HAL
open science

Robust Hand-Eye Calibration via Iteratively Re-weighted Rank-Constrained Semi-Definite Programming

Chinmay Samant, Adlane Habed, Michel de Mathelin, Laurent Goffin

To cite this version:

Chinmay Samant, Adlane Habed, Michel de Mathelin, Laurent Goffin. Robust Hand-Eye Calibration via Iteratively Re-weighted Rank-Constrained Semi-Definite Programming. IEEE/RSJ International Conference on Intelligent Robots and Systems, november 03-08, 2019, Macau, China, Nov 2019, Macao, China. 10.1109/IROS40897.2019.8968040 . hal-04322467

HAL Id: hal-04322467

https://hal.science/hal-04322467

Submitted on 4 Dec 2023

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.

Robust Hand-Eye Calibration via Iteratively Re-weighted Rank-Constrained Semi-Definite Programming

Chinmay Samant[1,2,3], Adlane Habed[1], Michel de Mathelin[1], Laurent Goffin[1]

***Abstract*— This paper addresses the problem of estimating the Euclidean transformation relating two rigidly attached reference frames from motion measurements: a problem that is commonly referred to as the *Hand-Eye calibration*. The motion measurements are often affected by synchronization issues and hardware malfunctions. When using pose sensors, such as Electromagnetic or Inertial Measurement Units, the measurements are often less reliable than those obtained by a traditional robot-link and camera pair. Corrupt measurements, whether due to disturbances, large synchronization mismatches or malfunctions, are considered outliers that ought to be filtered out of the estimation process. While this may be achieved non-deterministically by using Random Sample Consensus, we propose a deterministic, robust and accurate method for solving the Hand-Eye calibration problem despite the presence of large amounts of outliers and high levels of measurement noise. The proposed method is based on a reformulation of this estimation problem as a rank-constrained semi-definite programming problem allowing for robustness to be enforced via an iteratively re-weighted optimization approach.**

I. INTRODUCTION

Hand-Eye calibration consists in estimating a fixed Euclidean transformation relating the reference frames of two rigidly attached sensors providing motion measurements. Many robotic and medical imaging devices rely on the presence of a camera (the Eye) that is paired with other pose sensors such as the traditional robotic link (the Hand), Inertial Measurement Units (IMU), odometry, Electromagnetic (EM) or a combination of these. In Fig. 1,

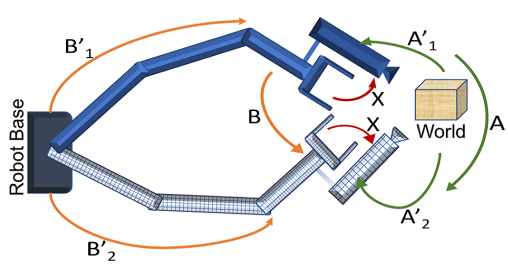

Fig. 1: Traditional Hand-Eye setup in robotics with a camera mounted on a robot gripper attached to the robot base. Transformations are obtained via the kinematic link and camera pose estimation.

A'_1 and A'_2 are the camera transformations obtained from camera pose estimation techniques while B'_1 and B'_2 are those obtained from the robotic link. One motion of the robot hand results into the relative transformations $A = A'_2 A'^{-1}_1$ and $B = B'_2 B'^{-1}_1$. These transformations, along with the rigid transformation X relating the camera and the robot gripper, satisfy $AX = XB$: a relationship originally proposed in [1, 2]. Matrices A, B and X are homogeneous transformations in $SE(3)$. In particular the unknown matrix X is of the form $\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$ where $R \in SO(3)$. Note that at least two relative motions (3 absolute poses) with non-parallel rotational axes are needed to estimate X [1]. In practice several relative motions are used to create multiple sets of matrices A_i and B_i, $i = 1, \ldots, n$, leading to an over-determined system (with possibly noisy data). Hence, X is estimated by solving the problem

$$\begin{aligned} &\min_X \sum_i^n \|(A_i X - X B_i)\| \\ &s.t. X \in SE(3) \end{aligned} \tag{1}$$

where $\|.\|$ denotes the Frobenius norm.

Early Hand-Eye calibration methods [1, 2, 3, 4, 5] were based on estimating the rotation and translation of the sought transformation X independently. However, as noted in [6], in doing so, estimation errors in one would naturally propagate to the estimation of the other. As a result, subsequent methods [7, 8, 9] solve for the rotation and translation simultaneously. For instance, this was carried out through the use of dual-quaternions in [8], by employing a new metric on $SE(3)$ and its Lie group algebra [7], and also by exploiting properties of the Kronecker product [9]. The method proposed in [10] solves the Hand-Eye calibration problem using Second Order Cone Programming. This method, however, fails to maintain the orthogonality of the rotation matrix leading to large estimation errors in the case of noisy data.

The Hand-Eye calibration problem is inherently non-linear because of X's rotational component membership to $SO(3)$: enforcing this membership is paramount. The various $SO(3)$ parameterizations employed to solve this estimation problem all led to formulations that require solving nonlinear programming problems. While these are generally solved using local optimization methods [11, 4], the globally optimal method in [12] solves the problem using Lassere's hierarchy of convex semi-definite programming (SDP) relaxations [13]. Globally optimal methods have also been investigated in [14, 15, 16] when using information from the image instead of relying upon the camera transformations.

