HAL
open science

Hand-Eye and Robot-World Calibration by Global Polynomial Optimization

Jan Heller, Didier Henrion, Tomas Pajdla

To cite this version:

Jan Heller, Didier Henrion, Tomas Pajdla. Hand-Eye and Robot-World Calibration by Global Polynomial Optimization. IEEE International Conference on Robotics and Automation (ICRA 2014), May 2014, Hong Kong, China. 18p. hal-00946579

HAL Id: hal-00946579

https://hal.science/hal-00946579

Submitted on 13 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hand-Eye and Robot-World Calibration by Global Polynomial Optimization

Jan Heller[1], Didier Henrion[2,3,1], Tomáš Pajdla[1]

Draft of February 13, 2014

Abstract

The need to relate measurements made by a camera to a different known coordinate system arises in many engineering applications. Historically, it appeared for the first time in the connection with cameras mounted on robotic systems. This problem is commonly known as hand-eye calibration. In this paper, we present several formulations of hand-eye calibration that lead to multivariate polynomial optimization problems. We show that the method of convex linear matrix inequality (LMI) relaxations can be used to effectively solve these problems and to obtain globally optimal solutions. Further, we show that the same approach can be used for the simultaneous hand-eye and robot-world calibration. Finally, we validate the proposed solutions using both synthetic and real datasets.

1 Introduction

Let us suppose a camera has been mounted on the end-effector—the hand—of a robotic manipulator. Further, let's suppose that this hand-eye robotic system has been manipulated into two distinct poses, see Figure 1. Let's denote the transformation from the camera coordinate system in the first pose of the rig to the world coordinate system as $\mathtt{A}'_1$ and the transformation from the second pose as $\mathtt{A}'_2$. Now, we can express the camera's relative movement from the first pose to the second one as

$$\mathtt{A} = \begin{pmatrix} \mathtt{R}_\mathtt{A} & \mathbf{t}_\mathtt{A} \\ \mathbf{0}^\top & 1 \end{pmatrix} = \begin{pmatrix} \mathtt{R}_{\mathtt{A}'_2} & \mathbf{t}_{\mathtt{A}'_2} \\ \mathbf{0}^\top & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathtt{R}_{\mathtt{A}'_1} & \mathbf{t}_{\mathtt{A}'_1} \\ \mathbf{0}^\top & 1 \end{pmatrix} = \mathtt{A}'^{-1}_2 \mathtt{A}'_1,$$

where $\mathtt{R}_\mathtt{A} \in SO(3)$ is a 3×3 rotation matrix and $\mathbf{t}_\mathtt{A} \in \mathbb{R}^3$ is a translation vector. Analogically, relative movement of the robotic end-effector can be described as

$$\mathtt{B} = \begin{pmatrix} \mathtt{R}_\mathtt{B} & \mathbf{t}_\mathtt{B} \\ \mathbf{0}^\top & 1 \end{pmatrix} = \begin{pmatrix} \mathtt{R}_{\mathtt{B}'_2} & \mathbf{t}_{\mathtt{B}'_2} \\ \mathbf{0}^\top & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathtt{R}_{\mathtt{B}'_1} & \mathbf{t}_{\mathtt{B}'_1} \\ \mathbf{0}^\top & 1 \end{pmatrix} = \mathtt{B}'^{-1}_2 \mathtt{B}'_1,$$

[1]Czech Technical University, Faculty of Electrical Engineering, Karlovo náměstí 13, Prague, Czech Republic.

[2]CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse; France.

[3]Université de Toulouse; UPS, INSA, INP, ISAE; UT1, UTM, LAAS; F-31077 Toulouse; France.

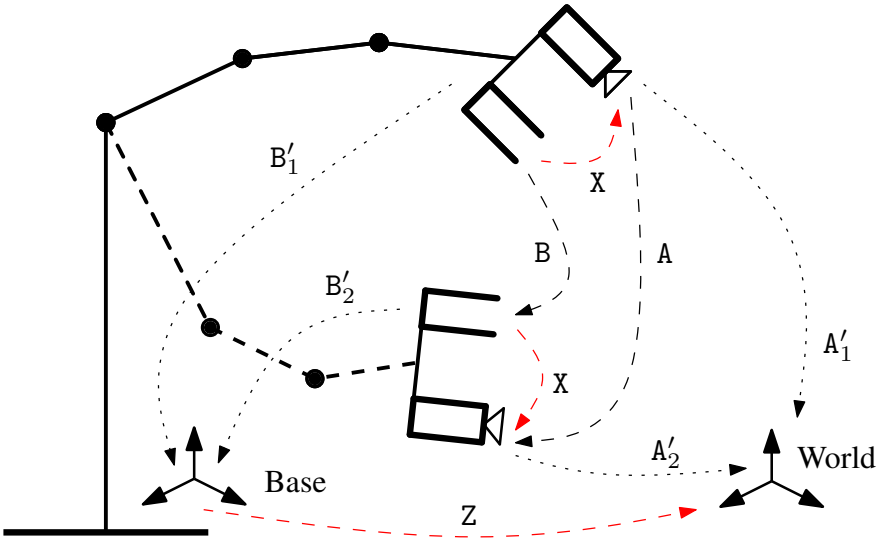


Figure 1: Hand-eye ($\mathtt{X}$) and robot-world ($\mathtt{Z}$) transformations. A hand-eye system is depicted in two distinct poses.

with $\mathtt{B}'_1, \mathtt{B}'_2 \in \mathbb{R}^{4\times 4}$ being the respective transformations from the end-effector's coordinate system to the robot base coordinate system. Assuming we know the transformations $\mathtt{A}$ and $\mathtt{B}$, the problem of finding the rigid transformation $\mathtt{X} \in \mathbb{R}^{4\times 4}$ from the coordinate system of end-effector to the coordinate system connected with the camera can be expressed analytically using the following kinematic loop:

$$\mathtt{AX} = \mathtt{XB}. \tag{1}$$

All of the earliest researchers investigating this problem realized that System 1 is underdetermined and that two poses are not enough to uniquely determine the transformation $\mathtt{X}$. In [29], Shiu and Ahmad showed that at least two relative motions with non-parallel rotational axes are needed. In practice, several relative motions are executed, leading to a set of matrices $\mathtt{A}_i, \mathtt{B}_i$, $i = 1, \ldots, n$ and to an overdetermined and—unless we can measure $\mathtt{A}_i, \mathtt{B}_i$ with perfect accuracy—noisy system of equations

$$\mathtt{A}_i\mathtt{X} = \mathtt{XB}_i, \quad i = 1, \ldots, n. \tag{2}$$

System 2 can be also expressed as

$$\begin{pmatrix} \mathtt{R}_{\mathtt{A}_i} & \mathbf{t}_{\mathtt{A}_i} \\ \mathbf{0}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathtt{R}_{\mathtt{X}} & \mathbf{t}_{\mathtt{X}} \\ \mathbf{0}^\top & 1 \end{pmatrix} = \begin{pmatrix} \mathtt{R}_{\mathtt{X}} & \mathbf{t}_{\mathtt{X}} \\ \mathbf{0}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathtt{R}_{\mathtt{B}_i} & \mathbf{t}_{\mathtt{B}_i} \\ \mathbf{0}^\top & 1 \end{pmatrix} \tag{3}$$

and further decomposed into a rotational matrix equation and translational vector equation

$$\mathtt{R}_{\mathtt{A}_i}\mathtt{R}_{\mathtt{X}} = \mathtt{R}_{\mathtt{X}}\mathtt{R}_{\mathtt{B}_i}, \tag{4}$$

$$\mathtt{R}_{\mathtt{A}_i}\mathbf{t}_{\mathtt{X}} + \mathbf{t}_{\mathtt{A}_i} = \mathtt{R}_{\mathtt{X}}\mathbf{t}_{\mathtt{B}_i} + \mathbf{t}_{\mathtt{X}}. \tag{5}$$

Notice that Equation 4 does not depend on the unknown translation $\mathbf{t}_{\mathtt{X}}$. Once rotation $\mathtt{R}_{\mathtt{X}}$ is known, Equation 5 leads to a system of linear equations in $\mathbf{t}_{\mathtt{X}}$ and the translation

can be easily determined using the tools of linear algebra. This fact has been exploited by all of the earliest solution strategies.

