Control Engineering Practice 109 (2021) 104726

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A fast and accurate new algorithm for hand–eye calibration on SO(3)×$\mathbb{R}^3$☆

Shuwei Qiu *, Miaomiao Wang, Mehrdad R. Kermani

Electrical and Computer Engineering, Western University, London, Ontario, Canada

ARTICLE INFO

Keywords:
Hand–eye calibration
$AX = XB$
Point set matching
Nonlinear estimator
Lie group SO(3)

ABSTRACT

In this paper, we propose a new solution for an old problem in robotics related to the calibration of the hand, i.e., the robot's end-effector, and the eye, i.e., a stereo vision camera. The problem is formulated as a point set matching problem and a nonlinear estimator on manifold SO(3)×$\mathbb{R}^3$ used for obtaining the solution. The main advantage of the proposed approach is that it allows to decouple the error associated with the rotational estimation on Lie group SO(3) from the error of the translational estimation on $\mathbb{R}^3$, which subsequently allows to tune the learning rates for the rotation and translation estimation, separately. This will result in significant increase in the convergence speed of the proposed approach. To show the advantages of our approach, we compare the results with those obtained from other conventional hand–eye calibration solutions as well as those based on point set matching. The experimental results will demonstrate that our proposed hand–eye calibration approach outperforms other approaches in terms of accuracy and computational speed.

1. Introduction

The new collaborative robot manipulators intended for interactive manipulation tasks use exteroceptive sensors such as stereo cameras as a common sensor. The need of using these cameras necessitates the camera calibration — a subject under in-depth research for decades (Harris & Teeder, 1993). The camera used for this purpose can be either on-board the robot (eye-in-hand) or mounted near the robot as an independent device (eye-to-hand). The information acquired from the camera is always represented in the coordinate system of the camera. To use the data from the camera along with the robot's proprioceptive measurements, it is necessary to represent the acquired data within the coordinate system of the robot. This requires estimating the transformation matrix between the camera's and the robot's coordinate system, which is the goal of the hand–eye calibration problem (Bishop & Spong, 1999). However, it is difficult to precisely estimate such a transformation matrix (Xiao et al., 2004). Variations of this problem exist in other application fields such as navigation systems for autonomous vehicles (Figueroa & Mahajan, 1994; Wang & Tayebi, 2020a, 2020b), vision-based autonomous landing for aerial vehicles (Bhargavapuri, Shastry, Sinha, Sahoo, & Kothari, 2019), visual servoing for autonomous underwater vehicles (Allibert, Hua, Krupínski, & Hamel, 2019), to name a few.

The hand–eye calibration problem is conventionally formulated as $AX = XB$ or $AX = YB$ problems (Shah, Eastman, & Hong, 2012). The $AX = XB$ formulation provides a solution for the unknown transformation matrix (X) between the robot end-effector and the camera. The $AX = YB$ formulation, on the other hand, provides solutions for both the unknown transformation matrix between the robot end-effector and the camera (X) as well as the unknown transformation matrix between the world and the robot base (Y). The objective of the current study is to obtain X in $AX = XB$ formulation. A typical configuration for solving this problem is shown in Fig. 1 in that A_i ($i = 1, 2$) denotes the homogeneous transformation matrix between the robot base and the robot gripper for the two different poses, B_i ($i = 1, 2$) denotes the homogeneous transformation matrix between the camera's frame and the world's frame affixed to the checkerboard pattern, and X is the homogeneous transformation matrix to be obtained. In this context, it is not difficult to show that,

$$A_1 X B_1 = A_2 X B_2 \Rightarrow A_2^{-1} A_1 X = X B_2 B_1^{-1} \Rightarrow AX = XB \tag{1}$$

where $A = A_2^{-1} A_1$ and $B = B_2 B_1^{-1}$. To make these notations consistent with those that are common in robotics literature, X, A_i, and B_i are relabeled as ${}^{G}_{C}T$ (between the gripper's frame and the camera's frame), ${}^{B}_{G}T_i$ (between the robot base's frame and the gripper's frame), and ${}^{C}_{W}T_i$ (between the camera's frame and the world's frame), respectively. Different methods have been proposed for obtaining ${}^{G}_{C}T$ (or X) within this context (Horaud & Dornaika, 1995; Park & Martin, 1994; Shah, 2013; Wu, Sun, Wang, & Liu, 2019). However, these methods treat the data from stereo cameras in exactly the same way as that from single cameras, and they overlook the benefit of depth recovering feature of stereo cameras. In the proposed solution, we leverage the depth

☆ This research was supported in part through funds received from the Natural Sciences and Engineering Research Council of Canada, Canada Foundation for Innovation, and Ontario Centres of Excellence, Canada.