The bulk of the literature on Hand-Eye calibration con-

[1]ICube Laboratory, University of Strasbourg (France)-CNRS.
[2]Siemens Healthcare, France
[3]Email: csamant@etu.unistra.fr

siders a robot-link and a camera, both generally providing satisfactorily accurate measurements. As a consequence, the problem of outlier infested data has known very little attention. Many applications involving a camera coupled with either EM sensors or IMUs are particularly prone to spurious measurement readings and/or large amounts of noise. This may be due to disturbances caused by various physical phenomena such as electromagnetic disturbances, edging or exiting the generated EM field, error accumulation, sensor synchronization issues, etc. Such problems affect predominantly medical instruments mounted with EM sensor where the EM transmitter field is constantly disturbed by EM noise. The excessive use of these instruments may also displace the sensors rendering the need for on-the-fly calibration. To account for outliers, the authors of [17] use a Random Sample Consensus (RANSAC) framework with the dual quaternion formulation (from [8]) of the transformations. However, RANSAC is non-deterministic and depends on the number of iterations and thresholds chosen within.

In this paper, we propose a deterministic, robust and accurate method for solving the Hand-Eye calibration problem even in the presence of large amounts of outliers and high levels of measurement noise. The proposed method is based on a rank-constrained SDP formulation of the Hand-Eye calibration. This SDP formulation is obtained by considering the membership of the rotational component of X to the convex hull of $SO(3)$. This membership is described by a convex Linear Matrix Inequality (LMI). The full membership to $SO(3)$ is then enforced via a single matrix rank constraint that, alone, encapsulates the nonlinearity of the problem. To enforce this rank constraint, we employ an efficient method alternating between solving two SDP problems: one for enforcing the rank while the other for recovering the calibration matrix. We exploit this formulation of the Hand-Eye calibration problem to robustify the estimation process through an iteratively re-weighted optimization scheme that allows to reduce the influence of outlier motions in the estimation problem.

The paper is organized as follows. Section II provides some background on SDP and LMIs, the convex hull of $SO(3)$ as well as rank-constrained estimation. Section III is dedicated to our formulation of the Hand-Eye calibration problem and the presentation of our robust algorithm. In Section IV, we give comprehensive comparative results of our experiments against state-of-art methods using both synthetic and real data. Section V concludes our work.

II. Background and Notations

In this section, we present the adopted notations as well as a brief background on Semi-Definite Programming, the notion of convex-hull of rotations in 3-space and the rank-constrained LMI feasibility problem. These are the main ingredients in our rank-constrained SDP reformulation of the Hand-Eye calibration problem (1).

A. Semi-Definite Programming

Semi-Definite Programming (SDP) problems are convex optimization problems with a linear objective function and Linear Matrix Inequality (LMI) constraints, that is,

$$\begin{aligned} &\min \sum_{i=1}^{m} b_i y_i \\ &\text{s.t. } C + \sum_{i=1}^{m} y_i D_i \geqslant 0, \end{aligned} \tag{2}$$

where $y_i \in \mathbb{R}$ are the unknown variables while $b_i \in \mathbb{R}$, matrices $C = C^T \in \mathbb{R}^{n\times n}$ and $D_i = D_i^T \in \mathbb{R}^{n\times n}$ are provided. The inequality sign means that the matrix $L = C + \sum_{i=1}^{m} y_i D_i$ is positive semi-definite ($\leqslant$ would refer to negative semi-definiteness) and the inequality constraint is called as a LMI. Matrix L is constrained between its smallest and the largest eigenvalues, respectively, λ_{min} and λ_{max}. For instance, $L - \lambda_{min} I \geqslant 0$ and $L - \lambda_{max} I \leqslant 0$ (matrix I is the identity matrix). There are several methods that can efficiently solve SDPs, notably interior point methods [18]. This article [18] also provides a generous history and survey of solving SDPs. Though these formulations appear to be very specific form of problems, they are rather widely used in the field of control theory [18, 19] and computer vision [20].

B. Rank-Constrained LMI Feasibility Problem

Consider a symmetric $m \times m$ matrix M and the rank-constrained semi-definite feasibility problem on the left of (3)

$$\begin{array}{ll} \underset{M}{\text{find}} & M \\ \text{s.t.} & M \in \mathscr{C} \\ & M \geqslant 0 \\ & \operatorname{rank} M \leqslant n \end{array} \equiv \begin{array}{ll} \underset{M}{\min} & \operatorname{trace}(MZ^\star) \\ \text{s.t.} & M \in \mathscr{C} \\ & M \geqslant 0 \end{array} \tag{3}$$

where $\mathscr{C}$ is a convex set containing positive semi-definite matrices of rank $n \leqslant m$ or less. If $M^\star$ is a solution to this problem, then there exists a $m \times m$ symmetric positive semi-definite matrix $Z^\star$, of rank at least $m - n$, such that $\operatorname{trace} M^\star Z^\star = 0$. With this $Z^\star$, the solution to the problem on the right-hand side of (3) is also $M^\star$.