In [29], Shiu and Ahmad proposed the first solution to the hand-eye calibration problem formulated as Equation 2. They used the angle-axis parametrization of the group of rotations $SO(3)$. The angle-axis parametrization was also used in [33, 34, 25]. Chou and Kamel [7] proposed to parametrize Equation 4 by quaternions. In [15], Horaud and Dornaika also used quaternions to recover $\mathtt{R_X}$ as a minimizer of the same objective function as [25].

In [6], Chen employed the screw motion theory to investigate the necessary and sufficient conditions for the solutions of Equation 2. The screw motion theory is based on the fact that a general homogeneous transformation can be accomplished by means of a translation along a unique axis and a rotation about the same axis. This is known as *Chasles theorem* [5] and such a description of a rigid motion is known as a *screw*. Chen concluded that in the case of noisy inputs the computation of $\mathtt{R_X}$ and $\mathbf{t_X}$ should not be decoupled, because otherwise the generality of the result would be negatively affected. In [8, 9], Daniilidis and Bayro-Corrochano showed how to parametrize Equation 2 using the algebraic counterparts of screws—dual quaternions—and how to solve for rotational and translational parts of $\mathtt{X}$ simultaneously. Another simultaneous solution of Equation 2 was proposed by Andreff *et al.* in [4] using the Kronecker product. However, their solution needs to be followed by a orthogonalization of the rotational part $\mathtt{R_X}$.

Several researchers also proposed iterative solutions to Equation 2. Zhuang and Shiu [37] proposed an iterative non-linear method to minimize function $\sum_{i=1}^{n} \|\mathtt{A}_i\mathtt{X} - \mathtt{X}\mathtt{B}_i\|^2$ to simultaneously estimate the rotational and translational parts of $\mathtt{X}$. As a part of [15], Horaud and Dornaika also proposed a simultaneous iterative method based of quaternions and Levenberg-Marquardt non-linear optimization [21]. They observed that the method performed well only after introducing two *ad hoc* selected scaling factors. Both methods need to be provided with an initial solution estimates and depending on the accuracy of the estimates may not converge to the global optima. In [35], Zhao suggested an iterative method that is guaranteed to converge globally optimally based on Second Order Cone Programming. However, his iterative method does not enforce the orthogonality of the rotation matrix and thus suffers from the same rotational error propagation as does the method [4]. Strobl and Hirzinger [31] suggested a novel metric on $SE(3)$ and an iterative method based on a parametrization of a stochastic model. However, for their method to perform optimally, some prior information on data noise characteristics is needed.

In [36], Zhuang *et al.* extended the hand-eye calibration problem to also include calibration of the robot-world transformation $\mathtt{Z}$, see Figure 1. Their method uses quaternion rotation representation to solve an equation analogical to the Equation 2,

$$\mathtt{A}'_i\mathtt{X} = \mathtt{Z}\mathtt{B}'_i,\ i = 1, \ldots, m, \tag{6}$$

where $\mathtt{Z} \in \mathbb{R}^{4\times 4}$ represents the transformation from the robot base coordinate system to the world coordinate system. In this case, the kinematic loop is closed using the absolute camera and robot poses $\mathtt{A}'_i, \mathtt{B}'_i$. Dornaika and Horaud [10] suggested a different solution, also based on quaternions. In [3], Li *et al.* proposed two different solutions based on Kronecker product and dual quaternions, analogous to the solutions to the hand-eye calibration problem in [4] and [9], respectively.

Recently, several authors [28, 16, 13, 12, 27] proposed hand-eye calibration methods that use image measurements directly, instead of using them to compute matrices $\mathtt{A}'_i$ as a pre-step. While such an approach can indeed eliminate the errors resulting from explicit computation of matrices $\mathtt{A}'_i$, it is only applicable in situations where the image correspondences are available, which may not always be the case.

In this paper, we propose a set of iterative methods to solve hand-eye and robot-world calibration based on Equations 2 and 6 that do not require initial estimates and provide globally optimal solutions in L_2-norm. Further, these methods solve for the rotational and translational part simultaneously. This is achieved by formulating the hand-eye and robot-world calibration problems as multivariate polynomial optimization problems over semialgebraic sets and by solving them using the method of convex linear matrix inequality (LMI) relaxations [17]. Besides providing global optimizers, one of the main advantages of LMI relaxations method is the fact that it naturally handles algebraic as well as semialgebraic constraints that are inherent to the most rotation parametrizations.

First, we review the method of convex LMI relaxations. Next, we formulate a set of hand-eye and robot-world calibration polynomial optimization problems. Finally in the experimental section, we provide a few implementation details and show the performance of the proposed solution using both synthetic and real data calibration scenarios.

2 Convex LMI Relaxations

Let $p_i(\mathbf{x})$, $i = 0, 1, \ldots, \ell$ be scalar multivariate polynomials in $\mathbf{x} = (x_1, x_2, \ldots, x_m)^\top \in \mathbb{R}^m$, *i.e.*, $p_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$. Formally, the problem of multivariate polynomial optimization can be stated as follows:

Problem 1 *(Multivariate polynomial optimization)*

$$\begin{array}{rl} \textit{minimize} & p_0(\mathbf{x}) \\ \textit{subject to} & p_i(\mathbf{x}) \geq 0,\ i = 1, \ldots, \ell, \\ \textit{where} & \mathbf{x} = (x_1, x_2, \ldots, x_m)^\top \in \mathbb{R}^m, \\ & p_0(\mathbf{x}), p_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]. \end{array}$$

Typically, this is a non-convex problem with many local minima. In theory, polynomial optimization problems can be handled using tools of elementary calculus, however, finding the global minimizer $\mathbf{x}^*$ is—in general—an NP-hard problem [23]. In [17], Lasserre cast the problem polynomial optimization over finite-dimensional semialgebraic set $S = \{\mathbf{x} \in \mathbb{R}^m \,|\, p_i(\mathbf{x}) \geq 0, i = 1, \ldots, \ell\}$ as the problem of linear optimization over the infinite-dimensional set of probability measures supported on S. In [26], Putinar proved that such probability measures can be represented via sequences $\mathbf{y} = (y_\alpha)_{\alpha \in \mathbb{N}}$ of its moments. Using this result, Lasserre showed that by truncating these sequences one can construct a hierarchy of convex relaxations $\mathcal{P}_1, \mathcal{P}_2, \ldots$ that produces a monotonically non-decreasing sequence of lower bounds on Problem 1 that converge to the global minimum. He also showed, that the series of the respective global optimizers $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots$

of problems $\mathcal{P}_1, \mathcal{P}_2, \ldots$ asymptotically converges to $\mathbf{x}^*$, $\lim_{i\to\infty} \mathbf{x}_i^* = \mathbf{x}^*$, and that under mild conditions global optimality of a relaxation can be detected and the global minimizers can be extracted by linear algebra from the solutions of the relaxation. Practically, $(\mathbf{x}_i^*)_{i\in\mathbb{N}}$ converges to $\mathbf{x}^*$ in finitely many steps, *i.e.*, there exists $j \in \mathbb{N}$, such that $\mathbf{x}_j^* = \mathbf{x}^*$. Problems for which the finite convergence does not occur are in some sense degenerate and exceptional [24]. The hierarchy $\mathcal{P}_1, \mathcal{P}_2, \ldots$ is sometimes also called *Lasserre's LMI hierarchy.*

Now, we will show how to formulate the relaxations $\mathcal{P}_1, \mathcal{P}_2, \ldots$ as semidefinite programs (SDP) solvable by any convenient SDP solver. In order to do that, let us define the linearization operator $L_{\mathbf{y}} \colon \mathbb{R}[\mathbf{x}] \to \mathbb{R}[\mathbf{y}]$ (also called *Riesz functional*) which takes a polynomial $p(\mathbf{x})$ and substitutes a new variable $y_{k_1 k_2 \ldots k_m} \in \mathbb{R}$ for every monomial $x_1^{k_1} x_2^{k_2} \ldots x_m^{k_m}$. First, The LMI relaxation $\mathcal{P}_\delta$ of order δ is built by linearizing all monomials $x_1^{k_1} x_2^{k_2} \ldots x_m^{k_m}$ of the objective function p_0 up to degree 2δ, *i.e.*, $k_1 + k_2 + \cdots + k_m \leq 2\delta$. If the objective function contains monomials of a higher degree, one has to start with a relaxation of a higher order. Next, let $\mathbf{v}_\delta(\mathbf{x})$ be the vector of all monomials up to degree δ. The semialgebraic set S is relaxed by introducing ℓ LMI constraints $L_{\mathbf{y}}(p_i(\mathbf{x})\mathbf{v}_{\delta-1}(\mathbf{x})\mathbf{v}_{\delta-1}(\mathbf{x})^\top) \succeq 0$[1]. Finally, we add the so-called LMI moment matrix constraint $L_{\mathbf{y}}(\mathbf{v}_\delta(\mathbf{x})\mathbf{v}_\delta(\mathbf{x})^\top) \succeq 0$. Formally, the LMI relaxation $\mathcal{P}_\delta$ of order δ can be written as