* Corresponding author.
E-mail addresses: sqiu47@uwo.ca (S. Qiu), mwang448@uwo.ca (M. Wang), mkerman2@uwo.ca (M.R. Kermani).

https://doi.org/10.1016/j.conengprac.2021.104726
Received 29 June 2020; Received in revised form 21 October 2020; Accepted 31 December 2020
Available online xxxx
0967-0661/

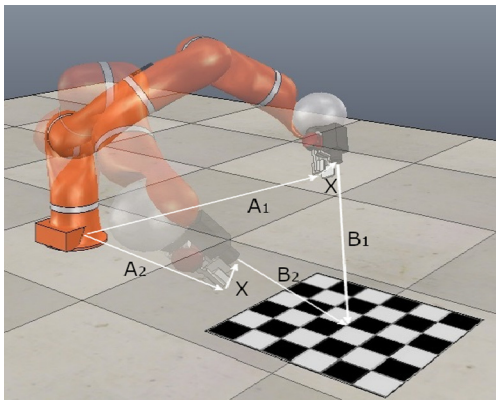

Fig. 1. The classical configuration for solving $AX = XB$.

recovering feature of stereo cameras to solve the hand–eye calibration problem as a point set matching problem. By doing so, a better calibration accuracy is achieved as shown experimentally.

In what follows, we briefly review the point set matching formulation of the hand–eye calibration problem. If ${}^{B}_{C}T$, the homogeneous transformation matrix between the robot base frame and the camera frame, is available, then the relation among ${}^{B}_{C}T$, ${}^{B}_{G}T$ (the homogeneous transformation matrix between the robot base frame and the gripper frame) and ${}^{G}_{C}T$ (the homogeneous transformation matrix between the gripper frame and the camera frame) can be expressed as,

$$ {}^{B}_{C}T = {}^{B}_{G}T \; {}^{G}_{C}T \tag{2} $$

By rearranging (2), one can deduce ${}^{G}_{C}T$ as,

$$ X = {}^{G}_{C}T = {}^{B}_{G}T^{-1} \; {}^{B}_{C}T \tag{3} $$

in that ${}^{B}_{G}T$ is available through robot kinematics and ${}^{B}_{C}T$ is the transformation matrix between the robot base frame ($\{B\}$) and the camera frame ($\{C\}$). Since matching the points represented in two frames (i.e., point set matching) is a common approach for estimating the homogeneous transformation matrix between two frames, one can formulate the hand–eye calibration problem as a point set matching problem. The point set matching approaches will be reviewed in Section 2. In our previous work, we studied this idea of formulating the hand–eye calibration problem as a point set matching problem and proposed a solution based on the gradient descent algorithm on the special Euclidean group SE(3) (Qiu, Wang, & Kermani, 2020). We named this solution as GD-SE(3). In this solution, the gradients of the rotation matrix (R) and the translation vector (p) were grouped in a homogeneous transformation matrix in SE(3). Because of this integration, the convergence speed of GD-SE(3) is slow especially when the initial value of the unknown transformation is far from the optimal solution.

In this paper, we propose a new solution on manifold $\mathrm{SO}(3) \times \mathbb{R}^3$ for the hand–eye calibration problem motivated by the nonlinear estimators proposed in Wang and Tayebi (2019, 2020a). The proposed algorithm offers both accurate results and fast convergence speed that rivals state-of-the-art solutions for the hand-aye calibration problem. We call our proposed algorithm HI-SO(3)R3 for its high convergence speed. The contributions of this work are summarized as follows:

- We propose a new algorithm on manifold $\mathrm{SO}(3) \times \mathbb{R}^3$ for matching two rigidly related point sets to estimate the homogeneous transformation matrix between them.
- We extend the nonlinear continuous observer proposed in Wang and Tayebi (2019, 2020a) to a nonlinear discrete observer (estimator) for point set matching.
- We show that the error of the rotational estimation on SO(3) are decoupled from that of the translational estimation on $\mathbb{R}^3$. With this decoupling property, the learning rates for rotation and translation estimation can be tuned separately, which significantly increases the convergence speed of the proposed algorithm without affecting its accuracy.
- We propose a strategy for selecting suitable learning rates for the estimation of the unknown rotation matrix and translation vector such that the convergence is guaranteed.
- We provide exhaustive comparison results with state-of-the-art solutions to demonstrate the effectiveness of our proposed algorithm.

The rest of the paper is organized as follows: Section 2 briefly outlines related works in the field of hand–eye calibration and point set matching. Section 3 introduces the proposed algorithm, provides corresponding mathematical derivations, and offers guidelines for selecting the learning rates. Section 4 experimentally evaluates the effectiveness of the proposed approach. At the end, Section 5 concludes the paper.

2. Related works

2.1. $AX = XB$ calibration

The most conventional formulation of the hand–eye calibration problem is given as $AX = XB$. This well-known formulation was first developed in Shiu and Ahmad (1989) and Tsai and Lenz (1989). Using homogeneous transformation matrix one can expand $AX = XB$ formulation as follows,

$$ \begin{bmatrix} R_A & p_A \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_X & p_X \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} R_X & p_X \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_B & p_B \\ \underline{0} & 1 \end{bmatrix} $$

$$ \Rightarrow \begin{cases} R_A R_X & = R_X R_B \\ R_A p_X + p_A & = R_X p_B + p_X \end{cases} \tag{4} $$

where R_i represents the rotation matrix and p_i represents the translation vector for $i = A, B$, and X, respectively. Different methods have been proposed to solve this problem that can be categorized as separable, simultaneous, and iterative methods (Shah et al., 2012).