As a consequence, alternatively, one may solve the rank constrained problem in (3) by alternating the solutions of the following two SDP problems

$$\begin{array}{ll} \underset{Z}{\min} & \operatorname{trace}(\mathrm{M}^\star \mathrm{Z}) \\ \text{s.t.} & 0 \leqslant Z \leqslant I \\ & \operatorname{trace}(Z) \geqslant m - n \end{array} \rightleftharpoons \begin{array}{ll} \underset{M}{\min} & \operatorname{trace}(MZ^\star) \\ \text{s.t.} & M \in \mathscr{C} \\ & M \geqslant 0. \end{array} \tag{4}$$

Matrix Z, estimated through solving the problem to the left in (4), acts as a direction matrix when estimating M in the problem to the right in (4). This process is intialized with some $Z^\star$ and repeated until convergence. The constraint $0 \leqslant Z \leqslant I$ enforces the semi-definiteness of Z as well as bounds its largest eigenvalue by unity. Along with $\operatorname{trace}(Z) \geqslant m - n$, these two constraints are used as surrogates to the rank of Z. The interested reader may refer to [21], chapter 4.4.1 for further details on this approach.

$$\mathtt{C}(R) = \begin{bmatrix} 1+R_{11}+R_{22}+R_{33} & R_{32}-R_{23} & R_{13}-R_{31} & R_{21}-R_{12} \\ R_{32}-R_{23} & 1+R_{11}-R_{22}-R_{33} & R_{21}+R_{12} & R_{13}+R_{31} \\ R_{13}-R_{31} & R_{21}+R_{12} & 1-R_{11}+R_{22}-R_{33} & R_{32}+R_{23} \\ R_{21}-R_{12} & R_{13}+R_{31} & R_{32}+R_{23} & 1-R_{11}-R_{22}+R_{33} \end{bmatrix}. \tag{5}$$

C. Convex-Hull of Rotations

Consider a 3×3 matrix R and the function $\mathtt{C}$ mapping $\mathbb{R}^{3\times 3}$ to $\mathbb{R}^{4\times 4}$, as given in (5), where R_{ij} denotes R's entry at the ith row and j^{th} column. The set

$$\operatorname{conv} SO(3) = \{R \in \mathbb{R}^{3\times 3} : \mathtt{C}(R) \succeqslant 0\} \tag{6}$$

represents the tightest convex hull of the set of 3-space rotation matrices. This result was first reported in [22] while characterizing orbitopes, then in [23] as part of the characterization of the convex hull to rotations. Indeed, should R be a rotation matrix, the conversion between a unit quaternion $z = a + \boldsymbol{i}b + \boldsymbol{j}c + \boldsymbol{k}d$ (where $|z| = 1$) and matrix R is given by

$$R = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & a^2-b^2-c^2+d^2 \end{bmatrix}. \tag{7}$$

It can be observed that the entries of R are constructed from the terms of a rank-1 4×4 symmetric matrix U

$$U = \frac{1}{a^2+b^2+c^2+d^2}\begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}, \tag{8}$$

the entries of which can be obtained by inverting the relations $R_{11} = a^2+b^2-c^2-d^2$, $R_{12} = 2bc-2ad$, ...etc. This inverse mapping characterizes matrix $\mathtt{C}(R)$ (as given by (5)). Matrix $\mathtt{C}(R)$ being positive semi-definite ensures R's membership to $\operatorname{conv} SO(3)$. If in addition $\mathtt{C}(R)$'s rank is 1, then R is a guaranteed to be a rotation matrix.

III. Method

In this section, we present the main contribution of the paper, that is: our SDP formulation of the Hand-Eye calibration problem and the underlying robust algorithm.

A. Semi-Definite Problem Formulation

We first formulate the Hand-Eye calibration problem as a rank-constrained SDP and show how this can be solved using alternating optimization of two convex problems. The formulation herein does not explicitly address the problem of robustness. This is addressed in the next section which also provides our full algorithm.

Assuming $n \geqslant 2$ motions available, an alternative formulation to the Hand-Eye calibration problem (1) is the following:

$$\begin{aligned} &\min_{X} \sum_i \sigma_{max}(A_i X - X B_i) \\ &\text{s.t. } X \in SE(3), \end{aligned} \tag{9}$$

where $\sigma_{max}(.)$ denotes the largest singular of its matrix argument. Note that this problem is not equivalent to the one stated in (1). The latter minimizes the L_2-norm, which is generally preferred over other norms. However, relying on the cost function in (9) carries a number of advantages. Firstly, minimizing the largest singular value of a matrix is a SDP problem. As a result, such minimization may be subjected to additional LMI constraints such as, for the problem at hand, the membership of the rotational component of X to the convex hull of rotations $\operatorname{conv} SO(3)$ (6) rather than $SO(3)$. Then, the largest singular value alone is an excellent indicator for the relative quality of motion measurements employed to estimate X. Indeed, with the correct estimate of X, motions with small largest singular values of $A_i X - X B_i$ indicate that A_i and B_i are good measurements and bad otherwise. Such singular values can hence be trusted to provide weights to the various motions involved in X's estimation.

To see that minimizing the largest singular value of a matrix is a SDP problem, observe that the singular values of a matrix, say $(AX - XB)$, are the square root of the eigenvalues of $(AX-XB)^T(AX-XB)$. Therefore, as discussed in Section II-A, the latter matrix and its largest eigenvalue σ_{max}^2 satisfy $(AX-XB)^T(AX-XB) - \sigma_{max}^2 \leqslant 0$. Equivalently, and because singular values are always nonnegative, we can write

$$\frac{1}{\sigma_{max}}(AX-XB)^T(AX-XB) - \sigma_{max} I \leqslant 0. \tag{10}$$

Note that the matrix on the left-hand side of (10) is the Schur complement of the matrix on the left-hand side of

$$\begin{bmatrix} \sigma_{max} I & (AX-XB) \\ (AX-XB)^T & \sigma_{max} I \end{bmatrix} \succeqslant 0. \tag{11}$$