Problem 2 *(the LMI relaxation $\mathcal{P}_\delta$ of order δ)*

$$\begin{array}{ll} \textit{minimize} & L_{\mathbf{y}}(p_0(\mathbf{x})) \\ \textit{subject to} & L_{\mathbf{y}}(p_i(\mathbf{x})\mathbf{v}_{\delta-1}(\mathbf{x})\mathbf{v}_{\delta-1}(\mathbf{x})^\top \succeq 0,\ i = 1, \ldots, \ell, \\ & L_{\mathbf{y}}(\mathbf{v}_\delta(\mathbf{x})\mathbf{v}_\delta(\mathbf{x})^\top) \succeq 0. \end{array}$$

Since there are exactly $d = \binom{m+2\delta}{m}$ monomials in $\mathbf{x} \in \mathbb{R}^m$ up to degree 2δ, SDP Problem 2 will have $\mathbf{y} \in \mathbb{R}^d$ linear variables. See [17] for the technical justification of this procedure.

3 Hand-Eye Calibration

In this section, three formulation of the hand-eye calibration problem are presented. In the first two, we formulate two parametrizations of of the following minimization problem:

$$\min_{\mathtt{X} \in SE(3)} \sum_{i=1}^{n} \left\| \mathtt{A}_i \mathtt{X} - \mathtt{X} \mathtt{B}_i \right\|^2,$$

i.e., the minimization the Frobenius norm

$$\|\mathtt{M}\| = \sqrt{\textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} |m_{ij}|^2} = \sqrt{\mathrm{tr}(\mathtt{M}^\top \mathtt{M})}$$

on the special Euclidean group

$$SE(3) = \left\{ \left(\begin{array}{cc} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{array} \right) \middle| \mathtt{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}.$$

[1] Notation "$\mathtt{M} \succeq 0$" stands for "$\mathtt{M}$ is positive semidefinite matrix".

The third formulation uses the dual quaternion parametrization to minimize the vector L_2-norm

$$\min_{\hat{\mathbf{q}}_{\mathtt{X}}} \sum_{i=1}^{n} \left\| \hat{\mathbf{a}}_i \otimes \hat{\mathbf{q}}_{\mathtt{X}} - \hat{\mathbf{q}}_{\mathtt{X}} \otimes \hat{\mathbf{b}}_i \right\|^2,$$

where $\hat{\mathbf{a}}_i, \hat{\mathbf{b}}_i, \hat{\mathbf{q}}_{\mathtt{X}}$ are the dual quaternion representations of $\mathtt{A}_i$,$\mathtt{B}_i$, and $\mathtt{X}$, respectively, and $\otimes$ is the dual quaternion multiplication.

All three formulations lead to the multivariate polynomial optimization problems. However, for the sake of brevity we will not cite the explicit form of the polynomials. The fact that these formulations are indeed polynomial can be easily checked by any tool for symbolic algebra computation.

3.1 Orthonormal parametrization

As is the case with all linear maps on finite-dimensional vector spaces, a rotation can be always expressed by a matrix $\mathtt{R}$, in this case of size 3×3. Since a rotation maps orthonormal basis of $\mathbb{R}^3$ to another orthonormal basis, the columns $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^3$ of the matrix $\mathtt{R} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$ themselves must form an orthonormal basis. The fact that the columns of $\mathtt{R}$ form a orthonormal basis can be also written as

$$\begin{array}{ll} \mathbf{v}^\top \mathbf{v} = 1, & \mathbf{u}^\top \mathbf{u} = 1, \\ \mathbf{v}^\top \mathbf{u} = 0, & \mathbf{v} \times \mathbf{u} = \mathbf{w}. \end{array}$$

This constitutes 6 constraints on the elements of the matrix $\mathtt{R}$, leaving it with 3 degrees of freedom. Using these constraints, we can parametrize the homogeneous transformation $\mathtt{X}$ as

$$\mathtt{X}(\mathbf{u}, \mathbf{v}, \mathbf{t}) = \begin{pmatrix} \mathtt{R}(\mathbf{u}, \mathbf{v}) & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix},$$

where $\mathtt{R}(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v}, \mathbf{u} \times \mathbf{v})$. This leads to the following parametrization of the hand-eye calibration problem:

Problem 3 *(uvhec method)*

$$\begin{array}{ll} \textit{minimize} & f_1(\mathbf{u}_{\mathtt{X}}, \mathbf{v}_{\mathtt{X}}, \mathbf{t}_{\mathtt{X}}) = \\ & \sum_{i=1}^{n} \left\| \mathtt{A}_i \mathtt{X}(\mathbf{u}_{\mathtt{X}}, \mathbf{v}_{\mathtt{X}}, \mathbf{t}_{\mathtt{X}}) - \mathtt{X}(\mathbf{u}_{\mathtt{X}}, \mathbf{v}_{\mathtt{X}}, \mathbf{t}_{\mathtt{X}}) \mathtt{B}_i \right\|^2 \\ \textit{subject to} & \mathbf{u}_{\mathtt{X}}^\top \mathbf{u}_{\mathtt{X}} = 1, \mathbf{v}_{\mathtt{X}}^\top \mathbf{v}_{\mathtt{X}} = 1, \mathbf{u}_{\mathtt{X}}^\top \mathbf{v}_{\mathtt{X}} = 0. \end{array}$$

The objective function f_1 of Problem 3 is a polynomial function of degree 4 and it is composed of 123 monomials in 9 variables.

3.2 Quaternion parametrization

Quaternions, $\mathbb{Q}$, form a four-dimensional associative normed division algebra over the real numbers. A quaternion $\mathbf{q} \in \mathbb{Q}$ consist of a real part and an imaginary part and is usually

denoted as

$$\mathbf{q} = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k},$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the imaginary units such that $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. As a set, quaternions are equal to $\mathbb{R}^4$ and it is sometimes useful to write them as vectors

$$\mathbf{q} \equiv (q_1, q_2, q_3, q_4)^\top = \left(q_1, \bar{\mathbf{q}}^\top\right)^\top,$$

where $\bar{\mathbf{q}} \in \mathbb{R}^3$ is the imaginary part of the quaternion. Addition of two quaternions $\mathbf{p}, \mathbf{q} \in \mathbb{Q}$, $\mathbf{p} = \left(p_1, \bar{\mathbf{p}}^\top\right)^\top$, $\mathbf{q} = \left(q_1, \bar{\mathbf{q}}^\top\right)^\top$ is equivalent to addition in $\mathbb{R}^4$. Quaternion multiplication, however, does not have a counterpart operation on vector spaces:

$$\mathbf{p} * \mathbf{q} = (p_1 q_1 - \bar{\mathbf{p}}^\top \bar{\mathbf{q}}, (p_1 \bar{\mathbf{q}} + q_1 \bar{\mathbf{p}} + \bar{\mathbf{p}} \times \bar{\mathbf{q}})^\top)^\top.$$

Because the group of unit quaternions with multiplication, modulo the multiplication by -1, is isomorphic to the group of rotations with composition, they can be used to represent rotations. Rotation about axis $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^\top$, $||\boldsymbol{\alpha}|| = 1$, by angle θ is represented by $\mathbf{q} \in \mathbb{Q}$ as

$$\mathbf{q} = \cos\frac{\theta}{2} + (\alpha_1\mathbf{i} + \alpha_2\mathbf{j} + \alpha_3\mathbf{k}) \sin\frac{\theta}{2}.$$