In separable methods, the rotation (R_X) and the translation (p_X) portion of the X matrix are solved in sequence. Park and Martin (1994) found a least-square solution for $AX = XB$ using Lie theory. They solved R_X and p_X by minimizing the following error functions,

$$ e_1 = \sum_{i=1}^{m} \| R_X \beta_i - \alpha_i \|^2 \tag{5} $$

$$ e_2 = \sum_{i=1}^{m} \| (R_{A_i} - I) p_X - R_X p_{B_i} + p_{A_i} \|^2 \tag{6} $$

where m is the number of measurements, $\alpha_i = \log R_{A_i}$, and $\beta_i = \log R_{B_i}$. R_X and p_X were solved as,

$$ R_X = (M^T M)^{-\frac{1}{2}} M^T, \quad p_X = (C^T C)^{-1} C^T d \tag{7} $$

in that M and C were defined as,

$$ M = \sum_{i=1}^{m} \beta_i \alpha_i^T $$

$$ C = \begin{bmatrix} I - R_{A_1} \\ \vdots \\ I - R_{A_m} \end{bmatrix}, \quad d = \begin{bmatrix} p_{A_1} - R_X p_{B_1} \\ \vdots \\ p_{A_m} - R_X p_{B_m} \end{bmatrix} $$

Wu et al. (2019) solved $AX = XB$ problem using 4D Procrustes analysis and they proposed, for the first time, to solve the Procrustes problem through eigendecomposition. Shah (2013) solved the Robot-World/hand–eye calibration problem in closed-form by exploiting the Kronecker product. The separable methods suffer from a common problem — the estimation error of R_X exacerbates the estimation of p_X (Shah et al., 2012; Zhang, 2016).

The simultaneous methods overcome this problem by solving for the rotational and translational components simultaneously. Horaud and Dornaika (1995) solved R_X and p_X simultaneously through a nonlinear technique after representing R_X as a unit quaternion. They were the first to use this nonlinear technique for simultaneous minimization of the rotation quaternion and the translation vector (Daniilidis, 1999). This approach was relatively slow.

The iterative methods solve R_X and p_X iteratively by employing optimization techniques (Shah et al., 2012; Zhang, 2016). Because of using the optimization technique, the error propagation from R_X to p_X is avoided. With real and simulated data, Tabb and Yousef (2017) compared various iterative methods with varying formulations, different parameterizations, and diverse cost functions. They proposed, for the first time, to use camera re-projection error for hand–eye calibration which prevented the artifacts and errors related to camera calibration from being propagated to the process of hand–eye calibration. However, this method is sensitive to the choice of the initial solution. To find an approximate solution to minimize the camera re-projection error, an initial solution obtained using an alternative method was used as an initial solution for their experiments. The readers are referred to Shah et al. (2012) for a more exhaustive review of the hand–eye calibration problem.

2.2. Point set matching

The problem of point set matching (or point set registration) is to estimate the homogeneous transformation matrix between two point sets, often called target points and moving points. This transformation matrix is rigid when the distance between the corresponding points is constant, otherwise, it is non-rigid (Maiseli, Gu, & Gao, 2017).

The most notable algorithm for point set matching is the Iterative Closest Point (ICP) algorithm. The seminal work of this algorithm was reported in Besl and McKay (1992) and Zhang (1994). ICP estimates the corresponding points between the points in two sets based on the distance between different points. ICP relies on the singular value decomposition (SVD) of the covariance matrix between two point sets to iteratively minimize the distance between all corresponding points. Several other variants of ICP have been reported in the literature that better estimate the correspondence and improve the computational efficiency of the algorithm (He, Liang, Yang, Li, & He, 2017; Tihonkih, Makovetskii, & Kuznetsov, 2016; Tykkälä, Audras, & Comport, 2011).

The problem of point set matching can also be considered as an estimation problem of the probability density. The Coherent Point Drift (CPD) algorithm (Myronenko & Song, 2010) was proposed within this context. CPD represents two sets of points as the centroids of two Gaussian mixture models (GMM). To estimate the homogeneous transformation matrix between the two point sets, CPD iteratively fits the GMM centroids of the moving points to that of the target points by maximizing the likelihood. The CPD algorithm forces the GMM centroids to move coherently to preserve the topology of the points. For an exhaustive review of point set matching algorithms, readers are referred to Maiseli et al. (2017).