Furthermore, by Schur's complement lemma (a proof of which can be found in [24]), the two inequalities (10) and (11) are equivalent. As a result, considering X of the form $\begin{bmatrix} R_X & t_X \\ 0^T & 1 \end{bmatrix}$, problem (9) may be equivalently restated as:

$$\begin{aligned} &\min_{R_X, t_X, \sigma} \sum_i^n \sigma_i \\ &\text{s.t.} \\ &\qquad \begin{bmatrix} \sigma_i I & (A_i X - X B_i) \\ (A_i X - X B_i)^T & \sigma_i I \end{bmatrix} \succeqslant 0, \\ &\qquad \mathtt{C}(R_X) \succeqslant 0, \\ &\qquad \operatorname{rank} \mathtt{C}(R_X) = 1. \end{aligned} \tag{12}$$

Note that $\mathtt{C}(R_X)$ ensures R_X's membership to $\operatorname{conv} SO(3)$ (6). Together with the rank condition, these two constraints

ensure the resulting R_X to be in $SO(3)$, hence $X \in SE(3)$. Problem (12) carries the advantage of confining the nonlinearity of the problem to a single rank constraint. A relaxed convex problem can simply be solved by dropping this constraint. However, the resulting solution would not guarantee the estimated R_X to be in $SO(3)$. We show, in the following, that a better solution satisfying $\mathtt{C}(R_X)$'s rank condition in practice can be obtained. The proposed approach is based on the alternating minimization procedure described in Section II-B. Given some initial 4×4 matrix $Z^\star$ of rank 3, the idea is to alternate between solving

$$\begin{aligned} &\min_{R_X, t_X, \sigma} \sum_i^n \sigma_i + \mu \,\mathrm{trace}(\mathtt{C}(R_X) Z^\star) \\ &\text{s.t.} \\ &\quad \begin{bmatrix} \sigma_i I & (A_i X - X B_i) \\ (A_i X - X B_i)^T & \sigma_i I \end{bmatrix} \succeq 0, \\ &\quad \mathtt{C}(R_X) \succeq 0, \end{aligned} \tag{13}$$

which involves a regularization term controlled by a fixed nonnegative parameter μ, and the problem

$$\begin{aligned} &\min_Z \,\mathrm{trace}(\mathtt{C}(R_X^\star) Z) \\ &\text{s.t. } \mathrm{trace}(Z) \geqslant 3,\ 0 \leqslant Z \leqslant I. \end{aligned} \tag{14}$$

Here, $R_X^\star$ is obtained by solving (13) while the direction matrix $Z^\star$ is the minimizer obtained by solving (14). Note that both (13) and (14) are SDP, hence convex, problems. The algorithm iterates until there is no change in X. Note that the choice and influence of the regularization parameter μ are discussed in the experiments section. The choice of the regularization parameter μ and the initial $Z^\star$ is delayed to the next section, following the presentation our full robust algorithm.

B. Robust Hand-Eye Calibration

The convexity of the alternating problems (13) and (14), along with the estimates of the largest singular values, provide a very suitable ground for an iteratively reweigthted procedure. Such procedures are known to handle efficiently both noise and outliers. To do so, consider a set of singular values $\sigma_i^{(k-1)}$ and $X^{(k-1)}$ all estimated at iteration $k-1$ of the algorithm, $k \geqslant 1$. With $X^{(k-1)}$, one can estimate the direction matrix $Z^{(k)}$ to be used to estimate $X^{(k)}$ at the next iteration k by solving (14). In turn, the weights $w_i^{(k)}$ are necessary to estimate $X^{(k)}$. They are inversely proportional to $\sigma_i^{(k-1)}$ and can be first computed as

$$w_i^{(k)} = \frac{1}{\sqrt{\sigma_i^{(k-1)}} + \epsilon}, \tag{15}$$

where ϵ is a very small number, then normalized as described in Algorithm 1. Given the weights and direction matrix, $X^{(k)}$ is estimated by solving

$$\begin{aligned} &\min_{R_X, t_X, \sigma} \sum_i^n w_i^{(k)} \sigma_i + \mu \,\mathrm{trace}(\mathtt{C}(R_X) Z^{(k)}) \\ &\text{s.t.} \\ &\quad w_i^{(k)} \begin{bmatrix} \sigma_i I & (A_i X - X B_i) \\ (A_i X - X B_i)^T & \sigma_i I \end{bmatrix} \succeq 0, \\ &\quad \mathtt{C}(R_X) \succeq 0. \end{aligned} \tag{16}$$

Such weighting scheme efficiently reduces the influence of corrupt measurements. The full algorithm is given by Algorithm 1. In this algorithm, $\gamma = 1e{-}7$ is a threshold value to determine that X's estimation is unchanged compared to its estimate at the previous iteration (in the Frobenius norm sense). This is the stopping criterion. The initial matrix $Z^{(0)}$ must be rank-3. In practice one may start from any diagonal matrix with all ones on the diagonal except for one zero-entry. We suggest to run the algorithm with all 4 such possible values of matrix $Z^{(0)}$ and retain the one with the minimum cost of the SDP. Furthermore, we found that $\mu = 1e{-}6$ was a good choice in all our experiments. Its influence on the algorithm is discussed in the experiments section. All weights can be initialized to 1.