The unity of a quaternion can be expressed using the quaternion conjugate $\mathbf{q}^* = \left(q_1, -\bar{\mathbf{q}}^\top\right)^\top$ as $\mathbf{q}^* * \mathbf{q} = 1$ or using the inner product as $\mathbf{q}^\top \mathbf{q} = 1$. Transformation $\mathtt{X}$ can be parametrized using the unit quaternion $\mathbf{q} = (q_1, q_2, q_3, q_4)$ as

$$\mathtt{X}(\mathbf{q}, \mathbf{t}) = \begin{pmatrix} \mathtt{R}(\mathbf{q}) & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix},$$

where

$$\mathtt{R}(\mathbf{q}) = \begin{pmatrix} q_1^2 + q_2^2 - q_3^2 - 2q_4^2 & 2q_2q_3 - 2q_4q_1 & 2q_2q_4 + 2q_3q_1 \\ 2q_2q_3 + 2q_4q_1 & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2q_3q_4 - 2q_2q_1 \\ 2q_2q_4 - 2q_3q_1 & 2q_3q_4 + 2q_2q_1 & q_1^2 - q_2^2 - q_3^2 + 2q_4^2 \end{pmatrix}$$

is the quaternion parametrization of a rotation matrix. This leads to the following parametrization of the hand-eye calibration problem:

Problem 4 *(qhec method)*

$$\begin{array}{ll} \textit{minimize} & f_2(\mathbf{q}_\mathtt{X}, \mathbf{t}_\mathtt{X}) = \\ & \quad \sum_{i=1}^n \left\|\mathtt{A}_i\mathtt{X}(\mathbf{q}_\mathtt{X}, \mathbf{t}_\mathtt{X}) - \mathtt{X}(\mathbf{q}_\mathtt{X}, \mathbf{t}_\mathtt{X})\mathtt{B}_i\right\|^2 \\ \textit{subject to} & \mathbf{q}_\mathtt{X}^\top \mathbf{q}_\mathtt{X} = 1, \\ & q_{\mathtt{X}1} \geq 0. \end{array}$$

The objective function f_2 of Problem 4 is a polynomial function of degree 4 and it is composed of 85 monomials in 7 variables. Since the unit quaternions are a double cover of $SO(3)$, function f_2 has at least two global optima $f_2(\mathbf{q}_\mathtt{X}^*, \mathbf{t}_\mathtt{X}^*) = f_2(-\mathbf{q}_\mathtt{X}^*, \mathbf{t}_\mathtt{X}^*)$. To help the SDP solver, we add the semialgebraic constraint $q_{\mathtt{X}1} \geq 0$ to Problem 4 to eliminate one of the global optima in the majority of cases, that is when $q_{\mathtt{X}1}^* \neq 0$. The algebraic constraint $\mathbf{q}_\mathtt{X}^\top \mathbf{q}_\mathtt{X} = 1$ enforces the unity of the resulting quaternion.

3.3 Dual Quaternion parametrization

Dual quaternions are the algebraic counterparts of screws. They form a Clifford algebra; a dual quaternion $\hat{\mathbf{q}} \in \mathbb{H}$ can be represented in the form

$$\hat{\mathbf{q}} = \mathbf{q} + \epsilon \mathbf{q}',$$

where $\mathbf{q}, \mathbf{q}' \in \mathbb{Q}$ and ϵ is the dual unit, $\epsilon\epsilon = 0$, that commutes with every element of the algebra. It is also convenient to write dual quaternions as vectors $(\mathbf{q}^\top, \mathbf{q}'^\top)^\top$, since the set of dual quaternions is equal to $\mathbb{R}^8$. Addition of two dual quaternions $\hat{\mathbf{p}}, \hat{\mathbf{q}} \in \mathbb{H}$, $\hat{\mathbf{p}} = \left(\mathbf{p}^\top, \mathbf{p}'^\top\right)^\top$, $\hat{\mathbf{q}} = \left(\mathbf{q}^\top, \mathbf{q}'^\top\right)^\top$ is equivalent to addition in $\mathbb{R}^8$. Multiplication can be expressed using quaternion multiplication as

$$\hat{\mathbf{p}} \otimes \hat{\mathbf{q}} = ((\mathbf{p} * \mathbf{q})^\top, (\mathbf{p} * \mathbf{q}' + \mathbf{p}' * \mathbf{q})^\top)^\top.$$

Similar to the way rotations in $\mathbb{R}^3$ can be represented by the quaternions of unit length, rigid motions in $\mathbb{R}^3$ can be represented by unit dual quaternions [22]: rotation represented by a quaternion $\mathbf{p} \in \mathbb{Q}$ followed by translation $\mathbf{t} \in \mathbb{R}^3$ is represented by the dual quaternion

$$\hat{\mathbf{q}}(\mathbf{p}, \mathbf{t}) = (\mathbf{p}^\top, ((0, \tfrac{1}{2}\mathbf{t}^\top)^\top * \mathbf{p})^\top)^\top.$$

Unity of a dual quaternion $\hat{\mathbf{q}}$ can be expressed using its conjugate $\hat{\mathbf{q}}^* = ({\mathbf{q}^*}^\top, {\mathbf{q}'^*}^\top)^\top$ as $\hat{\mathbf{q}}^* \otimes \hat{\mathbf{q}} = 1$ or using the quaternion parts as

$$\mathbf{q}^\top \mathbf{q} = 1 \quad \text{and} \quad q_1 q_5 + q_2 q_6 + q_3 q_7 + q_4 q_8 = 0.$$

Let $\hat{\mathbf{a}}_i$, $\hat{\mathbf{b}}_i$ be the dual quaternion representation of transformations $\mathtt{A}_i$, $\mathtt{B}_i$, respectively. Since Equation 2 can be expressed using dual quaternions multiplications as

$$\hat{\mathbf{a}}_i \otimes \hat{\mathbf{q}}_{\mathtt{X}} = \hat{\mathbf{q}}_{\mathtt{X}} \otimes \hat{\mathbf{b}}_i, \tag{7}$$

where $\hat{\mathbf{q}}_{\mathtt{X}}$ is the dual quaternion representation of the transformation $\mathtt{X}$, we can recover $\mathtt{X}$ using the following formulation:

Problem 5 *(dqhec method)*

$$\begin{array}{ll} \textit{minimize} & f_3(\hat{\mathbf{q}}_{\mathtt{X}}) = \sum_{i=1}^{n} \left\| \hat{\mathbf{a}}_i \otimes \hat{\mathbf{q}}_{\mathtt{X}} - \hat{\mathbf{q}}_{\mathtt{X}} \otimes \hat{\mathbf{b}}_i \right\|^2 \\ \textit{subject to} & \mathbf{q}_{\mathtt{X}}^\top \mathbf{q}_{\mathtt{X}} = 1, \\ & q_{\mathtt{X}1} q_{\mathtt{X}5} + q_{\mathtt{X}2} q_{\mathtt{X}6} + q_{\mathtt{X}3} q_{\mathtt{X}7} + q_{\mathtt{X}4} q_{\mathtt{X}8} = 0. \\ & q_{\mathtt{X}1} \geq 0. \end{array}$$

The objective function f_3 of Problem 5 is a polynomial function of degree 4 and it is composed of 177 monomials in 8 variables. Again, additional constraint $q_{\mathtt{X}1} \geq 0$ is added to eliminate one of the two global optima in most of the cases. Since we are using dual quaternions directly, we have to take care of the orientation ambiguity of the rotational quaternions $\mathbf{a}_i, \mathbf{b}_i$ when converting matrices $\mathtt{A}_i$, $\mathtt{B}_i$ into dual quaternions $\hat{\mathbf{a}}_i$, $\hat{\mathbf{b}}_i$. This is due

to the fact that even though quaternions $\mathbf{a}_i, -\mathbf{a}_i$ and $\mathbf{b}_i, -\mathbf{b}_i$ represent the same rotation, the sign matters when Equation 2 is expressed using dual quaternions, *i.e.*,

$$\hat{\mathbf{a}}_i(\mathbf{a}_i, \mathbf{t}_{\mathtt{A}_i}) \otimes \hat{\mathbf{q}}_{\mathtt{X}} - \hat{\mathbf{q}}_{\mathtt{X}} \otimes \hat{\mathbf{b}}_i(\mathbf{b}_i, \mathbf{t}_{\mathtt{B}_i}) \neq \hat{\mathbf{a}}_i(-\mathbf{a}_i, \mathbf{t}_{\mathtt{A}_i}) \otimes \hat{\mathbf{q}}_{\mathtt{X}} - \hat{\mathbf{q}}_{\mathtt{X}} \otimes \hat{\mathbf{b}}_i(\mathbf{b}_i, \mathbf{t}_{\mathtt{B}_i}).$$