Given the correspondence between the two sets of points, the problem of hand–eye calibration can be converted to a simplified problem of point set matching. The correspondence is required in hand–eye calibration otherwise the robot cannot reach the expected point in practice. In this study we propose a new solution for the hand–eye calibration problem on $SO(3)\times\mathbb{R}^3$ that unlike our previous approach on SE(3) (Qiu et al., 2020) has much faster convergence rate. The details of our algorithm are presented in the next section.

3. HI-SO(3)R3 algorithm

In this section, we provide the details of the new solution for the hand–eye calibration problem by considering it as a point set matching problem. The proposed solution is based on a nonlinear estimator on $SO(3)\times\mathbb{R}^3$, and it is referred to as HI-SO(3)R3.

3.1. Problem statement and objective

Let us consider two point sets, one in the camera frame and the other in the robot base frame (denoted as $\{{}^C X_i\}$ and $\{{}^B X_i\}$ ($i = 1, 2, \dots, n$), respectively, where n is the number of sample points). Let us also assume that a rigid correspondence between the points in two sets is available. Here, "rigid correspondence" has two meanings: (1) the two point sets are rigidly related, and (2) the correspondence is time-invariant. The two point sets can be related as follows,

$$ {}^B P_i = {}^B_C T \ {}^C P_i, \quad \forall i = 1, 2, \dots, n \tag{8}$$

where ${}^B_C T = \begin{bmatrix} {}^B_C R & {}^B_C p \\ \underline{0} & 1 \end{bmatrix}$ is the homogeneous transformation matrix between the robot base frame and the camera frame, in that ${}^B_C R$ and ${}^B_C p$ are the rotation matrix and translation vector of ${}^B_C T$, ${}^B P_i = \begin{bmatrix} {}^B X_i & 1 \end{bmatrix}^T$, and ${}^C P_i = \begin{bmatrix} {}^C X_i & 1 \end{bmatrix}^T$.

We denote ${}^B_C \hat{R}$ and ${}^B_C \hat{p}$ as the estimation of ${}^B_C R$ and ${}^B_C p$, respectively and define ${}^B_C \hat{T} = \begin{bmatrix} {}^B_C \hat{R} & {}^B_C \hat{p} \\ \underline{0} & 1 \end{bmatrix}$. The objective of HI-SO(3)R3 is to find the pair ${}^B_C \hat{R}$ and ${}^B_C \hat{p}$ (i.e., ${}^B_C \hat{T}$) in order to minimize the following error function,

$$ f({}^B_C \hat{T}, {}^C P, {}^B P) := \frac{1}{n} \sum_{i=1}^{n} \| {}^B_C \hat{T} \, {}^C P_i - {}^B P_i \|^2 = \frac{1}{n} \sum_{i=1}^{n} \| {}^B_C \hat{R} \ {}^C X_i + {}^B_C \hat{p} - {}^B X_i \|^2 \tag{9}$$

Moreover, using (3) and the result of (9) (i.e., ${}^B_C \hat{T}$), ${}^G_C T$ can be obtained as,

$$ {}^G_C T = {}^G_B T \ {}^B_C \hat{T} = {}^B_G T^{-1} \ {}^B_C \hat{T} \tag{10}$$

in that ${}^B_G T$ is derived from the robot kinematics. Thus, the hand–eye calibration problem for estimating ${}^G_C T$ boils down to the problem of estimating the homogeneous transformation matrix (${}^B_C T$) between the two sets of points ($\{{}^C X_i\}$ and $\{{}^B X_i\}$) within the context of hand–eye calibration.

3.2. Nonlinear estimator design

In this section, we provide the design of a nonlinear estimator to estimate ${}^B_C T$ and minimize the error function defined in (9). To this effect, we define ${}^B_C \hat{R}_k$ and ${}^B_C \hat{p}_k$ as the estimated value of these variables at the kth iteration of the algorithm. We also define the re-projection error of the ith point, e_{p_i} at the kth iteration as follows:

$$ e_{p_i} \triangleq {}^B X_i - ({}^B_C \hat{R}_k \, {}^C X_i + {}^B_C \hat{p}_k) \tag{11}$$

Given the relation between the coordinates of the points in two frames, i.e., ${}^B X_i = {}^B_C R \, {}^C X_i + {}^B_C p$ and the one given in (8), one can verify that $e_{p_i} = \underline{0}_3$ if ${}^B_C \hat{R}_k = {}^B_C R$ and ${}^B_C \hat{p}_k = {}^B_C p$. Let ${}^B X_c = \frac{1}{n} \sum_{i=1}^{n} {}^B X_i$ be the center of the point set $\{{}^B X_i\}$ with n points. We use e_{p_i} to introduce the following two innovation terms:

$$ \Delta_R \triangleq \sum_{i=1}^{n} e_{p_i} ({}^B X_i - {}^B X_c)^T \tag{12}$$

$$ \Delta_p \triangleq \sum_{i=1}^{n} e_{p_i} \tag{13}$$

We then propose the following discrete nonlinear estimator for estimating the values of ${}^B_C \hat{R}_k$ and ${}^B_C \hat{p}_k$ and its update criteria as,

