captured image of the calibration pattern, and perform ellipse-fitting on each contour (see Fig. 7 (right)). The center of each ellipse is used as a 2-D calibration point, which provides sub-pixel accuracy; the measured backprojection error is smaller than 0.2 pixels.

Results from eight representative data sets acquired this way are shown here. These have been selected carefully out of many more experiments that we have conducted, to describe effects that can be observed when using the proposed data selection methods. Four sets contain a large number of data continuously recorded while the endoscope was moved manually. These are denoted by *ART1*, *ART2*, *ART3*, and *ART4* in the following. The remaining four contain only a small number of poses, which were recorded at manually selected positions, as in a classic hand-eye calibration. They will be described later on.

The continuously recorded data sets differ mainly in the number of poses contained in each sequence (270 for *ART1*, 190 for *ART2*, 200 for *ART3* and *ART4*), and



(a) *ART1* (b) *ART2*

(c) *ART3* (d) *ART4*

Figure 8: Comparison of normalized rotation axes contained in the continuously recorded data sets after pre-selection: (a) *ART1*, (b) *ART2*, (c) *ART3*, (d) *ART4*. The variation in the rotation axes is much larger for *ART2* and *ART4* compared to *ART1* or *ART3*, making the former better suited for hand-eye calibration.

in the type of movement that was done while they were recorded. The camera-endoscope configuration, i. e., the hand-eye transformation, is different for all data sets as the camera was mounted on the endoscope anew every time. An overview over the main properties of the data sets including the results of automatic threshold computation are shown in Table 1.

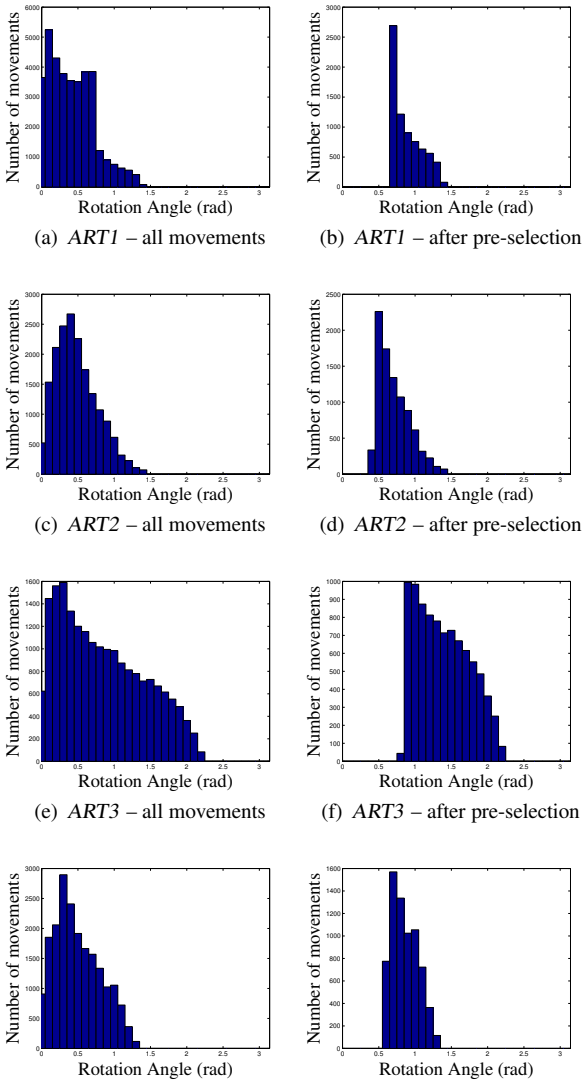
As the accuracy of the hand-eye calibration depends considerably on the amount of rotational movement in terms of both, distinct rotation axes as well as large rotation angles, the quality of a data set can be assessed by plotting the distribution of normalized rotation axes and histograms of rotation angles shown in Figs. 8 and 9, respectively.

As can be seen in the plots of normalized rotation axes after pre-selection in Fig. 8, the rotational movement in *ART1* and *ART3* is smaller than in *ART4*, and considerably smaller than in *ART2*. It is important to note that a data set has to contain a certain amount of rotational movement to get good results, therefore, based on rotation axes, the data sets *ART2* and *ART4* are much better suited for hand-eye calibration than *ART1* and *ART3*, due to the small coverage of the latter.

Figure 9 shows histograms of rotation angles contained in the data sets before (left column) and after (right column) pre-selection of movements using an automatically computed rotation angle threshold. It can be observed that the majority of movements before pre-

Sequence	*ART1*	*ART2*	*ART3*	*ART4*
number of poses	270	190	200	200
total number of relative movements	36315	17955	19900	19900
number of movements after applying threshold	7261	8976	8954	6964
minimum angle in data set	0.00000171°	0.0388°	0.135°	0.0406°
maximum angle in data set	80.3°	80.9°	126°	73.8°
minimum angle after applying threshold	38.5°	25.0°	48.4°	34.5°
maximum angle after applying threshold	80.3°	80.9°	126°	73.8°
median angle in data set	21.8°	25.2°	42.8°	25.2°
median angle after threshold	47.0°	41.2°	73.4°	47.6°

Table 1: Properties of the data sets used in the experiments, including data selection information.



(a) *ART1* – all movements (b) *ART1* – after pre-selection

(c) *ART2* – all movements (d) *ART2* – after pre-selection

(e) *ART3* – all movements (f) *ART3* – after pre-selection

(g) *ART4* – all movements (h) *ART4* – after pre-selection

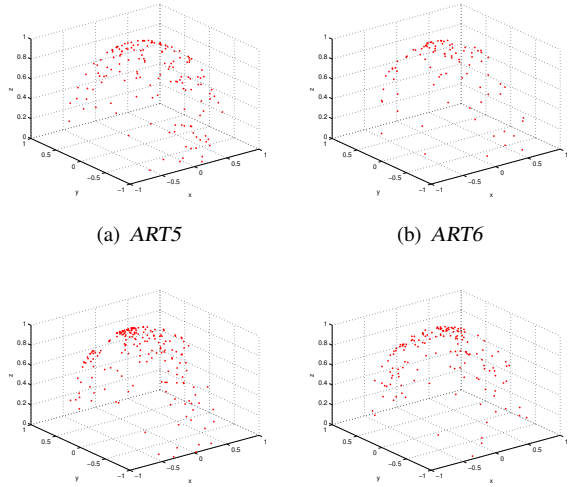
Figure 9: Histograms of rotation angles (in rad) in the data sets: (a), (c), (e), (g) complete data; (b), (d), (f), (h) after pre-selection.

selection have small rotation angles, and only *ART3* contains angles greater than 90° at all. Recall that for obtaining a very accurate calibration we would need movements with rotation angles around 90°. Based on the histograms we can conclude that *ART3* is the best-suited data set; however, combined with the distribution of axes (Fig. 8(c)), which has low coverage, this is put into perspective, so judging the calibration outcome for *ART3* in advance from the plots is inconclusive.