To check the "compatibility" of the quaternions $\mathbf{a}_i$ and $\mathbf{b}_i$, the *screw congruence theorem* [6] can be used. It provides a necessary condition for the solution of Equation 7:

$$\hat{\mathbf{a}}_i \otimes \hat{\mathbf{q}}_{\mathtt{X}} = \hat{\mathbf{q}}_{\mathtt{X}} \otimes \hat{\mathbf{b}}_i \Rightarrow (a_1, a_1')^\top = (b_1, b_1')^\top.$$

Dual quaternions were first applied to the hand-eye calibration problem by Daniilidis and Bayro-Corrochano in [8, 9]. Their method, however, does not make use of all of the information in the camera and robot motions. Since the method assumes that $(a_1, a_1')^\top = (b_1, b_1')^\top$, it only uses the imaginary parts $\bar{\mathbf{a}}_i$, $\bar{\mathbf{a}}_i'$, $\bar{\mathbf{b}}_i$, $\bar{\mathbf{b}}_i'$. In [20], Malti and Barreto proposed a method based on dual quaternions that uses also the real components, however their solution does not solve for rotation and translation simultaneously.

4 Simultaneous Hand-Eye and Robot-World Calibration

Due to the apparent similarity of Equations 2 and 6, analogies to Problems 3, 4, and 5 can be formulated for the simultaneous hand-eye and robot-world calibration problem.

4.1 Orthonormal parametrization

Problem 6 is based on orthonormal parametrization of Equation 6. The objective function f_4 is a polynomial function of degree 4 and it is composed of 280 monomials in 18 variables.

Problem 6 *(uvherwc method)*

$$\begin{array}{ll} \textit{minimize} & f_4(\mathbf{u}_{\mathtt{X}}, \mathbf{v}_{\mathtt{X}}, \mathbf{t}_{\mathtt{X}}, \mathbf{u}_{\mathtt{Z}}, \mathbf{v}_{\mathtt{Z}}, \mathbf{t}_{\mathtt{Z}}) = \\ & \quad \sum_{i=1}^{m} \|\mathtt{A}_i'\mathtt{X}(\mathbf{u}_{\mathtt{X}}, \mathbf{v}_{\mathtt{X}}, \mathbf{t}_{\mathtt{X}}) - \mathtt{Z}(\mathbf{u}_{\mathtt{Z}}, \mathbf{v}_{\mathtt{Z}}, \mathbf{t}_{\mathtt{Z}})\mathtt{B}_i'\|^2 \\ \textit{subject to} & \mathbf{u}_{\mathtt{X}}^\top \mathbf{u}_{\mathtt{X}} = 1, \mathbf{v}_{\mathtt{X}}^\top \mathbf{v}_{\mathtt{X}} = 1, \mathbf{u}_{\mathtt{X}}^\top \mathbf{v}_{\mathtt{X}} = 0, \\ & \mathbf{u}_{\mathtt{Z}}^\top \mathbf{u}_{\mathtt{Z}} = 1, \mathbf{v}_{\mathtt{Z}}^\top \mathbf{v}_{\mathtt{Z}} = 1, \mathbf{u}_{\mathtt{Z}}^\top \mathbf{v}_{\mathtt{Z}} = 0. \end{array}$$

4.2 Quaternion parametrization

Problem 7 is based on quaternion parametrization of Equation 6. The objective function f_5 is a polynomial function of degree 4 and is composed of 209 monomials in 14 variables.

Problem 7 *(qherwc method)*

$$
\begin{aligned}
\textit{minimize} \quad & f_5(\mathbf{q}_{\mathtt{X}}, \mathbf{t}_{\mathtt{X}}, \mathbf{q}_{\mathtt{Z}}, \mathbf{t}_{\mathtt{Z}}) = \\
& \sum_{i=1}^{m} \left\| \mathtt{A}'_i \mathtt{X}(\mathbf{q}_{\mathtt{X}}, \mathbf{t}_{\mathtt{X}}) - \mathtt{Z}(\mathbf{q}_{\mathtt{Z}}, \mathbf{t}_{\mathtt{Z}}) \mathtt{B}'_i \right\|^2 \\
\textit{subject to} \quad & \mathbf{q}_{\mathtt{X}}^\top \mathbf{q}_{\mathtt{X}} = 1,\ q_{\mathtt{X}1} \geq 0, \\
& \mathbf{q}_{\mathtt{Z}}^\top \mathbf{q}_{\mathtt{Z}} = 1,\ q_{\mathtt{Z}1} \geq 0.
\end{aligned}
$$

4.3 Dual Quaternion parametrization

Problem 8 is based on quaternion parametrization of Equation 6. The objective function f_6 is a polynomial function of degree 2 and it is composed of 112 monomials in 16 variables.

Problem 8 *(dqherwc method)*

$$
\begin{aligned}
\textit{minimize} \quad & f_6(\hat{\mathbf{q}}_{\mathtt{X}}) = \sum_{i=1}^{m} \left\| \hat{\mathbf{a}}'_i \otimes \hat{\mathbf{q}}_{\mathtt{X}} - \hat{\mathbf{q}}_{\mathtt{Z}} \otimes \hat{\mathbf{b}}'_i \right\|^2 \\
\textit{subject to} \quad & \mathbf{q}_{\mathtt{X}}^\top \mathbf{q}_{\mathtt{X}} = 1,\ q_{\mathtt{X}1} \geq 0, \mathbf{q}_{\mathtt{Z}}^\top \mathbf{q}_{\mathtt{Z}} = 1,\ q_{\mathtt{Z}1} \geq 0, \\
& q_{\mathtt{X}1} q_{\mathtt{X}5} + q_{\mathtt{X}2} q_{\mathtt{X}6} + q_{\mathtt{X}3} q_{\mathtt{X}7} + q_{\mathtt{X}4} q_{\mathtt{X}8} = 0, \\
& q_{\mathtt{Z}1} q_{\mathtt{Z}5} + q_{\mathtt{Z}2} q_{\mathtt{Z}6} + q_{\mathtt{Z}3} q_{\mathtt{Z}7} + q_{\mathtt{Z}4} q_{\mathtt{Z}8} = 0.
\end{aligned}
$$

As is the case of Problem 5, we need to take care of the orientation ambiguity of the rotational quaternions $\mathbf{a}_i, \mathbf{b}_i$. This time however, the screw congruence theorem does not apply. The obvious solution it to try all of the 2^m sign combinations and keep the combination with the smallest value of $f_6(\hat{\mathbf{q}}_{\mathtt{X}}^*)$. Running the SDP optimization 2^m times can be quite computationally expensive, so in our experiments we use the dual quaternion method of Li [3]—which needs to be run 2^m times in practice as well, since it suffers from the same quaternion ambiguity problem—to recover the sign combination before running the SDP solver, because it runs faster. The sign problem is also inherent to the method of Dornaika [10].

5 Experiments

In this section we present both synthetic and real data experiments to validate the proposed calibration methods. For the real data experiment we used a Motoman MA1400 serial 6-DOF manipulator with an Asus Xtion Pro sensor rigidly attached to it. The Xtion Pro sensor was equipped with a camera with the resolution of 640×480 pixels. We simulated the same setup in the synthetic experiment in order to better judge the result of the experiment with real data.