$$ {}^B_C \hat{R}_{k+1} = \mathcal{R}(\Delta_R) \, {}^B_C \hat{R}_k \tag{14}$$

$$ {}^B_C \hat{p}_{k+1} = \mathcal{R}(\Delta_R) \left({}^B_C \hat{p}_k + \alpha_p \Delta_p - {}^B X_c \right) + {}^B X_c \tag{15}$$

where $\mathcal{R}(\Delta_R) = \exp\left(-\frac{\alpha_R}{2}(\Delta_R - \Delta_R^T)\right)$, and the scalars α_R and α_p are the learning rates for the estimation of ${}^B_C\hat{R}$ and ${}^B_C\hat{p}$, respectively. Our proposed solution is motivated by the continuous nonlinear observers presented in Wang and Tayebi (2019, 2020a). However unlike the continuous observers, our estimator updates discretely and the data from an IMU (inertial measurement unit) is not required. Hence, the convergence results and the selection of the gains in Wang and Tayebi (2019, 2020a) are not applicable to our method.

Although a large value of the learning rate can increase the convergence speed, the value of α_R and α_p cannot be arbitrarily selected otherwise the estimation process would not converge. To guarantee the convergence, the selection of α_R and α_p is discussed next.

3.3. Learning rate selection

3.3.1. Selecting α_R

Let us define the rotational error between the truth and the estimate of ${}^B_CR$ at the kth iteration as,

$$E_k^R = {}^B_CR\,{}^B_C\hat{R}_k^T \tag{16}$$

where ${}^B_CR$ is the truth and ${}^B_C\hat{R}_k$ is its estimate at the kth iteration. According to Wang and Tayebi (2020a), the innovation term Δ_R defined in (12) can be rewritten as $\Delta_R = (I - E_k^R)^T\mathcal{M}$ with $\mathcal{M} \triangleq \sum_{i=1}^n ({}^BX_i - {}^BX_c)({}^BX_i - {}^BX_c)^T$, and it follows from (14) and (16) that

$$E_{k+1}^R = E_k^R\mathcal{R}(\Delta_R)^T \tag{17}$$

It has been shown that Berkane and Tayebi (2019, Theorem 1), if there exist at least three non-collinear sample points in the point set $\{{}^BX_i\}$ with noise-free measurements $\{{}^CX_i\}$, then the rotation estimation error E_k^R will converge to I for any initial condition that satisfies $\mathrm{tr}(E_0^R) \neq -1$ (i.e., the angle of initial rotational error E_0^R is strictly less than 180°) when the value of α_R in the estimator in (14) is selected as,

$$0 < \alpha_R\left(tr(\mathcal{M}) - \lambda_i^{\mathcal{M}}\right) < 1,\ i = 1,2,3 \tag{18}$$

in that $tr(\mathcal{M})$ denotes the trace of $\mathcal{M}$, and $\lambda_i^{\mathcal{M}}$ is the ith eigenvalue of $\mathcal{M}$. The stability analysis for this approach has been previously reported and is not repeated here for the sake of brevity (Berkane & Tayebi, 2019, Theorem 1).

Remark 1. The condition in (18) for the scalar gain α_R was developed for the worst cases. If α_R is selected slightly larger than the bound in (18), the rotational estimation error E_k^R may still converge to I, especially when E_k^R is close to I. The experimental results will show that the proposed algorithm works when α_R is slightly larger than the bound given in (18).

Remark 2. Theoretically, in the absence of measurement noise, $\Delta_R - \Delta_R^T = \underline{0}_{3\times3}$ implies either $E_k^R = I$ or $\mathrm{tr}(E_k^R) = -1$, in that the latter at $k = 0$ may cause the convergence issue of our estimator (Wang & Tayebi, 2020a). However, due to the unavoidable measurement noise in practice, the estimation error E_k^R with $\mathrm{tr}(E_0^R) = -1$ will leave the undesired equilibrium point and converge to I.

3.3.2. Selecting α_p

Let us define the translational error between the truth and the estimate of ${}^B_Cp$ at the kth iteration as,

$$E_k^p = {}^B_Cp - {}^BX_c - E_k^R({}^B_C\hat{p}_k - {}^BX_c) \tag{19}$$

where ${}^BX_c$ is the center of the point set $\{{}^BX_i\}$, ${}^B_Cp$ is the truth translation vector, and ${}^B_C\hat{p}_k$ is its estimated value at the kth iteration. Using (13), (16) and (19), one can rewrite Δ_p in terms of estimation errors E_k^R and E_k^p as,

$$\Delta_p = n(E_k^R)^T E_k^p \tag{20}$$

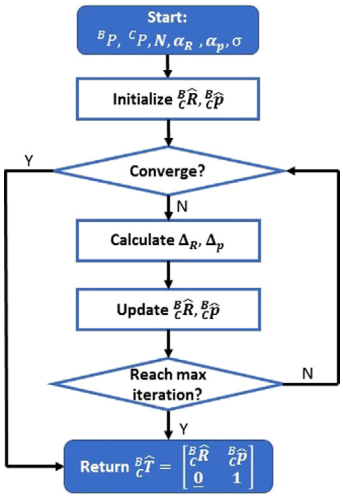

Fig. 2. Flow chart of the proposed HI-SO(3)R3 algorithm.