Based on these figures it can already be predicted that *ART1* is probably a data set that results in a low-accuracy calibration, as both, rotation axes and angles, are not suited well for hand-eye calibration. Also, it is likely that *ART2* and *ART4* will give better results, as the coverage of axes is good, and the histograms of angles are acceptable, although not optimal.

In order to get an idea of the quality of the hand-eye calibration obtained from continuously recorded sequences, the results were compared to the accuracy of performing hand-eye calibration in the classic way, i. e., using a small number of poses recorded at manually selected distinct positions that are well-suited for hand-eye calibration. They have been acquired using the same camera-endoscope configuration as continuously recorded data, and are denoted by *ART5* (18 poses), *ART6* (14 poses), both corresponding to *ART2*, *ART7* (20 poses, corresponding to *ART3*), and *ART8* (18 poses, corresponding to *ART4*).

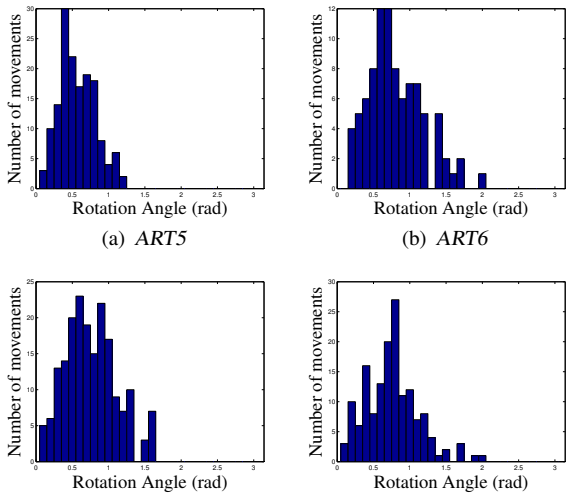
Figure 10 shows plots of the distribution of rotation axes of all relative movements contained in these four data sets. Comparing these figures to the plots in Fig. 8, it can be observed that the coverage is much better than for the continuous sequences, although they obviously contain much less data; but it is not quantity that counts. Histograms of the rotation angles (without any pre-selection) are shown in Fig. 11. As opposed to the histograms in Fig. 9, the manually recorded data sets contain much less very small angles close to 0°, and although they too are far from being optimal, except for *ART5* they all contain angles greater than (but still close to) 90°.



(a) *ART5* (b) *ART6*

(c) *ART7* (d) *ART8*

Figure 10: Comparison of normalized rotation axes contained in the data sets recorded at manually selected positions: (a) *ART5*, (b) *ART6*, (c) *ART7*, (d) *ART8*. These data sets contain less poses than the continuously recorded ones, but the distribution of rotation axes is much better suited for hand-eye calibration.

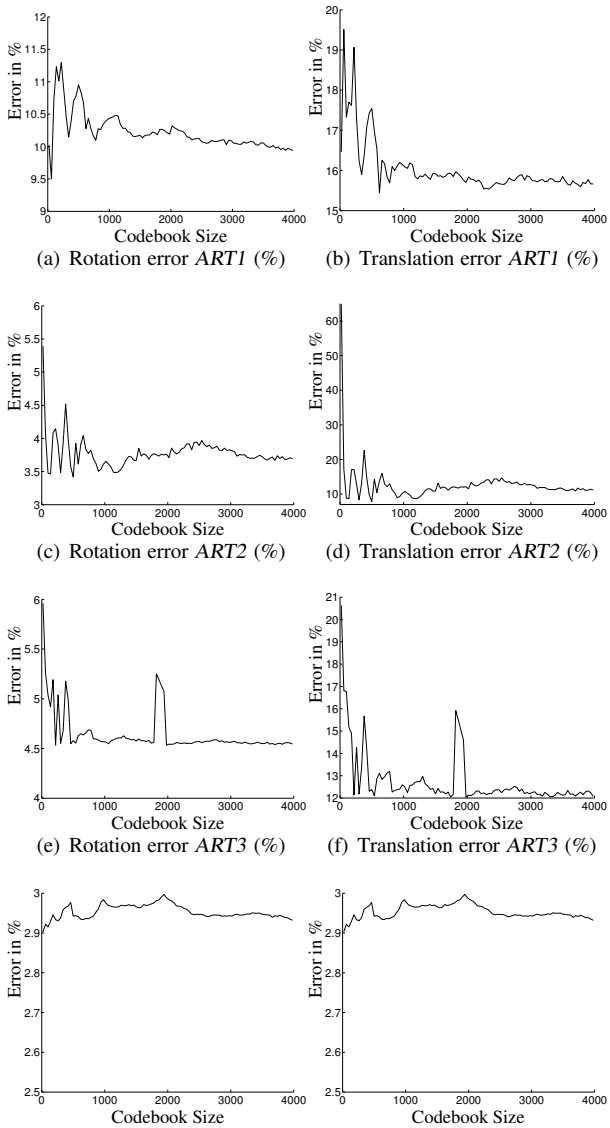


(a) *ART5* (b) *ART6*

(c) *ART7* (d) *ART8*

Figure 11: Histograms of rotation angles (in rad) in the data sets recorded at manually selected positions.

To summarize: Even without performing an actual calibration, it can be concluded from the plots alone that in most cases continuously recorded data will lead to a less accurate calibration compared to data containing manually selected positions. However, in many applications it may be very inconvenient or even impossible (e. g., due to time constraints during a surgical intervention as in case of the endoscope) to use manually selected positions. In the following sections we present an evaluation of the data selection algorithms, compare continuously recorded data to data acquired at manually selected positions calibrated the classic way, and show that even for the manually selected data sets performing a data selection can result in a more accurate calibration.



(a) Rotation error *ART1* (%) (b) Translation error *ART1* (%)

(c) Rotation error *ART2* (%) (d) Translation error *ART2* (%)

(e) Rotation error *ART3* (%) (f) Translation error *ART3* (%)

(g) Rotation error *ART4* (%) (h) Translation error *ART4* (%)

Figure 12: Mean relative residual errors in rotation and translation dependent on the codebook size used for vector quantization for the data sets *ART1*, *ART2*, *ART3*, and *ART4*.

4.3 Codebook Size

In this section the influence of one of the two data selection parameters is analyzed, namely the codebook size used for vector quantization. The hand-eye cali-

bration residual errors were computed varying the codebook size, while the other parameter was left constant. For data selection, the 3D vector quantization algorithm on normalized rotation axes (Sect. 3.3) was chosen, the pre-selection step that removes movements having small rotation angles was done using the automatic threshold computation (Sect. 3.5).

Plots of the relative residual errors in rotation and translation for the continuously recorded data sets *ART1*, *ART2*, *ART3*, and *ART4* are shown in Fig. 12; the graphs for absolute residuals have been omitted, as they look very much like the relative ones and therefore do not provide much more information.