Algorithm 1: Robust Hand-Eye calibration algorithm

Input: $A_{i=1\ldots n}, B_{i=1\ldots n}, \gamma, Z^{(0)}$
Output: X

$w_{i=1\ldots n}^{(0)} = 1,\ \zeta = +\infty,\ k = 0$
Estimate $X^{(0)}$ by solving (16)
while $\zeta > \gamma$ **do**
  $k = k + 1$
  Estimate $Z^{(k)}$ by solving (14)
  $w_{i=1\ldots n}^{(k)} = \frac{1}{\sqrt{\sigma_i^{(k-1)}}} + \epsilon$
  $w_{i=1\ldots n}^{(k)} = \frac{w_{i=1\ldots n}^{(k)}}{\sum_{j=1}^n w_j^{(k)}}$
  Estimate $X^{(k)}$ by solving (16)
  $\zeta = \| X^{(k-1)} - X^{(k)} \|$
return $X = X^{(k)}$

IV. Experiments and results

We have conducted experiments using 3 types of datasets: synthetic dataset, real motion dataset of hand-eye robot setup and real motion dataset from an EM sensor-camera setup. In all the experiments, our robust Algorithm 1 was used with parameter $\mu = 1e-6$, threshold $\gamma = 1e-7$ (as discussed given in Section III-B). Also, our initial $Z^{(0)}$ was arbitrarily chosen as the diagonal matrix with all ones but the first entry.

A. Synthetic Data Experiments

In the synthetic experiments, we simulated a scene of 50 points confined to the surface of a 1 meter radius sphere. Random views of the scene were generated by moving the virtual camera around the sphere surface while

constraining all the points to be visible in all the images. The camera intrinsic parameters were adjusted accordingly. The camera was placed at a 2.5 meters mean distance from the center of the sphere and 0.7 meters standard deviation. We provided a hand-eye transformation X which, along with the camera pose, was used to create ground truth absolute hand transformations. These were used to create relative motions for both camera and hand. For each level of changing noise or percentage of outliers, we ran 100 independent trials over which the error measurements were averaged.

To quantify the results, we used the RMS errors in rotation unit quaternion $\|q - \hat{q}\|$ and RMS of the relative errors in translation $\|t - \hat{t}\| / \|t\|$. These are customary metrics used in [2, 4, 8]. Though the metrics for rotation and translation are given independently, note that the success of a method depends upon the performance in both metrics of rotations and translations combined.

In the synthetic experiment results, methods *qhec*, *uvhec* and *dqhec* are from [12], *tsai* refers to [2], *Inria* is from [4] and *navy* refers to [5]. For these methods, we used the Matlab implementations from [25]. *'traceZ'* refers to the Matlab implementation of Algorithm 1 using YALMIP [26] as a parser and SeDuMi [27] as a SDP solver.

1) Experiments with varying noise: In the first experiment, we used 30 motions of the virtual Hand-Eye setup throughout 100 trials for increasing noise levels. An absolute Gaussian noise was added on the quaternion rotation of hand transformations with increasing standard deviation (0 to 0.2 quaternions). Relative Gaussian noise was added on the translation vector of both hand transformations (0 to 20%). For the camera transformations, a Gaussian pixel noise was added on the 2D pixels from the camera views with an increasing intensity (0 to 2 pixels). The results are illustrated in Fig. 2. Note that the best performing methods are *qhec* (as well as *uvhec*) [12] and ours with a better translation estimation with our method.

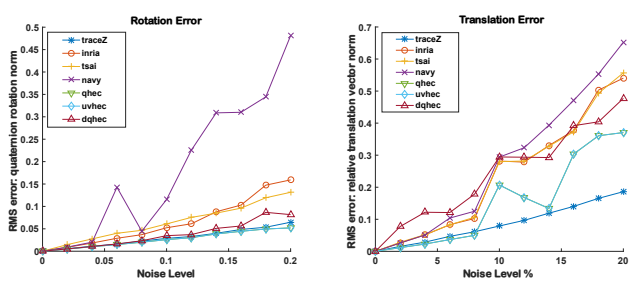

Fig. 2: Gaussian noise on hand and camera motions. RMS of quaternion rotation errors (left), RMS of relative translation vector errors (right).

In the second experiment, we used the same noise adding scheme as above except that the Gaussian noise on the hand transformations is replaced by uniform noise. It is generally hard to model the noise from IMUs/EM sensors and it is usually non-Gaussian. Pixel noise remained Gaussian as in the previous experiment. Our results are illustrated in Fig. 3 in which we note a similar performance as the previous experiment.

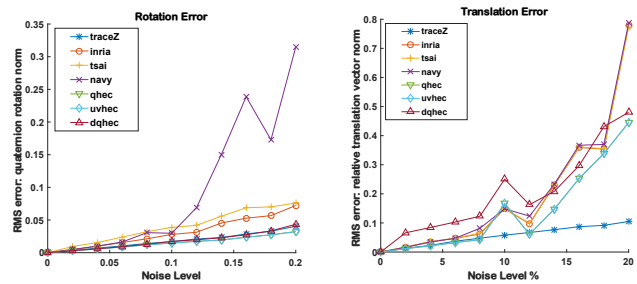

Fig. 3: Uniform noise on hand and camera motions. RMS of quaternion rotation errors (left), RMS of relative translation vector errors (right).

2) Experiments with Outliers: In this experiment, 30 motions with a marginal noise were used throughout the experiment. We added a constant amount of 0.05 standard deviation absolute Gaussian noise on quaternion of hand transformations, 5% relative Gaussian noise on translation vectors of the hand and 1 pixel Gaussian noise on cameras. To create the outlier data, we added an increasing percentage of outliers (totally random data) in the total number of motions. Only hand transformations (occurring from IMU, EM sensors, odometry or the combinations of these sensor measurements) were affected by outliers. The camera transformations were not affected by outliers since cameras have a direct view of the target. Also, there are various robust algorithms already present on the camera hardware/software platforms allowing one to obtain good camera pose estimation. The results reported in Fig. 4 show that the outliers are efficiently rejected up to 70% (only 9 out of 30 good relative motions). After 70%, our method starts to fail, albeit not by a big margin.