where n is the number of points in the point set. From (14)–(16), (19) and (20), one can show that,

$$\begin{aligned} & E_{k+1}^p = {}^B_Cp - {}^BX_c - E_{k+1}^R({}^B_C\hat{p}_{k+1} - {}^BX_c) \\ \Leftrightarrow\ & E_{k+1}^p = E_k^p - \alpha_pE_k^R\Delta_p \\ \Leftrightarrow\ & E_{k+1}^p = E_k^p - n\alpha_pE_k^R(E_k^R)^TE_k^p \\ \Leftrightarrow\ & E_{k+1}^p = E_k^p - n\alpha_pE_k^p \\ \Leftrightarrow\ & E_k^p = (1 - n\alpha_p)^kE_0^p \end{aligned} \tag{21}$$

where E_0^p denotes the initial translational error (at $k = 0$). It is obvious from (21) that E_k^p will exponentially converge to $\underline{0}_3$ with noise-free measurements if α_p is selected as

$$0 < n\alpha_p < 1. \tag{22}$$

It is worth to note that α_p and α_R including the matrix $\mathcal{M}$ and $\lambda_i^{\mathcal{M}}(i = 1,2,3)$ (the eigenvalues of $\mathcal{M}$) are calculated before conducting the iterative estimation of the proposed algorithm (i.e., they are calculated off-line).

Remark 3. If α_p does not satisfy the bound in (22), E_k^p may still converge but the convergence is not guaranteed (Wang & Tayebi, 2020a). The experimental results will show that the proposed algorithm works if α_p is selected slightly larger than the bound given in (22).

Remark 4. As seen in (17) and (21) the convergence of the estimation errors E_k^R and E_k^p are independent. In other words, the convergence time and accuracy of E_k^R and E_k^p will not be affected by each other. This decoupling property results from the special design of the innovation terms Δ_R and Δ_p. This decoupling property results in a dramatic reduction in the algorithm convergence time. Note also that $E_k^R = {}^B_CR\,{}^B_C\hat{R}_k^T = I$ and $E_k^p = {}^B_Cp - {}^BX_c - E_k^R({}^B_C\hat{p}_k - {}^BX_c) = \underline{0}_3$ imply that ${}^B_C\hat{R}_k = {}^B_CR$ and ${}^B_C\hat{p}_k = {}^B_Cp$, respectively. Consequently, we obtain ${}^B_C\hat{T}_k = {}^B_CT$.

3.4. Proposed algorithm

Having introduced the nonlinear estimator and the requirements for selecting the learning rates, the flow chart of our proposed algorithm, HI-SO(3)R3, is given in Fig. 2 followed by the details of the algorithm.

The initial value of ${}^B_C\hat{R}$ can be randomly chosen in SO(3) and the initial value of ${}^B_C\hat{p}$ can be chosen to be $\underline{0}_3$. Alternatively, ${}^B_C\hat{R}$ and ${}^B_C\hat{p}$ can be initialized by singular value decomposition (SVD). A simple method is applied to terminate iterations appropriately. Considering the recent

three iterations, we calculate the average change of the rotation matrix, "dR" and the translation vector "dp" to determine the algorithm termination. The algorithm will stop either when both "dR" and "dp" are below a given tolerance "σ", or when the maximum number of iteration N is surpassed. The following pseudo-codes provides more details regarding the algorithm.

Algorithm The HI-SO(3)R3 algorithm

Input: two point sets ($\{{}^{B}X_i\}$ and $\{{}^{C}X_i\}$, i = 1, 2,…, n) with established correspondence, maximum iteration (N), learning rates (α_R and α_p), the tolerance (σ), initial guess (R_0 and p_0).

Output: ${}^{B}_{C}\hat{T}$ (the estimated transformation matrix between the two input point sets).

Initialization: k = 0, ${}^{B}_{C}\hat{R}_k = R_0$, ${}^{B}_{C}\hat{p}_k = p_0$

${}^{B}X_c = \frac{1}{n}\sum_{i=1}^{n} {}^{B}X_i$

repeat

  Convergence detection: {

  q_k = rot2qua(${}^{B}_{C}\hat{R}_k$) ▷ Convert ${}^{B}_{C}\hat{R}_k$ into quaternion

  if $0 < k < 3$ **then**

    $dR = \frac{1}{k+1}\sum_{j=0}^{k} \text{acos}(\frac{q_j \cdot q_{j+1}}{\|q_j\|\|q_{j+1}\|})$,