The codebook size has been varied from 20 to 4000 relative movements, which is 0.22% to 57% (depending on the data set) of the number of relative movements remaining after pre-selection with respect to the rotation angle, and only in between 11% and 22% of the total number of relative movements contained in the sequence. Typically, what can be observed is the behavior of the data sets *ART1* and *ART2*, and, with the exception of some outliers in the middle, also *ART3* as shown in Figs. 12(a) to 12(f): Fluctuations are quite high for small codebook sizes, the residual error becoming more stable the more data is used for calibration. In contrast to the other data sets, *ART4* is stable from the beginning (note the scale of the vertical axis). This is probably due to the fact *ART4* is obviously the best suited data set for hand-eye calibration judging by the residual errors.

For practical purposes using low values for the codebook size is preferable to high ones, as the number of vectors in the codebook has a twofold impact on computation time: vector quantization itself will take longer to compute, and so will the actual hand-eye calibration as more data are used. All data sets shown can arguably be considered to be relatively stable from a codebook size of 2000 on for all sequences, which corresponds to about 5.5% (*ART1*), 10% (*ART3*, *ART4*), and 11% (*ART2*) of the total number of relative movements before pre-selection, or about 22% to 29% of the relative movements after pre-selection.

Note that the residuals for *ART1* are considerably higher than those of the other sequences. This is not due to the data selection or hand-eye calibration algorithms used, but, as mentioned before, inherent in the data, which does not contain sufficient information for an accurate calibration due to only small rotational movement (i. e., rotation axes are more concentrated and the amount of rotation is smaller than in the other data sets).

4.4 Rotation Angle Threshold

The second data selection parameter that influences hand-eye calibration accuracy determines how much data is discarded during the pre-selection stage due to small rotation angles. For the experimental evaluation in this section the automatic threshold computation algorithm described in Sect. 3.5 was used. During the experiments codebook size was fixed for each data set, while the percentage of relative movements that are removed by automatic threshold computation was variable. The codebook sizes used are 1200 (*ART1*), 1000 (*ART2*), 900 (*ART3*), and 700 (*ART4*). Residual error plots dependent on the fraction of data left after pre-selection according to the rotation angle are shown in Fig. 13; a value of 1 on the horizontal axis is equivalent to 100% of the data used, i. e., no pre-selection at all, while 0 would indicate that no data was left after pre-selection. As the latter case is pointless, the evaluation was only done up to a fraction where enough data was left to allow for hand-eye calibration, namely 7%.

The problem when removing relative movements based on the rotation angle alone is that an unknown amount of these movements may be suited quite well for hand-eye calibration in terms of non-parallel rotation axes, so there will always be a trade-off. This issue has been addressed before in Sects. 3.6 and 3.7, where data selection algorithms have been presented that do not require a threshold for pre-selection, as none is done. Whether this results in a higher calibration accuracy than applying a rotation angle threshold is evaluated in Sect. 4.5.

As shown in the histograms in Fig. 9, the data sets contain a considerable number of movements having relatively small rotation angle. Therefore, judging based solely on rotation angle, what would be expected is that the plots in Fig. 13 are u-shaped, i. e., when no movements are removed from the data set, the residual error should be high because unsuitable data are used for calibration and skew the result. On the other end, when most movements have been removed, the calibration residual should increase due to the very small amount of noisy data that is used, making hand-eye calibration more sensitive to erroneous movements.

While the increasing residual error at the lower end can be observed in all plots in Fig. 13, the increase at the right end is visible only for the *ART3* and *ART4* data sets.

In general, a value of 20% to 40% for the data to be left after pre-selection with respect to the rotation angle is a relatively good choice for automatic threshold computation in most cases. If possible, smaller values are preferable to higher ones, because the computation time of vector quantization depends on the size of the data sets after pre-selection.

Method	Translation		Rotation	
consecutive movements	19.9 mm	49.8%	2.89°	14.7%
3D VQ, normalized axes, automatic threshold	5.67 mm	15.8%	1.53°	10.3%
2D VQ, polar coordinates, automatic threshold	5.43 mm	15.3%	1.51°	10.3%
3D VQ, axis/angle, no threshold	5.75 mm	15.8%	1.49°	10.1%
3D VQ, axis/angle, automatic threshold	5.75 mm	16.0%	1.44°	10.1%
4D VQ, quaternions, no threshold	5.98 mm	16.4%	1.52°	10.2%
4D VQ, quaternions, automatic threshold	5.61 mm	15.7%	1.43°	10.0%

Table 2: Comparison of different data selection methods, sequence *ART1*. Parameters: codebook size 1200, $\delta = 0.2$.

Method	Translation		Rotation	
consecutive movements	4.98 mm	23.9%	0.854°	4.26%
3D VQ, normalized axes, automatic threshold	1.98 mm	10.4%	0.626°	3.63%
2D VQ, polar coordinates, automatic threshold	3.12 mm	15.6%	0.797°	4.12%
3D VQ, axis/angle, no threshold	1.52 mm	8.09%	0.563°	3.46%
3D VQ, axis/angle, automatic threshold	3.37 mm	16.6%	0.777°	4.07%
4D VQ, quaternions, no threshold	2.11 mm	11.0%	0.662°	3.73%
4D VQ, quaternions, automatic threshold	2.21 mm	11.5%	0.652°	3.70%

Table 3: Comparison of different data selection methods, sequence *ART2*. Parameters: codebook size 1000, $\delta = 0.5$.

4.5 Comparison of Data Selection Methods

In this section a comparison of the performance of the various vector quantization based data selection methods is presented. As has been implied that data selection is essential if continuous movements are used, the results when using relative movements between consecutive poses (i. e., without any data selection) are shown as well. The following methods were compared:

consecutive movements: relative movements between consecutive poses are used. No data selection is done. These results are presented in order to show how much can be gained by data selection when continuously recorded sequences are used.

3D VQ, normalized axes: data selection using normalized 3D rotation axes with two DOF as presented in Sect. 3.3.

2D VQ, polar coordinates: data selection based on the polar coordinate representation of normalized rotation axes as presented in Sect. 3.4.

3D VQ, axis/angle: data selection with three DOF based on the axis/angle representation of rotation axes, where the rotation angle is encoded as the norm of the axis. This method has been presented in Sect. 3.6.

4D VQ, quaternions: data selection with three DOF based on the quaternion representation of rotations as shown in Sect. 3.7.

In case of '**3D VQ, normalized axes**' and '**2D VQ, polar coordinates**' rotation angle thresholds were computed automatically using the method presented in Sect. 3.5. Obviously, this is not applicable for the '**consecutive movements**' experiments.

Note that for the methods '**3D VQ, axis/angle**' and '**4D VQ, quaternions**' no pre-selection of the data with respect to small rotation angles is necessary as the angle is handled implicitly during vector quantization. The residuals for this case are denoted by '**no threshold**'. However, the removal of relative movements with small rotation angles can nevertheless be done, and the results using automatically computed thresholds are also shown.

The tables show absolute and relative residual errors for rotation and translation. In case of translation, the residuals were computed using (11) for absolute errors and (12) for relative ones. The relative rotation error was computed based on quaternions using (13). In order to get an impression of the order of magnitude of the absolute rotational error, it has been decided to show these in degrees rather than to give the absolute quaternionic residual. For this purpose the absolute residuals have been computed from the rotation angle (in axis/angle representation) of the residual rotation matrix given by (17). This value is highly correlated to the quaternionic residual. Note, however, that due to the different rotation representations, there will be slight deviations in some cases when absolute and relative residuals from different experiments are compared.