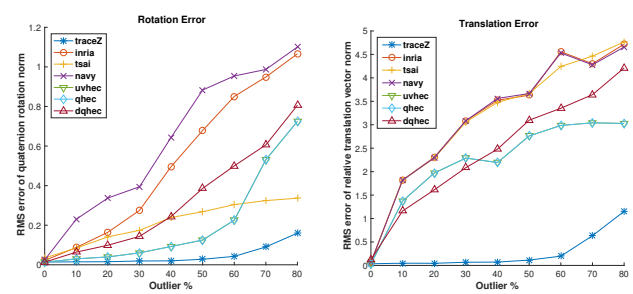

Fig. 4: Outliers replacing increasing percentage of motions. RMS of quaternion rotation errors (left), RMS of relative translation vector errors (right).

3) Time performance of our method: In this experiment, the noise levels and outliers were used as in the outlier experiments. Here we present two types of graphs (Fig. 5, Top). The line graph presents the average time taken by one iteration of the weighting of the motions (the 'while loop' in Algorithm 1). The bar graph presents the number of iterations taken by the 'while loop' before termination.

The average of this time was computed for 50 trials for each outlier (completely random data) percentage. The bar part of the graph shows that the number of iterations hardly increases with increasing number of outliers. Similarly, the line graph also fluctuates only in milliseconds highlighting the apparent disconnection between the ratio of outliers and computation time.

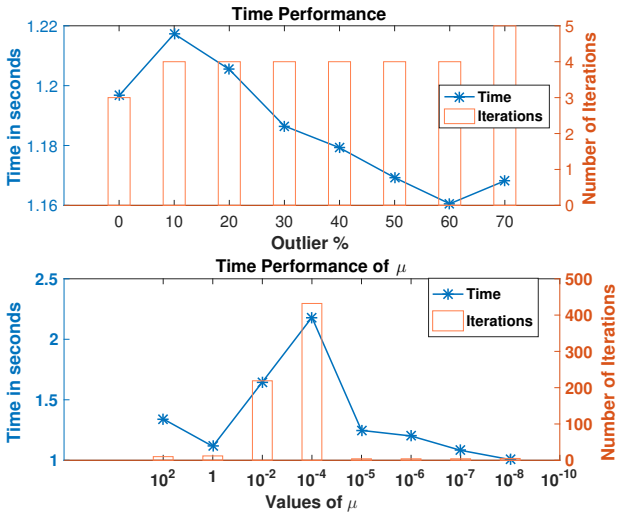

Fig. 5: Top: Average time performance of each iteration. Bottom: Average time performance while changing the regularization parameter. Left Y-axis: Average time taken by each iteration in 50 trials. Right Y-axis: Total number of iterations.

4) Experiments with regularization parameter: We have used μ as a regularization parameter in our problem formulation (13). Changing the value of μ affects the time of convergence of our method. In this experiment, we imposed a nominal absolute noise of 0.05 standard deviation (in quaternions) to hand motion rotations and 5% relative Gaussian noise on hand translations. 1 pixel noise was added to cameras. Fig. 5 (bottom) shows the time per iteration and number of iterations taken for each μ. We observe that choosing a correct μ value is critical for the time. For instance, with $\mu = 1e-4$, the program takes very long (approx. 16 minutes). With $\mu = 1e-5$, it only takes a 4 to 5 seconds to terminate.

B. Real Data Experiments with Robot-Camera Setup

The authors of [17] published an open source dataset for hand-eye calibration recorded on a real robot: a UR-10 arm equipped with a RealSense SR300 RGB-D sensor mounted rigidly close to the end effector. They also provide an algorithm to time-align the motions according to the time-stamps and also to pre-filter the motions and reject those with very similar information. Only those with the maximum information are retained. Using this algorithm, we could obtain 1686 'time-aligned' motion pairs to test our method. Their 'RANSAC Scalar (RS) based inlier check' algorithm uses 30 pre-selected motions out of 1686 motions to perform Hand-Eye calibration. The RS method uses RANSAC framework with Dual-Quaternion parameterization [8] to eliminate the outliers. RS method is a RANSAC framework and different Hand-Eye calibration methods can be used within this framework. The goal of this experiment was not to demonstrate the performance of the Hand-Eye calibration methods but to compare the performance of the non-deterministic RANSAC outlier removal approach against our deterministic method. We used the following chain to recompute camera motions $\hat{A}_j$. Subscript $j = 1 \ldots n$ relates to all the motions corresponding to the A_i where $i = 1$.

$$\hat{A}_i = X B_i B_j^{-1} X^{-1} A_j. \tag{17}$$

Then, we compared $\hat{A}_{i=1}$ against the camera pose A_1 given by camera motions from the dataset. We evaluated the difference between the two rotations in angle metric and the relative translation using Euclidean distance. These are the same metrics used in [17]. The final rotation and translation error is the RMS error of all $\hat{A}_1$s computed with all corresponding A_js. For this experiment, we randomly drew 100 sets of 30 motions out of 1686 time-aligned motions *without* using the pre-selection algorithm from [17]. Using each sample of these 30 motions, we computed X. We compared the results using our kinematic chain (17). Fig. 6 summarizes the results and shows that the overall performance is superior at every sample even without using the pre-selection. Note that, in Fig. 6, 'ransac' line is entirely flat because RS method selects the best 30 motions to produce one Hand-Eye calibration output against which we compare our results from 100 samples. In this comparison, especially in Fig. 6 (left), our worst result obtained with some randomly obtained 30 motions is better than the result from RS method with pre-selected motions.