5.1 Implementation

We implemented all 6 calibration methods in MATLAB using GloptiPoly [14], an interface that automatically constructs LMI relaxations of polynomial problems and converts them

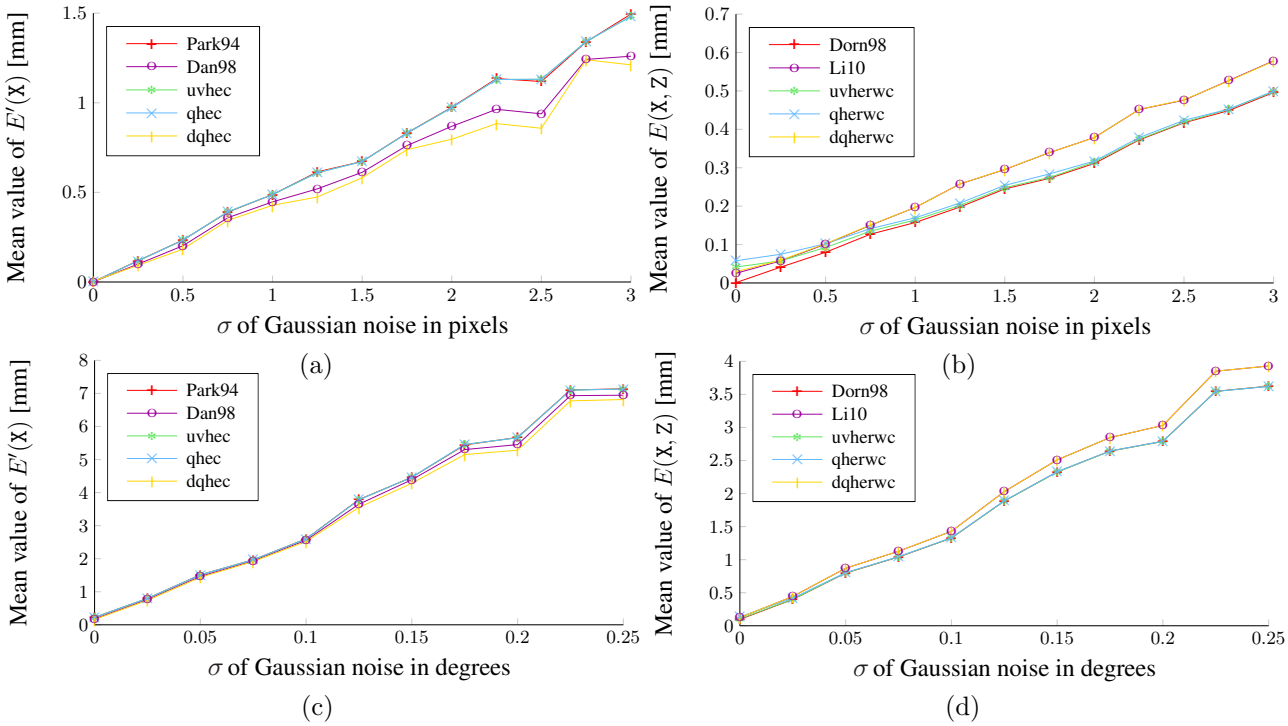


Figure 2: *Synthetic data experiment results.* (a–b) Image noise experiment. (c–d) Joint noise experiment.

into a format understandable by the SDP solver SeDuMi [32]. Further, GloptiPoly can also recover the solution to the original polynomial problem and certify its optimality. Using YALMIP toolbox [19], the SeDuMi format can be further converted into the input formats of several other SDP solvers. In our experiments, we used the LMI relaxations of the second order and SeDuMi 1.3 and MOSEK [1] 7.0 as SDP solvers. GloptiPoly certified that global optimum has been reached by all the methods for all the problem instances encountered during the experiments. The following table sums up the relevant information on the proposed methods:

Method	Variables	Degree	Monomials	Moments	Time[1](s)	Time[2](s)
uvhec	9	4	124	715	1.45	0.45
qhec	7	4	85	330	0.48	0.29
dqhec	8	4	177	495	0.99	0.46
uvherwc	18	4	280	7315	953.76	60.89
qherwc	14	4	209	3060	68.38	7.91
dqherwc	16	2	112	4845	309.25	16.85

[1]SeDuMi,[2]MOSEK

To obtain the timings, we used a 3.5GHz Intel Core i7 based desktop computer running 64-bit Linux. The presented times include time spent in the SDP solver as well as GloptiPoly and YALMIP overheads. To gauge the sizes of the SDP problems involved, the column "Moments" specifies the number of moments (dimension of vector $\mathbf{y}$ in Problem 2) in the second order LMI relaxation for the respective method. Note, that even though the objective functions f_i depend on n or m, *i.e.*, on the number of relative motions or poses, the number of monomials does not. This means that the sizes of the LMI relaxations and the sizes of the resulting SDP problems also do not depend on n or m and are constant.

As a pre-step to all of the methods, we scale the translations $\mathbf{t}_{\mathtt{A}_i}, \mathbf{t}_{\mathtt{B}_i}$ by the factor of $\alpha = \max_i \{\|\mathbf{t}_{\mathtt{A}_i}\|, \|\mathbf{t}_{\mathtt{B}_i}\|\}$, so that the length of the longest translation is 1. This helps with the convergence of the SDP solver and removes the influence of the chosen physical units on the accuracy of the result. In cases where $\mathbf{t}_{\mathtt{X}}$ and $\mathbf{t}_{\mathtt{Z}}$ are explicitly optimized, we add one or two more constraints $\mathbf{t}_{\mathtt{X}}^\top \mathbf{t}_{\mathtt{X}} \leq 2$, $\mathbf{t}_{\mathtt{Z}}^\top \mathbf{t}_{\mathtt{Z}} \leq 10$. These constraints result from the way our experiments were constructed and are not technically necessary, however, they help the SDP solvers to further speed-up the converge. In the experiments, we set the SeDuMi parameter `eps` $= 10^{-20}$ and the MOSEK parameters `MSK_DPAR_INTPNT_CO_TOL_{P|D}FEAS` $= 10^{-20}$.

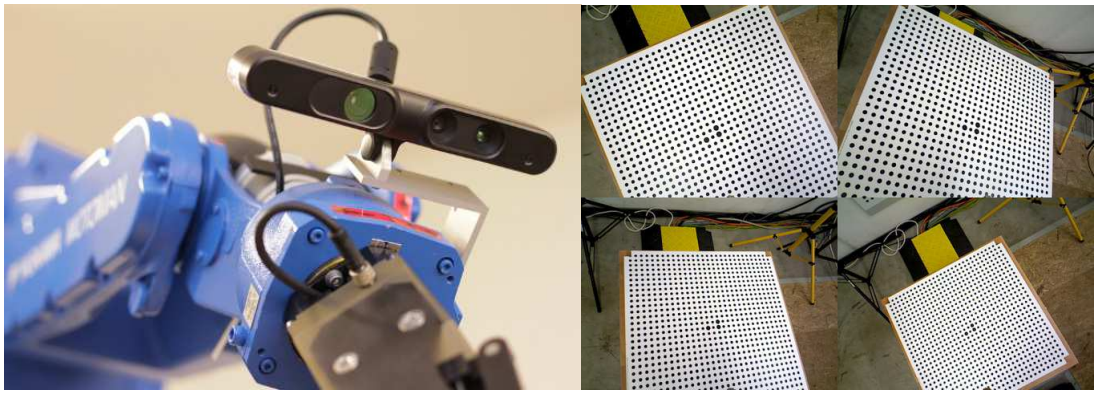
The source code as well as examples are available at

`http://cmp.felk.cvut.cz/~hellej1/mpherwc/`

5.2 Synthetic Experiments

In the synthetic experiments, we investigated the influence of both image and joint noises on the calibration accuracy. We placed a virtual planar calibration target consisting of a grid of 16×16 known points in front of a simulated MA1400 serial manipulator. The distance between the points was set to 12.5 mm in each direction, making the calibration target 200×200 mm in size.