The calibration results are shown in Tables 2 to 5.

Method	Translation		Rotation	
consecutive movements	9.94 mm	37.3%	2.61°	7.61%
3D VQ, normalized axes, automatic threshold	2.80 mm	12.3%	1.05°	4.57%
2D VQ, polar coordinates, automatic threshold	4.01 mm	16.7%	1.28°	4.95%
3D VQ, axis/angle, no threshold	3.96 mm	16.5%	1.48°	5.28%
3D VQ, axis/angle, automatic threshold	3.69 mm	15.6%	1.36°	5.09%
4D VQ, quaternions, no threshold	4.12 mm	17.1%	1.55°	5.40%
4D VQ, quaternions, automatic threshold	3.94 mm	16.5%	1.42°	5.20%

Table 4: Comparison of different data selection methods, sequence *ART3*. Parameters: codebook size 1000, $\delta = 0.45$.

Method	Translation		Rotation	
consecutive movements	8.34 mm	17.7%	1.10°	4.59%
3D VQ, normalized axes, automatic threshold	1.67 mm	4.87%	0.555°	2.93%
2D VQ, polar coordinates, automatic threshold	1.68 mm	4.89%	0.561°	2.95%
3D VQ, axis/angle, no threshold	1.68 mm	4.85%	0.578°	2.99%
3D VQ, axis/angle, automatic threshold	1.65 mm	4.85%	0.544°	2.90%
4D VQ, quaternions, no threshold	1.75 mm	4.96%	0.587°	3.01%
4D VQ, quaternions, automatic threshold	1.65 mm	4.86%	0.541°	2.90%

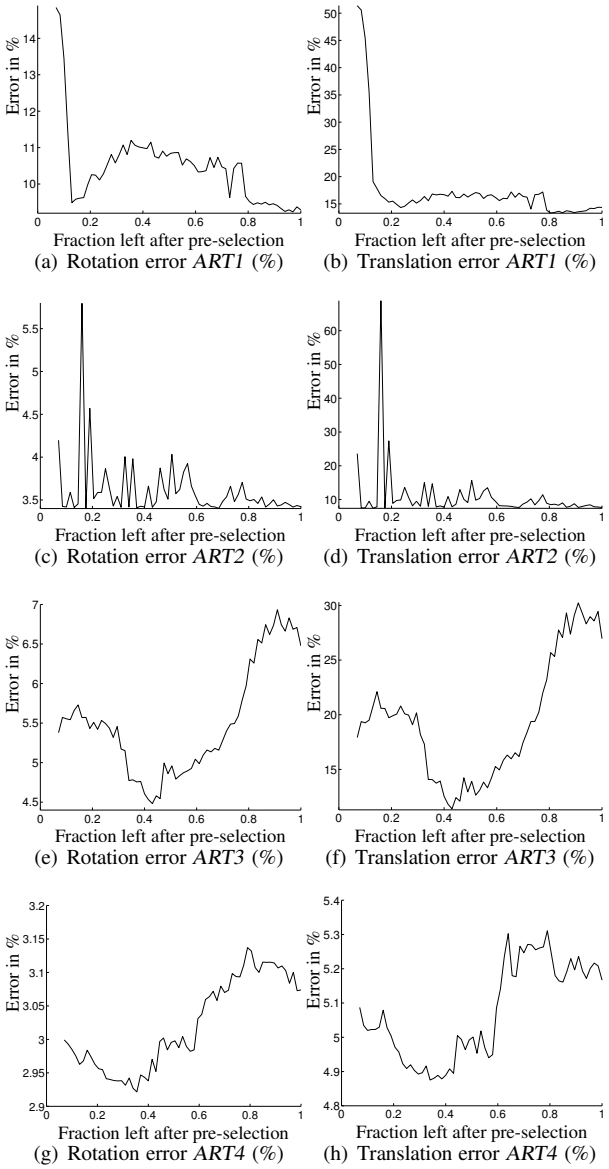
Table 5: Comparison of different data selection methods, sequence *ART4*. Parameters: codebook size 700, $\delta = 0.35$.

As predicted, using consecutive movements is always the worst case, with residual errors that render the calibration result totally useless in most cases. It can be observed that the data selection method used has virtually no influence on the residual errors in rotation. This result is as expected, as the rotation matrix can always be computed, even for movements which are not general enough, while this is not true for translation. Recommending a single data selection method based on the results presented here is not easy, as there is no single method that consistently gives the best results. However, the following insights are to be gained: Using any data selection is better than doing nothing, as all methods perform much better than using consecutive movements directly. We believe that the method of choice is using normalized rotation axes with a pre-selection according to the rotation angle with an automatically computed threshold ('**3D VQ, normalized axes**'), because it is consistently good and ranked in the top three in all experiments.

The results using rotation representations where the rotation angle is implicitly coded, like quaternions and axis/angle ('**3D VQ, axis/angle, no threshold**' and '**4D VQ, quaternions, no threshold**'), did not perform as well as the others, with the exception of the *ART2* experiment. Results improved when the automatic threshold computation was done, removing movements with small rotation angles.

With the exception of *ART1*, using polar coordinates was inferior to most other methods. As can be seen in Table 5, which shows the results for the *ART4* experiment, if the data set is suited well for calibration, i. e., the coverage of rotation axes is high and there is a sufficient amount of rotation angles close to 90°, it makes virtually no difference which data selection method is used. The difference between the best and worst results (not taking into account using consecutive movements for obvious reasons) is only 0.1 mm, and the difference between the best result and the recommended method using normalized rotation axes is only 0.02 mm.

The experiments were run on a computer with an Intel P4, 2.6 GHz CPU. While the implementation was not particularly optimized for speed, we want to give at least a rough guideline regarding the computation times that can be expected for data selection and hand-eye calibration. Using consecutive movements is obviously the fastest way to calibrate, as no data selection is performed, and takes about 50 msec. For the remaining methods, computation time depends mainly on the amount of data during vector quantization and its dimensionality. Not surprisingly, the fastest algorithm is the one based on polar coordinates, as only a 2D quantization has to be done; computation times varied between 5.8 sec (*ART4*) and 11 sec (*ART1*). Accordingly, using quaternions and no threshold (i. e., 4D quantization and all data) is always the slowest option, with computation times between 10 sec (*ART4*) and 31 sec (*ART1*). This is similar for the axis/angle method using no threshold, where the time required is only slightly less, namely 9.5 sec (*ART2*) to 26 sec (*ART1*). When using a pre-selection



(a) Rotation error *ART1* (%) (b) Translation error *ART1* (%)
(c) Rotation error *ART2* (%) (d) Translation error *ART2* (%)
(e) Rotation error *ART3* (%) (f) Translation error *ART3* (%)
(g) Rotation error *ART4* (%) (h) Translation error *ART4* (%)

Figure 13: Mean relative and absolute errors in rotation and translation dependent on the fraction of relative movements left after pre-selection with respect to the rotation angle for the data sets *ART1*, *ART2*, *ART3*, and *ART4*.

with an automatically computed threshold, the computation times of these two algorithms are very similar to the ones for the normalized rotation axes method, which varies between 6.1 sec (*ART4*) and 12 sec (*ART1*).