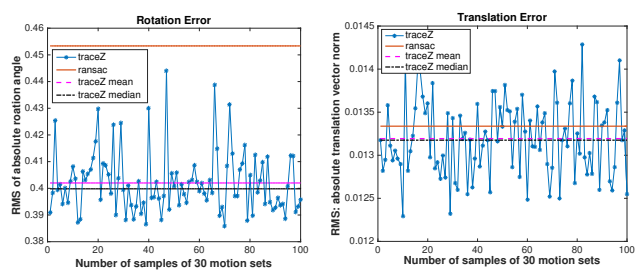

Fig. 6: Real data from robot hand. Comparison against 'ransac' method [17] using 30 pre-selected motions.

C. Real Data Experiments with EM-Camera Setup

Fig. 7 shows our own setup for Hand-Eye calibration experiment. In this setup, we used a checkerboard calibration pattern in the view of a calibrated camera providing us the camera transformations. We used NDI Aurora electromagnetic tracking system to generate hand transformations. This system consists of an EM transmitter, an EM receiver and a power unit. The EM receiver was mounted on the calibration pattern. Hence, the origin of the calibration pattern and the EM sensor (receiver) are related via a rigid transformation X. The transformation

between the camera and EM transmitter was also fixed. We recorded a sequence of motions by moving the calibration pattern in front of the camera with the receiver being in the EM field of the transmitter at the same time. We avoided the cases of optical occlusions as well as rapid movements of checkerboard pattern to acquire time-synchronized pairs of camera-EM transformations. Our software can synchronize the image and EM-data acquisition while recording the data. We added high electromagnetic interference to the electromagnetic receiver for some of the EM transformations. This can be easily achieved by holding a varying electromagnetic field device such as a cell-phone (while calling) close to the EM receiver. The noise is extreme enough to distort the EM field near the EM receiver randomly. The Windows-based driver/software which comes with NDI Aurora system can display the sensor accuracy values in real-time. We can observe (in real-time) the high disturbance being caused by cell-phone EM field, which helps us to record an 'outlier' value during data acquisition. As we only needed a randomly caused high EM disturbance, we can accept this dataset without accurate specifications of cell-phone EM field.

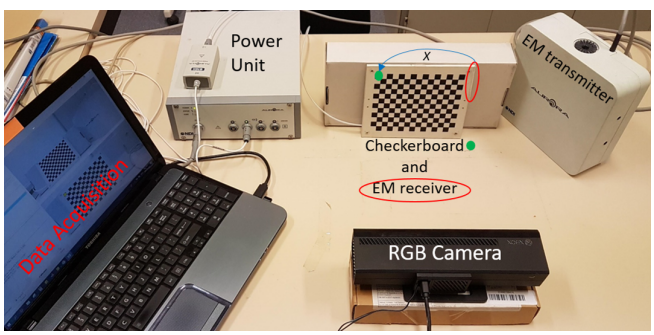

Fig. 7: Our camera-EM data acquisition setup. The camera and EM transmitter remains the fixed. With each motion, the checkerboard pattern moves in front of camera along with the attached EM receiver.

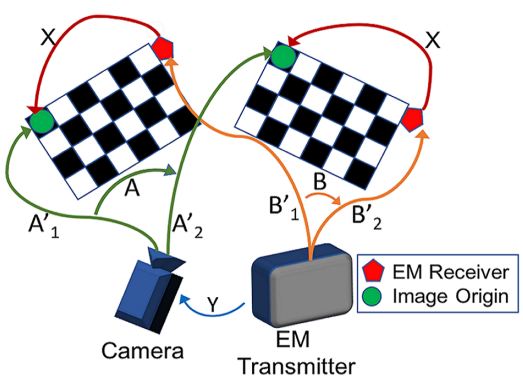

Fig. 8: Schematic of EM sensor - camera setup to demonstrate Hand-Eye calibration. Contrary to the original Hand-Eye calibration setup, the camera is fixed and the calibration pattern moves: the chain of transformation remains equivalent.

Fig. 8 shows the schematic of our EM-camera setup. There exists a Y fixed transformation between the camera and EM transmitter. This is equivalent to World-Robot Base calibration in the robot hand-camera setup. Estimation of this transformation is not necessary to compute Hand-Eye calibration of the checkerboard as long as the EM transmitter and camera are not moved during the experiment. Similar to the experiments in Section IV-B, we used the chain of transformations given in (17) with a small modification to compute the camera transformations. Instead of only using A_1, we used all the N motions A_is ($i = 1 \dots N$) for this experiment. Then we re-projected the computed pixels onto to the reference image using 2D homography equation $\hat{x} = Hx$, where H is 3×3 non-singular matrix given by the camera intrinsics and x are the coordinates of 2D points. Fig. 9 shows the average pixel error (Euclidean distance) corresponding to all $\hat{A}_i$s computed from all A_js (17). In this experiment, we used 20 (for Easy) and 10 (for Hard) relative motions of the EM sensor(Hand)-checkerboard(Eye) setup. In Fig. 9, 'Easy' dataset is without outliers and in 'Hard' dataset, 3 out of 10 of the motions are affected by EM interference outliers. We can see that in the 'Easy' case our method performs fairly in comparison to the other methods (similarly seen in the Gaussian noise simulation experiment in Fig. 2) but excels in eliminating the outliers in the 'Hard' case. We compared our results against *Inria* [4], *Tsai* [2], *Navy*[5] and *Heller-dqhec* [12]. In comparison to synthetic experiment section IV-A, we left out a few methods because of their poor performance with this dataset.