To create a calibration task, we started with 9 camera poses $\mathtt{A}'_i$ randomly generated onto half-sphere of radius of ~ 30 mm oriented as to face approximately the center of the calibration target. To simulate the typical situation when the camera faces approximately the same direction as the robot's end effector, a hand-eye rotation $\mathtt{R}_{\mathtt{X}}$ was randomly generated so that the rotation corresponded to identity to a up to $5°$ degrees difference in each axis. The translation $\mathbf{t}_{\mathtt{X}}$ was randomly generated to move the camera up to ~ 200 mm away from end effector. The arm poses $\mathtt{B}'_i$ were computed using the camera poses and the hand-eye transformation as $\mathtt{X}^{-1}\mathtt{A}'_i$. Finally, we generated a robot-world transformation $\mathtt{Z}$ using a random rotation and a random translation up to ~ 2000 mm in length and used $\mathtt{Z}$ to transform the grid points from the robot into the world coordinate system. We generated 10 sets of 9 camera positions, 10 hand-eye transformations and 1 robot-world transformation and combined them into 100 calibration tasks. For every calibration task, we also computed all possible relative movements $\mathtt{A}_i$, $\mathtt{B}_i$, $i = 1, \ldots, 36$ to be used by the hand-eye calibration methods. In order to better judge the accuracy the proposed methods, we implemented several hand-eye and robot-world calibration methods to compare them against: *Park94* [25] and *Dan98* [9], as representatives of hand-eye calibration methods, and *Dorn98* [10] and *Li10*, the dual quaternion variant of



(a) (b)

Figure 3: *Real data experiment setup.* (a) Detail of the Xtion Pro sensor and the robot's end effector. (b) Examples of input images.

[3], as representatives of simultaneous hand-eye and robot-world calibration methods.

Image Noise Experiment

To simulate the influence of image noise, we projected the calibration points into 640×480 pixel images using the generated camera poses $\mathtt{A}'_i$ and corrupted the image projections $\mathbf{u}_{ij}$ in each calibration task with Gaussian noise in 13 noise levels, $\sigma \in \langle 0, 3 \rangle$ px in $1/4$ px steps. Finally, we recovered noised camera poses $\mathtt{A}'_i$ using EPnP algorithm [18].

In order to evaluate the performance of the hand-eye and robot-world calibration methods *uvherwc*, *qherwc*, and *dqherwc*, we have sampled the virtual workspace of the robot—a cube of size $70 \times 70 \times 70$ cm around the calibration target—with uniformly distributed $\ell = 9240$ points $\mathbf{Y}_j$. Using these points, measured in the robot coordinate frame, we defined a calibration error measure $E(\mathtt{X}, \mathtt{Z})$ that expresses the error of the calibration as the difference of the positions the workspace points transformed into the camera coordinate frame using the computed transformations $\mathtt{X}$, $\mathtt{Z}$ and the ground truth transformations $\mathtt{X}^{\mathrm{gt}}$, $\mathtt{Z}^{\mathrm{gt}}$. Formally,

$$E(\mathtt{X}, \mathtt{Z}) = \tfrac{1}{9\ell} \sum_{i=1}^{9} \sum_{j=1}^{\ell} \left\| \mathtt{X}\mathtt{B}_i'^{-1}\mathtt{Z}^{-1}\mathtt{Z}^{\mathrm{gt}}\mathbf{Y}_j - \mathtt{X}^{\mathrm{gt}}\mathtt{B}_i'^{-1}\mathbf{Y}_j \right\| .$$

To evaluate the performance of the hand-eye calibration methods *uvhec*, *qhec*, and *dqhec*, we do not have to transform the points $\mathbf{Y}_j$ into the world coordinate frame first, so we defined a slightly simpler 3D error function

$$E'(\mathtt{X}) = \tfrac{1}{36\ell} \sum_{i=1}^{36} \sum_{j=1}^{\ell} \left\| \mathtt{X}\mathtt{B}_i'^{-1}\mathbf{Y}_j - \mathtt{X}^{\mathrm{gt}}\mathtt{B}_i'^{-1}\mathbf{Y}_j \right\| .$$

Figure 2(a) shows the mean of the values $E'(\mathtt{X})$ for all 100 calibration tasks for different methods and image noise levels. It shows the best performance by the method *dqhec* followed by *Dan98* and finally by *Park94*, *uvhec*, and *qhec*, all with the same performance level. In this experiment, we used SeDuMi as the SDP solver.

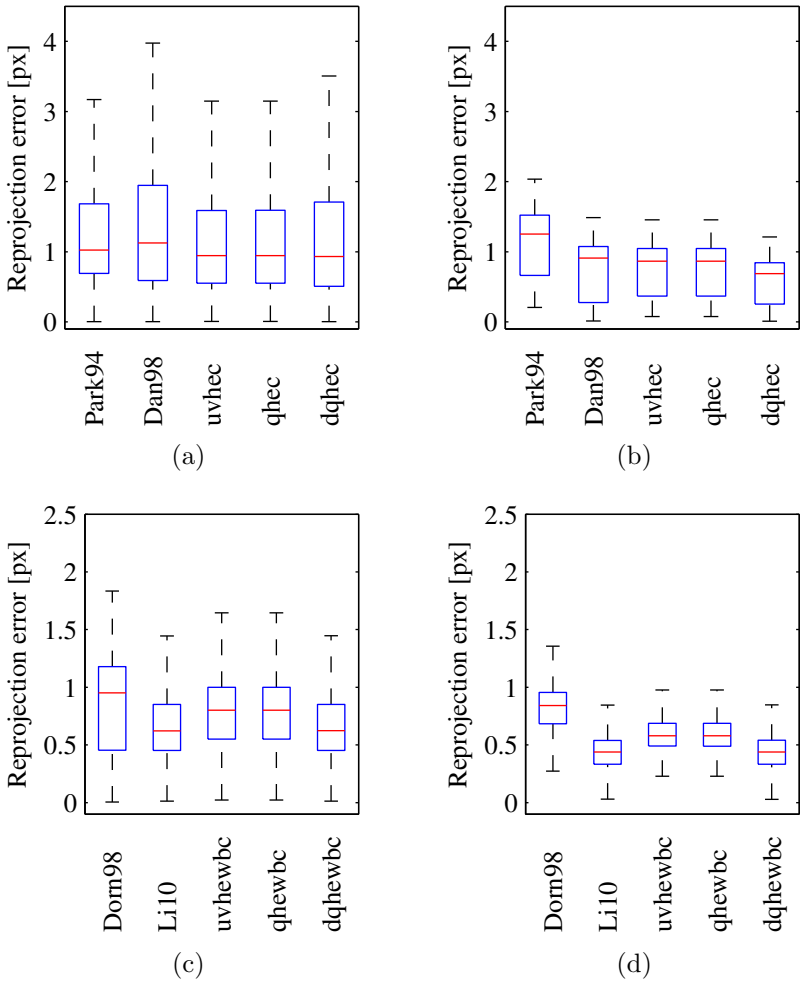


Figure 4: *Real data experiment resuts.* (a) Reprojection errors e'_{ij} of the calibration image set (b) Reprojection errors e'_{ij} of the validation image set. (a) Reprojection errors e_{ij} of the calibration image set (b) Reprojection errors e_{ij} of the validation image set. (the red line marks the median, the edges of the box are the 25th and 75th percentiles)

Figure 2(b) shows the same statistics for the values of $E(\mathtt{X}, \mathtt{Z})$ and suggest quite comparable performance of methods *Dorn98*, *uvhewrc*, and *qhewbc*. This time, the dual quaternion formulations *Li10* and *dqherwc* performed worse that the other methods. In this experiment, we used MOSEK as the SDP solver. This resulted in much faster convergence times, but also in slightly worse accuracy for the lower noise levels.

Joint Noise Experiment

In this experiment, we simulated the performance in the presence of joint noise. We started with the same 100 calibration tasks as in the case of the image noise experiment. Further, we recovered the joint coordinates [30] of the virtual MA1400 manipulator for every pose $\mathtt{B}'_i$ with respect to the Denavit–Hartenberg convention. Next, we corrupted the joint coordinates by random offsets—we used the same offsets for the joint coordinates of

the same task—in 11 noise levels, $\sigma \in \langle 0, 0.25 \rangle$ deg in steps of 0.025 degrees. Finally, we recovered noised poses $\mathtt{B}'_i$ using the forward kinematics. To further simulate the real world conditions, we corrupted projections $\mathbf{u}_{ij}$ by image noise of $\sigma = 0.5$ px for every joint noise level. We evaluated the performance using the same error measures $E(\mathtt{X}, \mathtt{Z})$ and $E'(\mathtt{X})$.

Figures 2(c) and 2(d) show the mean of the values $E'(\mathtt{X})$ and $E'(\mathtt{X}, \mathtt{Z})$. Again, the method *qhec* outperforms its competitors, whereas the dual quaternion formulations lose on the rest in the simultaneous hand-eye and robot-world calibration experiment.