4.6 Comparison to Manual Selection

There are obviously situations, where acquiring well-suited data at manually selected positions is less convenient than in others. In a classic setup with a robot manipulator arm, the robot can easily be programmed to move to defined positions that are used for calibration. However, in setups like hand-eye calibration of a camera mounted on an endoscope or a stereo camera system used in Augmented Reality, the device is manually controlled by the user, and getting good data is not that simple any more. Therefore, two topics will be addressed in this section, namely:

1. How well does hand-eye calibration based on continuously recorded sequences perform compared to calibration done the classic way using a small number of poses recorded at manually selected distinct positions that are well-suited for hand-eye calibration?

2. Can calibration accuracy be increased by running a data selection on data sets with manually selected positions, which is good already when consecutive poses are used?

Data Set	Translation		Rotation	
ART5	2.31 mm	10.2%	0.556°	3.46%
ART6	1.65 mm	8.02%	0.556°	3.47%
ART2	1.98 mm	10.4%	0.626°	3.63%
ART7	1.85 mm	8.41%	0.892°	4.34%
ART3	2.80 mm	12.3%	1.05°	4.57%
ART8	2.08 mm	5.49%	0.672°	3.21%
ART4	1.67 mm	4.87%	0.555°	2.93%

Table 6: Comparison of the continuously recorded sequences *ART2*, *ART3*, and *ART4* to data sets using the same camera-endoscope configuration recorded at manually selected positions with a small number of poses. It can be observed that a manual selection of positions is not always superior to the conveniently recorded continuous sequences.

Table 6 shows calibration results for data sets consisting of about 20 poses recorded at manually selected positions (calibrated the classic way), which were described in Sect. 4.2, and compares them to the corresponding continuously recorded data (calibrated using data selection with normalized rotation axes and automatic threshold computation). The camera-endoscope configuration (i. e., actual hand-eye transformation) of *ART5* and *ART6* is the same as for the continuously recorded data set *ART2*. Likewise, *ART7* has the same configuration as *ART3*, and *ART8* the same as *ART4*.

Comparing *ART5* and *ART6* on one side to *ART2* on the other shows that the manual selection of positions while recording data is not always better than the conveniently recorded continuous sequence. Particularly the translation estimate of *ART2* is considerably better than that of *ART5*. For *ART7* and *ART3* the result for the manually selected positions (*ART7*) is much better than that of the continuously recorded *ART3*. The situation

	Consecutive Frames				Data Selection			
Data Set	**Translation**		**Rotation**		**Translation**		**Rotation**	
ART5	2.31 mm	10.2%	0.556°	3.46%	2.13 mm	9.76%	0.552°	3.45%
ART6	1.65 mm	8.02%	0.556°	3.47%	1.32 mm	7.11%	0.538°	3.41%
ART7	1.85 mm	8.41%	0.892°	4.34%	1.73 mm	8.14%	0.865°	4.30%
ART8	2.08 mm	5.49%	0.672°	3.21%	1.86 mm	5.15%	0.518°	2.80%

Table 7: Comparison between the classic hand-eye calibration method using consecutive poses (left) and the method using the data selection proposed in this work (right). Clearly, an additional data selection improves the calibration accuracy on these data sets, which contain only a small number of poses recorded at manually selected distinct positions.

for the last set *ART8* and *ART4* shows again that a continuously recorded sequence can lead to better results than a manual selection of the positions for recording.

As mentioned before, it is obvious that much depends on the data set itself and the poses contained in it. When there is not sufficient information available, the calibration results are not good as well. Of course, in practice this may be more often the case for continuously recorded sequences, as the user usually does not choose the positions as carefully as in the case where the data are acquired at distinct positions.

Even though the data selection methods have been developed for continuously recorded sequences, where they are essential, the question remains whether applying them to the data sets containing only a small number of poses at manually selected positions would lead to even better calibration results. Therefore, a comparison between the classic method using consecutive poses and the method using the data selection proposed in this paper is presented in Table 7. Again, the data selection method chosen for this comparison was vector quantization of normalized rotation axes with automatic threshold computation. In all cases, the calibration with data selection was superior to the one using consecutive poses as is usually done for hand-eye calibration. Except for *ART8*, where the absolute residual error in rotation changed by $0.154°$, which is a 23% decrease compared to calibration without data selection, the rotation residuals did not change by much. The translational residual error, however, improved considerably for all data sets, which is a characteristic effect of using well-suited data for hand-eye calibration.

The results make clear that even when the user tries to perform a small amount of well-described calibration movements as accurately as possible, using data selection will still increase calibration accuracy.

5 Conclusion

This paper proposes new data selection methods that can improve the accuracy of hand-eye calibration in many cases, and make it possible in the first place in situations where a manual selection of positions is inconvenient or even impossible. Examples include areas other than the classic setup with a robot manipulator arm and a camera, like calibration of a camera mounted on an endoscope, where the "hand" data are provided by an optical tracking system as used in the experiments' section of this paper, or applications like self-calibration of a rigid stereo camera system, which appear to be unrelated to hand-eye calibration at a first glance, but nevertheless can be solved using algorithms adopted from hand-eye calibration. The result of the data selection algorithm is a data set that is well-suited for hand-eye calibration, because it removes relative movements with small rotation angles and selects those movements where the rotation axes are different. Data selection as presented in this paper is based on applying a clustering algorithm on the data. For this purpose, we proposed to use vector quantization, but in general any clustering algorithm is suitable.

A variety of methods is proposed, which differ in the dimensionality (2D, 3D, 4D) of the vector quantization compared to the degrees of freedom (two or three), and whether an automatically computed threshold or no threshold at all is used for incorporation of the rotation angle. The methods using no threshold are based on 3D and 4D vector quantization using the axis/angle or quaternion representation of rotations, respectively. They are a trade-off between the non-parallelism criterion for the rotation axes and the fact that for movements with small rotation angles the axis is not well-defined, while the former methods remove movements with small angles in a pre-processing step and use only the differences in the rotation axes as a selection criterion.

The performance of the proposed vector quantization based data selection methods was evaluated using data obtained from an optical tracking system (hand) and an endoscopic camera (eye) that was calibrated using a calibration pattern.

Firstly, the parameters that influence hand-eye calibration accuracy were evaluated: The codebook size of the quantizer and the thresholds used for pre-selection of movements with respect to their rotation angle. It has been found that the fluctuations of the residual error for

small codebook sizes are relatively high and become less for increasing codebook size. Based on the experimental results it is recommended to use about 10% of the total number of relative movements as the codebook size, corresponding to about 20% to 30% of the movements left after pre-selection.