    $dp = \frac{1}{k+1}\sum_{j=0}^{k} \|{}^{B}_{C}p_j - {}^{B}_{C}p_{j+1}\|$

  end if

  if $k \geq 3$ **then**

    $dR = \frac{1}{3}\sum_{j=k-3}^{k-1} \text{acos}(\frac{q_j \cdot q_{j+1}}{\|q_j\|\|q_{j+1}\|})$

    $dp = \frac{1}{3}\sum_{j=k-3}^{k-1} \|{}^{B}_{C}p_j - {}^{B}_{C}p_{j+1}\|$

  end if

  if $dR < \sigma$ and $dp < \sigma$ **then**

    Return ${}^{B}_{C}\hat{T}_k = \begin{bmatrix} {}^{B}_{C}\hat{R}_k & {}^{B}_{C}\hat{p}_k \\ \underline{0} & 1 \end{bmatrix}$

  end if}

  $\Delta_R = 0$, $\Delta_p = 0$

  while $1 \leq i \leq n$ **do**

    ${}^{B}_{C}\hat{R}_k \, {}^{C}X_i + {}^{B}_{C}\hat{p}_k)$

    $\Delta_p = \Delta_p + e_{p_i}$

    $\Delta_R = \Delta_R + e_{p_i}({}^{B}X_i - {}^{B}X_c)$

  end while

  $\mathcal{R}(\Delta_R) = \exp\left(-\frac{\alpha_R}{2}(\Delta_R - \Delta_R^T)\right)$

  ${}^{B}_{C}\hat{R}_{k+1} = \mathcal{R}(\Delta_R)\,{}^{B}_{C}\hat{R}_k$

  ${}^{B}_{C}\hat{p}_{k+1} = \mathcal{R}(\Delta_R)\left({}^{B}_{C}\hat{p}_k + \alpha_p\Delta_p - {}^{B}X_c\right) + {}^{B}X_c$

  k = k + 1

until k = N

return ${}^{B}_{C}\hat{T}_k = \begin{bmatrix} {}^{B}_{C}\hat{R}_k & {}^{B}_{C}\hat{p}_k \\ \underline{0} & 1 \end{bmatrix}$

4. Experimental results

This section outlines the experiments designed to demonstrate the performance of HI-SO(3)R3 in comparison with some other hand–eye calibration algorithms and point set matching algorithms.

4.1. Hardware setup

We used a KUKA Light-Weight Robot (LWR) IV to conduct our experiments. A special fixture that included a screw with a sharp tip was fabricated and attached to the KUKA robot. The fixture was designed such that the tip of the screw coincided with the gripper frame. The calibration device used in these experiments was a typical checkerboard pattern containing 54 corners. These components are shown in Fig. 3. We used two different stereo cameras, Intel RealSense D435 and KYT-U100-960R301, also shown in Fig. 3 for conducting our experiments. The Intel camera has better resolution than the KYT camera. All statistics were collected using MATLAB r2019b on a personal computer powered by an i5-6400 CPU with 16 GB RAM.

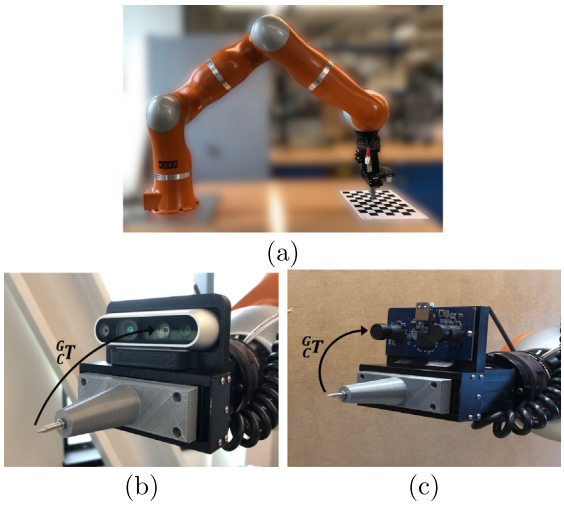

Fig. 3. Experimental setup. (a) The robot is touching a corner on the checkerboard. (b) The screw with Intel RealSense D435. (c) The screw with the KYT stereo camera.

4.2. Experiment data collection

4.2.1. Conventional formulation

At the outset of our experimental studies, we aimed at evaluating the performance of some of the conventional methods of hand–eye calibration in comparison to our proposed algorithm. To this end, we captured 100 different images of the checkerboard pattern. During image acquisition, we also recorded the gripper poses (i.e., position and orientation) using the internal encoders of the KUKA robot. The A and B matrix in (1) were calculated as $A = A_2^{-1}A_1 = {}^{B}_{G}T_2^{-1}\,{}^{B}_{G}T_1$ and $B = B_2B_1^{-1} = {}^{C}_{W}T_2\,{}^{C}_{W}T_1^{-1}$, respectively, where ${}^{B}_{G}T$ as defined previously is the transformation matrix between the base of the KUKA robot and the gripper and ${}^{C}_{W}T$ is the transformation matrix between the camera's and the world's frame. Different implementations of the traditional methods reported in Horaud and Dornaika (1995), Park and Martin (1994), Shah (2013) and Wu et al. (2019) were used for this evaluation.