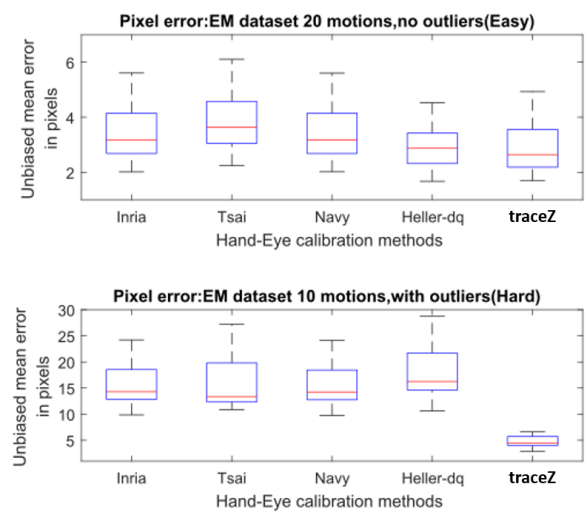

Fig. 9: Dataset from EM-sensor setup. The graphs show the distribution of the pixel errors. In the box plot, the red central line indicates the median pixel error. The top and bottom blue lines of the box indicate 75th and 25th percentile respectively. The whiskers extend to the most extreme data points.

V. Conclusion

We have presented a method to solve the Hand-Eye calibration problem. The data measurements from sensors such as EM or IMUs can be corrupted due to interference, time misalignment and/or communication errors. We have reformulated the problem using a SDP formulation in

which the nonlinearity of the problem is confined to a single rank constraint. Based on this, we proposed an alternating procedure to cope with the nonlinearity and an iterative re-weighting scheme against the outliers. This is a deterministic and robust method that has provided excellent results in comparison to the existing ones.

Acknowledgement

This research was funded by Siemens Healthcare France (under CIFRE ANRT convention with the ICube - University of Strasbourg, CNRS - Laboratory) and by the IHU Strasbourg through the ARES project.

References

[1] 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*, vol. 5, no. 1, pp. 16–29, 1989.

[2] 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*, vol. 5, no. 3, pp. 345–358, 1989.

[3] J. C. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *The International Journal of Robotics Research*, vol. 10, no. 3, pp. 240–254, 1991.

[4] R. Horaud and F. Dornaika, "Hand-eye calibration," *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, 1995.

[5] F. C. Park and B. J. Martin, "Robot sensor calibration: solving ax= xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[6] H. H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 145–151.

[7] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4647–4653.

[8] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[9] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *Second International Conference on 3-D Digital Imaging and Modeling, 1999*. IEEE, 1999, pp. 430–436.

[10] Z. Zhao, "Hand-eye calibration using convex optimization," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2947–2952.

[11] H. Zhuang and Y. C. Shiu, "A noise tolerant algorithm for wrist-mounted robotic sensor calibration with or without sensor orientation measurement," in *Proceedings of the lEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 1992, pp. 1095–1100.

[12] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and robot-world calibration by global polynomial optimization," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 3157–3164.

[13] J. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[14] J. Heller, M. Havlena, and T. Pajdla, "Globally optimal hand-eye calibration using branch-and-bound," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 1027–1033, 2016.

[15] T. Ruland, T. Pajdla, and L. Krüger, "Globally optimal hand-eye calibration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1035–1042.

[16] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla, "Structure-from-motion based hand-eye calibration using L-∞ minimization," in *IEEE Conference on Computer Vision and Pattern Recognition, 2011*. IEEE, 2011, pp. 3497–3503.

[17] F. Furrer, M. Fehr, T. Novkovic, H. Sommer, I. Gilitschenski, and R. Siegwart, "Evaluation of combined time-offset estimation and hand-eye calibration on robotic datasets," in *Field and Service Robotics*. Springer, 2018, pp. 145–159.

[18] L. Vandenberghe and V. Balakrishnan, "Algorithms and software for lmi problems in control," *IEEE Control Systems*, vol. 17, no. 5, pp. 89–95, 1997.

[19] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM studies in applied mathematics: 15, 1994, vol. 15.

[20] D. P. Paudel, A. Habed, C. Demonceaux, and P. Vasseur, "Lmi-based 2d-3d registration: From uncalibrated images to euclidean scene." in *Computer Vision and Pattern Recognition*, 2015, pp. 4494–4502.

[21] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing USA, 2015.

[22] R. Sanyal, F. Sottile, and B. Sturmfels, "Orbitopes," *Mathematika*, vol. 57, no. 2, pp. 275–314, 2011.

[23] J. Saunderson, P. A. Parrilo, and A. S. Willsky, "Semidefinite descriptions of the convex hull of rotation matrices," *SIAM Journal on Optimization*, vol. 25, 03 2014.

[24] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *Journal of Process Control*, vol. 10, no. 4, pp. 363 – 385, 2000.

[25] C. Wengert, "Hand-eye calibration implementation," Jan 2016. [Online]. Available: https://github.com/christianwengert/calib_toolbox_addon

[26] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, Taipei, Taiwan, 2004.

[27] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.