Before vector quantization, a pre-selection step is performed that discards movements with small rotation angles from the data, as these are not suitable for hand-eye calibration. The data remaining after this pre-selection has a high influence on the computation time required during vector quantization. These experiments were performed for fixed codebook sizes (but different ones for each data set), while the percentage of movements that are removed by automatic threshold computation was varied. High residual errors can be expected when either most movements have been removed or when all movements have been left in the data used for calibration. In most cases a value of 20% to 40% for the data remaining after pre-selection is a good choice for automatic threshold computation.

Next, the different data selection methods proposed in this paper were compared. The new methods are based on vector quantization and differ in the parameterization used for representing 3D rotation (normalized rotation axes, polar coordinates, axis/angle, quaternions) and in the way the data pre-selection with respect to the rotation angle is done (automatically computed thresholds or no thresholds). These methods were compared to using consecutive movements, i. e., using the movements as they are, without any data selection. As predicted, using consecutive movements is always the worst case, with residual errors that render the calibration results totally useless in most cases.

As expected, the data selection method used has virtually no influence on the residual errors in rotation, only the translational residuals differ. For obtaining the best results regarding accuracy, it is recommended to apply the vector quantization based data selection using normalized rotation axes. It has to be taken into account, however, that much depends on the data set itself, i. e., when the information contained in the movements is not general enough in terms of different rotation axes and high rotation angles, there is no way to obtain good calibration results, no matter which data selection or hand-eye calibration algorithm is used.

It has also been found that even in situations where calibration data is acquired using a small number of manually selected positions, applying an additional data selection can improve the accuracy of hand-eye calibration over the classic approach using consecutive movements. Even when the user is asked to perform a small number of well-described calibration movements, applying data selection on the acquired data will still result in improved accuracy of the hand-eye transformation.

A Appendix: 3D Rotation Parameterization

As the data selection algorithms presented in Sect. 3 make extensive use of various representations of 3D rotations, a brief overview over the commonly used ones is given here. A rotation in 3D is usually given by a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ having the property that the column (and row) vectors are orthonormal:

$$\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} = \boldsymbol{I}_{3\times 3}, \quad \det(\boldsymbol{R}) = 1 \quad , \tag{18}$$

Therefore, a rotation matrix $\boldsymbol{R}$ has nine elements but only three degrees of freedom. The set of all these matrices forms the rotation group SO(3).

A.1 Cardan and Euler Angles

The Cardan and Euler angle representations are discussed only shortly, as they are not employed for data selection due to problems inherent to them. However, as both representations are widely used, we do not want to omit them completely.

To obtain the Cardan angle representation an arbitrary rotation matrix $\boldsymbol{R}$ is decomposed into a product of three rotations by the angles α, β, and γ about the x-, y-, and z-axis of the coordinate system. In contrast to the Euler angle representation the Cardan angles α, β, and γ are defined with respect to the axes of the original coordinate system.

In the Euler angle representation, an arbitrary rotation matrix $\boldsymbol{R}$ is also decomposed into a product of three rotations by the angles ϕ, ψ, and φ, where

- ϕ defines a rotation about the z-axis of the original coordinate system,
- ψ defines a rotation about the x'-axis, which is the image of the x-axis of the original coordinate system after the first rotation,
- φ defines a rotation about the z''-axis, which is the image of the z-axis of the original coordinate system after the previous two rotations have been computed.

Cardan and Euler angles are probably the most well known parameterizations for rotations in 3D. These two representations sometimes get mixed up in literature, but usually the conclusions drawn for Cardan and Euler angles stay the same. One of the main drawbacks is that since matrix multiplication is not commutative, the Cardan/Euler angle representation is not unique, meaning that a permutation of the order of the rotations about the

coordinate-system axes yields different Cardan/Euler angles. Probably the most important drawback of this parameterizations is the existence of so-called *gimbal lock* singularities, where one degree of freedom is lost, i. e., two of the three angles belong to the same degree of freedom. For a more detailed discussion see Watt and Watt (1992).

A.2 Axis/Angle

An arbitrary rotation $\boldsymbol{R}$ can be represented as a rotation about *one* axis $\boldsymbol{r} \in \mathbb{R}^3$ by the angle θ. This will be denoted here as *Axis/Angle representation*. Since only the direction of the rotation axis $\boldsymbol{r}$ is of importance, $\boldsymbol{r}$ has only two degrees of freedom and thus can be normalized to one. Hence, axis and angle can be combined into a single vector $\boldsymbol{\omega}$ with three degrees of freedom, its direction giving the rotation axis and its length the rotation angle, thus making it a minimal parameterization:

$$\boldsymbol{\omega} = \theta \boldsymbol{r}, \text{ and for } \theta \neq 0\text{: } \theta = |\boldsymbol{\omega}|,\ \boldsymbol{r} = \frac{\boldsymbol{\omega}}{|\boldsymbol{\omega}|} \,. \tag{19}$$

Computing a rotation matrix $\boldsymbol{R}$ from $\boldsymbol{\omega}$ can be done using Rodrigues' formula (Hartley and Zisserman, 2003; Faugeras, 1993):

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{I}_{3\times3} + \frac{\sin\theta}{\theta}[\boldsymbol{\omega}]_\times + \frac{1-\cos\theta}{\theta^2}[\boldsymbol{\omega}]_\times^2 \\ &= \boldsymbol{I}_{3\times3} + \sin\theta[\boldsymbol{r}]_\times + (1-\cos\theta)[\boldsymbol{r}]_\times^2 \quad , \end{aligned} \tag{20}$$

where $[\boldsymbol{x}]_\times$ denotes the skew-symmetric matrix that represents the outer vector product of $\boldsymbol{x}$ and $\boldsymbol{y}$ as a matrix multiplication:

$$\boldsymbol{x} \times \boldsymbol{y} = [\boldsymbol{x}]_\times \boldsymbol{y} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \boldsymbol{y} \quad . \tag{21}$$

The computation of axis and angle from a rotation matrix $\boldsymbol{R}$ is done as follows: Eigen-decomposition of $\boldsymbol{R}$ yields the three Eigen-values 1 and $\cos\theta \pm \mathrm{i}\sin\theta$. The axis $\boldsymbol{r}$ is the Eigen-vector corresponding to the Eigen-value 1. The angle θ is calculated from one of the remaining Eigen-values. Note that the axis/angle representation is not unique: a rotation about an axis $\boldsymbol{r}$ by an angle θ is the same as a rotation about the axis $-\boldsymbol{r}$ by the angle $2\pi-\theta$. Therefore, one has to check the consistency of the direction of the axis and the angle, which can be done by inserting both into equation (20). Another problem arises for a rotation angle of $0°$, i. e., if $\boldsymbol{R} = \boldsymbol{I}_{3\times3}$. In this case all three Eigen-values are equal to one, which results in a non-unique rotation axis. This is obvious, since for an angle of $0°$ no rotation is done at all, meaning that the axis can obviously be chosen arbitrarily.

A.3 Quaternions

Quaternions are numbers[3] that are in a certain sense similar to complex numbers: Instead of only one imaginary part, quaternions have three of them. The concept of quaternions was introduced by Sir William Rowan Hamilton and presented to the Royal Irish Academy in 1843 (Hamilton, 1844, 1847, 1848). The set of quaternions is usually denoted as $\mathbb{H}$. Unit quaternions form the special unitary group SU(2), which can be represented as all complex unitary 2×2 matrices having determinant one. Since SU(2) is a double cover of the special orthogonal group SO(3), there exist two quaternions for each rotation matrix. More details on quaternions can be found in Kuipers (1999); Conway and Smith (2003); Faugeras (1993).