5.3 Real Data Experiment

For this experiment, we used a real MA1400 serial manipulator with a Xtion Pro sensor attached to its 5th link, see Figure 3(a). We manipulated the robotic arm into 10 poses $\mathtt{B}'_i$ and acquired a calibration image set consisting of the same number of images of a 30×30 points calibration grid. The grid was placed ~ 1 m in front of and ~ 0.5 m above the base of the robotic arm. The point spacing of the calibration grid was 24 mm. We also acquired a validation image set consisting of 3 images. See Figure 3(b) for example images from the calibration sequences. Next, we used OpenCV [2] library to obtain the internal calibration matrix $\mathtt{K}$ [11] of the sensor as well as the camera poses $\mathtt{A}'_i$, $i = 1, \ldots, 13$.

Since there was no ground truth information available, we had to use a performance measure different from $E(\mathtt{X}, \mathtt{Z})$ and $E'(\mathtt{X})$ used in the synthetic data experiments. Suppose, that a function $\mathcal{P}(\mathbf{Y}, \mathtt{A}'_i, \mathtt{K})$ projects calibration grid points $\mathbf{Y}_j \in \mathbb{R}^3$, $j = 1, \ldots, 900$ in the world coordinate frame into the i-th image taken by a camera described by its pose $\mathtt{A}'_i$ and internal camera calibration matrix $\mathtt{K}$. The reprojection error of the point $\mathbf{Y}_j$ is defined as $e_{ij}(\mathbf{u}_i, \mathbf{Y}_j, \mathtt{A}'_i, \mathtt{K}) = \|\mathbf{u}_{ij} - \mathcal{P}(\mathbf{Y}_j, \mathtt{A}'_i, \mathtt{K})\|$, where $\mathbf{u}_{ij}$ are the pixel coordinates of the point $\mathbf{Y}_j$ in the i-th image. In case a calibration method recovers both $\mathtt{X}$ and $\mathtt{Z}$, we can judge the quality of the calibration by expressing the reprojection error as $e_{ij}(\mathbf{u}_i, \mathbf{Y}_j, \mathtt{Z}\mathtt{B}_i\mathtt{X}^{-1}, \mathtt{K})$. However, in case only transformation $\mathtt{X}$ is recovered, the camera pose $\mathtt{A}'_i$ can be expressed only with the help of an additional pose $\mathtt{A}'_i = \mathtt{A}'_k\mathtt{X}\mathtt{B}'^{-1}_k\mathtt{B}'_i\mathtt{X}^{-1}$ and we have to define modified reprojection error $e'_{ij} = \frac{1}{m-1}\sum_{k=1,k\neq i}^{m} e_{ij}(\mathbf{u}_{ij}, \mathbf{Y}_j, \mathtt{A}'_k\mathtt{X}\mathtt{B}'^{-1}_k\mathtt{B}'_i\mathtt{X}^{-1}, \mathtt{K})$.

Figure 4(a) shows the statistics of the modified reprojection errors e'_{ij} computed from the calibration image set created by the MATLAB function `boxplot` for methods *Park94*, *Dan99*, *uvhec*, *qhec*, and *dqhec*. Figure 4(b) shows the same statistics, this time for the validation image set. As is the case of synthetics data experiments, *dqhec* slightly outperforms its competitors. Figures 4(c,d) show the reprojection errors e_{ij} computed for the methods *Dorn98*, *Li10*, *uvherwc*, *qherwc*, and *dqherwc* for the calibration and validation image sets, respectively.

6 Conclusion

In this paper, we showed that the method of convex LMI relaxations can be naturally applied to the hand-eye and robot-world calibration problems. We presented three hand-eye and three hand-eye and robot-world calibration parametrizations and by applying the

method of convex LMI relaxations we obtained globally optimal solutions. These formulations provide a new insight into the behavior and complexity of the original problems. The *qhec* parametrization showed the best performance overall. Methods *uvherwc* and *qherwc* do not necessarily provide more accurate results than the previously proposed methods, however, since they do not suffer from the quaternion sign ambiguity, the running time is constant and not exponential in the number of poses.

Acknowledgements

The authors were supported by the EC under projects FP7-ICT-288553 CloPeMa, FP7-SPACE-2012-312377 PRoViDE, and by The Technology Agency of the Czech Republic under the project TA03010398 RoMeSy. This work also benefited from discussions with Florian Bugarin.

References

[1] Mosek: large scale optimization software. `www.mosek.com`.

[2] Open source computer vision library. `www.opencv.org`.

[3] Lin Wang Aiguo Li and Defeng Wu. Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product. *International Journal of the Physical Sciences*, Vol. 5(10):pp. 1530–1536, September 2010.

[4] Nicolas Andreff, Radu Horaud, and Bernard Espiau. On-line hand-eye calibration. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 430–436. IEEE, 1999.

[5] Michel Chasles. Note sur les propriétés générales du système de deux corps semblables entre eux, placés d'une manière quelconque dans l'espace et sur le déplacement fini, ou infiniment petit d'un corps solide libre. *Bulletin des Sciences Mathématiques de Férussac*, XIV:321–326, 1831.

[6] Homer H Chen. A screw motion approach to uniqueness analysis of head-eye geometry. In *CVPR*, pages 145–151. IEEE, 1991.

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

[8] K. Daniilidis and E. Bayro-Corrochano. The dual quaternion approach to hand-eye calibration. In *ICPR '96, Volume I*, page 318, Washington, DC, USA, 1996. IEEE Computer Society.

[9] Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *International Journal of Robotics Research*, 18:286–298, 1998.

[10] Fadi Dornaika and Radu Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE Transactions on Robotics and Automation*, 14:617–622, 1998.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[12] Jan Heller, Michal Havlena, and Tomáš Pajdla. A branch-and-bound algorithm for globally optimal hand-eye calibration. In *CVPR*, 2012.

[13] Jan Heller, Michal Havlena, Akihiro Sugimoto, and Tomáš Pajdla. Structure-from-motion based hand-eye calibration using l_∞ minimization. In *CVPR*, 2011.

[14] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

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

[16] Sin-Jung Kim, Mun-Ho Jeong, Joong-Jae Lee, Ji-Yong Lee, Kang-Geon Kim, Bum-Jae You, and Sang-Rok Oh. Robot head-eye calibration using the minimum variance method. In *ROBIO*, 2010.

[17] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11:796–817, 2001.

[18] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

[19] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD*, Taipei, Taiwan, 2004.

[20] Abed Malti and Joao P Barreto. Robust hand-eye calibration for computer aided medical endoscopy. *IEEE International Conference on Robotics and Automation (ICRA)*, 5543–5549, 2010.

[21] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.

[22] J Michael McCarthy. *Introduction to theoretical kinematics*. MIT press, 1990.

[23] Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129, 1987.

[24] J. Nie. Optimality conditions and finite convergence of Lasserre's hierarchy. *arXiv:1206.0319* June 2012. To appear in *Mathematical Programming*

[25] F.C. Park and B.J. Martin. Robot sensor calibration: solving AX=XB on the euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5):717 –721, October 1994.

[26] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[27] Thomas Ruland, Tomáš Pajdla, and Lars Kruger. Globally optimal hand-eye calibration. In *CVPR*, 2012.

[28] Yongduek Seo, Young-Ju Choi, and Sang Wook Lee. A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system. In *IEEE 12th International Conference on Computer Vision*, pages 1173 –1178, sept. 2009.

[29] Y.C. Shiu and S. Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB. *IEEE Transactions on Robotics and Automation*, 1989.

[30] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.

[31] K. H. Strobl and G. Hirzinger. Optimal Hand-Eye Calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4647–4653, Beijing, China, October 2006.

[32] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 1999.

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

[34] C.C. Wang. Extrinsic calibration of a vision sensor mounted on a robot. *Robotics and Automation, IEEE Transactions on*, 1992.

[35] Zijian Zhao. Hand-eye calibration using convex optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2947–2952, 2011.

[36] Hanqi Zhuang, Z.S. Roth, and R. Sudhakar. Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form $AX = YB$. *IEEE Transactions on Robotics and Automation*, 10(4):549 –554, aug 1994.

[37] Hanqi Zhuang and Yiu Cheung Shiu. A noise tolerant algorithm for wrist-mounted robotic sensor calibration with or without sensor orientation measurement. In *lEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pp. 1095–1100, 1992.