A quaternion $\boldsymbol{q}$ is defined as follows:

$$\boldsymbol{q} = q_\mathrm{r} + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k},\ \ q_\mathrm{r}, q_1, q_2, q_3 \in \mathbb{R} \,, \tag{22}$$

where q_r is the real part and q_1, q_2, q_3 are the imaginary parts. Multiplication and summation are done component-wise, with

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1 \quad , \tag{23}$$

which is equivalent to

$$\begin{aligned} \mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = -1 \quad , \\ \mathrm{ij} = -\mathrm{ji} = \mathrm{k},\ \mathrm{jk} = -\mathrm{kj} = \mathrm{i},\ \mathrm{ki} = -\mathrm{ik} = \mathrm{j} \,. \end{aligned} \tag{24}$$

A quaternion is often written as a 4-tuple

$$\boldsymbol{q} = (q_\mathrm{r}, q_1, q_2, q_3) \text{ or } \boldsymbol{q} = (q_\mathrm{r}, \boldsymbol{q}_\mathrm{im}) \quad , \tag{25}$$

where $\boldsymbol{q}_\mathrm{im}$ is a 3-vector containing the imaginary parts. In contrast to complex numbers, the commutative law of multiplication is *not* valid (cf. (24)), i. e.,

$$\exists \boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathbb{H},\ \text{where } \boldsymbol{q}_1\boldsymbol{q}_2 \neq \boldsymbol{q}_2\boldsymbol{q}_1 \quad . \tag{26}$$

Similar to complex numbers, a conjugate quaternion is defined as

$$\boldsymbol{q}^* = q_\mathrm{r} - q_1\mathrm{i} - q_2\mathrm{j} - q_3\mathrm{k} \quad . \tag{27}$$

The norm of a quaternion $\boldsymbol{q}$ is given by

$$|\boldsymbol{q}| = \sqrt{\boldsymbol{q}\boldsymbol{q}^*} = \sqrt{\boldsymbol{q}^*\boldsymbol{q}} = \sqrt{{q_\mathrm{r}}^2 + q_1^2 + q_2^2 + q_3^2} \,. \tag{28}$$

The multiplicative inverse of $\boldsymbol{q}$ is

$$\boldsymbol{q}^{-1} = \frac{1}{\boldsymbol{q}\boldsymbol{q}^*}\,\boldsymbol{q}^* \quad . \tag{29}$$

[3] i. e., they form one of the four existing normed division algebras; the others are the real and complex numbers, and the Octonions (Conway and Smith, 2003; Baez, 2001). The latter have 7 complex parts and are neither commutative nor associative w. r. t. multiplication.

Hence, for a unit quaternion ($|\boldsymbol{q}| = 1$), the inverse of multiplication equals the conjugate, i. e., $\boldsymbol{q}^{-1} = \boldsymbol{q}^*$.

Just as the multiplication of two unit complex numbers defines a rotation in two dimensions, a multiplication of two unit quaternions yields a rotation in 3D. Let $\boldsymbol{p}$ be a 3D point to be rotated, $\boldsymbol{r}$ a rotation axis with $|\boldsymbol{r}| = 1$, and θ the angle of rotation about this axis. Define the following two quaternions:

$$\begin{aligned} \boldsymbol{q} &= \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\cdot\boldsymbol{r}\right) \quad , \\ \boldsymbol{p}' &= (0, \boldsymbol{p}) \quad . \end{aligned} \tag{30}$$

Then

$$\boldsymbol{p}'_{\text{rot}} = \boldsymbol{q}\,\boldsymbol{p}'\,\boldsymbol{q}^{-1} = \boldsymbol{q}\,\boldsymbol{p}'\,\boldsymbol{q}^* \tag{31}$$

since $\boldsymbol{q}$ is a unit quaternion. $\boldsymbol{p}'_{\text{rot}}$ is the quaternion corresponding to the rotated point.

Since a quaternion representing a rotation is computed from axis and angle, it is also not unique, because the two quaternions $\boldsymbol{q}_1 = \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\cdot\boldsymbol{r}\right)$ and $\boldsymbol{q}_2 = \left(\cos\frac{2\pi-\theta}{2}, \sin\frac{2\pi-\theta}{2}\cdot(-\boldsymbol{r})\right) = \left(-\cos\frac{\theta}{2}, -\sin\frac{\theta}{2}\cdot\boldsymbol{r}\right)$ define the same rotation. Which of the two quaternions is used does not matter, but one has to be careful when measuring the distance of two rotations (e. g., for describing rotation residual errors) by the distance between quaternions. However, in contrast to the axis/angle representation, where $\boldsymbol{R} = \boldsymbol{I}_{3\times3}$ results in an undefined rotation axis $\boldsymbol{r}$, the corresponding quaternion is defined and equals 1 (i. e., $(1, 0, 0, 0)$).

The computation of a quaternion from a rotation matrix is done using the axis/angle representation as shown in (30). The computation of a rotation matrix $\boldsymbol{R}$ from a quaternion can be done as follows (Faugeras, 1993):

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{pmatrix} , \tag{32}$$

where

$$\begin{aligned} \boldsymbol{r}_1 &= \begin{pmatrix} {q_\text{r}}^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1q_2 + q_\text{r}q_3) \\ 2(q_1q_3 - q_\text{r}q_2) \end{pmatrix} , \\ \boldsymbol{r}_2 &= \begin{pmatrix} 2(q_1q_2 - q_\text{r}q_3) \\ {q_\text{r}}^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_2q_3 + q_\text{r}q_1) \end{pmatrix} , \\ \boldsymbol{r}_3 &= \begin{pmatrix} 2(q_1q_3 + q_\text{r}q_2) \\ 2(q_2q_3 - q_\text{r}q_1) \\ {q_\text{r}}^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} . \end{aligned} \tag{33}$$

A.4 Discussion

The different representations for rotation matrices introduced in this section model 3×3 rotation matrices, which have nine elements but only three degrees of freedom, with less than nine parameters.

Due to the many disadvantages of Euler and Cardan angle representations, using these should be avoided whenever possible. Axis/Angle is a minimal parameterization having the drawback that for rotations with small angles the rotation axis is not well-defined. This problem does not occur when using unit quaternions; however, quaternions are a non-minimal parameterization, because they have four elements with three degrees of freedom. Both axis/angle and quaternions are non-unique, i. e., there are always two different representations for the same rotation. However, this causes only slight problems in practice, which are not comparable to the non-uniqueness of Cardan and Euler angle representations. Therefore, due to their advantages, the rotation representation of choice for estimating rotation is either axis/angle or quaternions, depending on the application.

An interesting result supporting this conclusion is based on the fact that quaternions as well as axis/angle representation are a so-called *fair parameterization* of 3D rotations, while Cardan and Euler angles are not (Hornegger and Tomasi, 1999). A parameterization is fair if it does not introduce more numerical sensitivity than is inherent to the problem itself, which is guaranteed if any rigid transformation of the space to be parameterized results in an orthogonal transformation of the parameters.

Acknowledgments

The *ART* data sets used in the experiments were provided by F. Vogt, SFB 603, TP B6.

References

Andreff, N. (1997). Towards the Embedding of On-Line Hand-Eye Calibration into Visual Servoing. In *IEEE/RSJ/INRIA Workshop On New Trends in Image-based Robot Servoing*, pages 64–70.

Andreff, N., Horaud, R., and Espiau, B. (1999). On-line Hand-Eye Calibration. In *Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99)*, pages 430–436, Ottawa.

Andreff, N., Horaud, R., and Espiau, B. (2001). Robot Hand-Eye Calibration Using Structure from Motion. *International Journal of Robotics Research*, 20:228–248.

Aron, M., Simon, G., and Berger, M.-O. (2004). Handling Uncertain Sensor Data in Vision-Based Camera Tracking. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 58–67. IEEE Computer Society.

Baez, J. C. (2001). The Octonions. *Bulletin of the American Mathematical Society*, 39(2):145–205.

Chen, H. (1991). A Screw Motion Approach to Uniqueness Analysis of Head-Eye Geometry. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 145–151, Maui, Hawaii. IEEE Computer Society Press.

Chou, J. C. K. and Kamel, M. (1991). Finding the Position and Orientation of a Sensor on a Robot Manipulator Using Quaternions. *International Journal of Robotics Research*, 10(3):240–254.

Conway, J. H. and Smith, D. A. (2003). *On Quaternions and Octonions: Their Geometry, Arithmetic, and Symmetry*. A K Peters, Ltd.

Daniilidis, K. (1999). Hand-Eye Calibration Using Dual Quaternions. *International Journal of Robotics Research*, 18(3):286–298.

Daniilidis, K. (2001). Using the Algebra of Dual Quaternions for Motion Alignment. In Sommer, G., editor, *Geometric Computing with Clifford Algebras*, chapter 20, pages 489–500. Springer-Verlag.

Dornaika, F. and Chung, R. (2003). Stereo Geometry from 3D Ego-Motion Streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(2):308–323.

Fassi, I. and Legnani, G. (2005). Hand to sensor calibration: A geometrical interpretation of the matrix equation AX=XB. *Journal of Robotic Systems*, 22(9):497–506.

Faugeras, O. (1993). *Three–Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA.

Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images, The Laws that Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, Cambridge, Massachusetts.

Hamilton, W. R. (1844). On a New Species of Imaginary Quantities Connected with a Theory of Quaternions. *Proceedings of the Royal Irish Academy*, 2:424–434.

Hamilton, W. R. (1847). On Quaternions. *Proceedings of the Royal Irish Academy*, 3:1–16.

Hamilton, W. R. (1848). Researches Respecting Quaternions: First Series. *Transactions of the Royal Irish Academy*, 21:199–296.

Hartley, R. I. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2nd edition.

Horaud, R. and Dornaika, F. (1995). Hand-Eye Calibration. *International Journal of Robotics Research*, 14(3):195–210.

Hornegger, J. and Tomasi, C. (1999). Representation Issues in the ML Estimation of Camera Motion. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 640–647, Corfu, Greece. IEEE Computer Society Press.

Kuipers, J. B. (1999). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press. Fifth printing, and first paperback printing 2002.

Linde, Y., Buzo, A., and Gray, R. (1980). An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, 28(1):84–95.

Luong, Q.-T. and Faugeras, O. (1993). Self–Calibration of a Stereo Rig from Unknown Camera Motions and Point Correspondences. Technical Report 2014, INRIA.

Luong, Q.-T. and Faugeras, O. (2001). Self–Calibration of a Stereo Rig from Unknown Camera Motions and Point Correspondences. In Gruen, A. and Huang, T., editors, *Calibration and Orientation of Cameras in Computer Vision*, pages 195–229. Springer-Verlag, Berlin, Heidelberg.

Schmidt, J. (2006). *3-D Reconstruction and Stereo Self-Calibration for Augmented Reality*. Logos Verlag, Berlin.

Schmidt, J., Vogt, F., and Niemann, H. (2003). Robust Hand-Eye Calibration of an Endoscopic Surgery Robot Using Dual Quaternions. In Michaelis, B. and Krel, G., editors, *Pattern Recognition, Proceedings of the 25th DAGM Symposium*, volume 2781 of *Lecture Notes in Computer Science*, pages 548–556, Berlin, Heidelberg. Springer-Verlag.

Schmidt, J., Vogt, F., and Niemann, H. (2004). Vector Quantization Based Data Selection for Hand-Eye Calibration. In Girod, B., Magnor, M., and Seidel, H.-P., editors, *Vision, Modeling, and Visualization 2004*, pages 21–28, Stanford, USA. Aka/IOS Press, Berlin.

Schmidt, J., Vogt, F., and Niemann, H. (2005). Calibration-Free Hand-Eye Calibration: A Structure-from-Motion Approach. In Kropatsch, W. G., Sablatnig, R., and Hanbury, A., editors, *Pattern Recognition, 27th DAGM Symposium*, volume 3663 of *Lecture*

Notes in Computer Science, pages 67–74. Springer-Verlag, Berlin, Heidelberg, New York.

Shi, F., Wang, J., and Liu, Y. (2005). An Approach to Improve Online Hand-Eye Calibration. In Marques, J. S., de la Blanca, N. P., and Pina, P., editors, *IbPRIA (1)*, volume 3522 of *Lecture Notes in Computer Science*, pages 647–655. Springer.

Shiu, Y. C. and Ahmad, S. (1989). Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form AX = XB. *IEEE Transactions on Robotics and Automation*, 5(3):16–29.

Tsai, R. Y. (1987). 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.

Tsai, R. Y. and Lenz, R. K. (1989). A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358.

Vogt, F. (2006). *Augmented Light Field Visualization and Real-Time Image Enhancement for Computer Assisted Endoscopic Surgery*. Der Andere Verlag, Tönning.

Wang, C. (1992). Extrinsic Calibration of a Vision Sensor Mounted on a Robot. *IEEE Transactions on Robotics and Automation*, 8(2):161–175.

Watt, A. and Watt, M. (1992). *Advanced Animation and Rendering Techniques*. Addison-Wesley.

Zhang, J., Shi, F., and Liu, Y. (2005). An Adaptive Selection of Motion for Online Hand-Eye Calibration. In Zhang, S. and Jarvis, R., editors, *Australian Conference on Artificial Intelligence*, volume 3809 of *Lecture Notes in Computer Science*, pages 520–529. Springer.

Zhang, Z. (1998). A Flexible New Technique for Camera Calibration. Technical Report MSR-TR-98-71, Microsoft Research.

Zhang, Z. (2000). A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Zhao, Z. and Liu, Y. (2006). Hand-Eye Calibration Based on Screw Motions. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 1022–1026. IEEE Computer Society.