4.2.2. Point set matching formulation

We also examined and compared the performance of our algorithm with some widely used point set matching algorithms. The coordinates of the checkerboard corners expressed in the robot base frame were used as the target points. The coordinates of the target points were collected when the tip of the screw made physical contact with each corner as shown in Fig. 3(a). Referring to (8), these coordinates constituted $\{{}^{B}P\}$. By taking advantage of the stereo camera, we estimated the checkerboard corners' 3D coordinates expressed in the camera frame from the images acquired previously. These coordinates constituted $\{{}^{C}P\}$. The collected data (i.e.$\{{}^{B}P\}$ and $\{{}^{C}P\}$) was used to compare the performance of HI-SO(3)R3, GD-SE(3), ICP, and CPD algorithms. To this effect, 100 sets of $\{{}^{C}P\}$ were obtained among which 70% were used for training and the remaining 30% were used for validation.

4.3. Performance evaluation

The performance of the implemented algorithms was assessed and compared using the previously acquired data.

4.3.1. Training phase

During the training phase, we used the obtained values of ${}^{B}_{G}T$ and ${}^{C}_{W}T$ matrices (i.e., A and B matrices) and several conventional algorithms (Horaud & Dornaika, 1995; Park & Martin, 1994; Shah, 2013; Wu et al., 2019) to estimate ${}^{G}_{C}T$ (i.e., X matrix). We also used $\{{}^{B}P\}$ and $\{{}^{C}P\}$ to estimate ${}^{B}_{C}T$ as per (8) by implementing ICP (Besl & McKay,

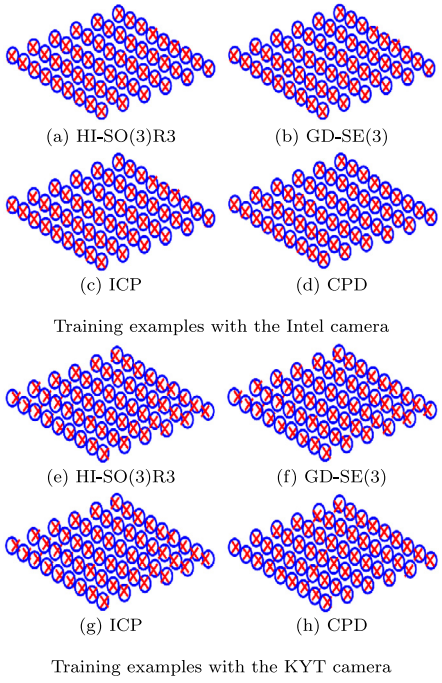

Fig. 4. Training examples: estimated points ($\{{}^{B}\hat{P}\}$, denoted by red crosses) vs. the truth points ($\{{}^{B}P\}$, denoted by blue circles).

1992), CPD (Myronenko & Song, 2010), GD-SE(3) (Qiu et al., 2020) and HI-SO(3)R3 algorithms. We then used the estimated value of ${}^{B}_{C}\hat{T}$ to obtain ${}^{G}_{C}T$ as per (10). To reduce the influence of the measurement noises, the estimation of ${}^{G}_{C}T$ was obtained through averaging the results from the training data sets.

As for parameter selection in these algorithms, we selected the parameters that achieved the best performance of each implemented algorithm. More specifically, for the algorithms that required an initial solution (e.g., ICP, GD-SE(3), and HI-SO(3)R3), the initial rotation matrix R_0 was randomly generated in SO(3) and the initial translation vector p_0 was set to $\underline{0}_3$. The learning rates for HI-SO(3)R3 were selected as $\alpha_R = \frac{1.5}{tr(\mathcal{M})-\lambda^{\mathcal{M}}_{min}}$ and $\alpha_P = \frac{0.9}{n}$ in that $\lambda^{\mathcal{M}}_{min}$ is the minimum eigenvalue of the $\mathcal{M}$ matrix and n is the number of points. The tolerance σ for detecting the convergence of the algorithm and the maximum number of iteration (N) in HI-SO(3)R3 and ICP were chosen to be 1×10^{-4} and 1×10^{4}, respectively. To obtain the same accuracy, σ and N used in GD-SE(3) were selected to be 1×10^{-7} and 1×10^{6}, respectively. The aforementioned parameters were the same with different cameras. To achieve its best performance, the d parameter in Wu's method (Wu et al., 2019) was chosen as 1×10^{14} and 1×10^{6} for the tests with Intel camera and the KYT camera, respectively.

The correspondence between $\{{}^{C}P\}$ and $\{{}^{B}P\}$ is a necessary component of HI-SO(3)R3, GD-SE(3), and ICP algorithms whether it is known or is calculated as part of the algorithm. To shed light on the effect of known vs. calculated correspondence, we used the CPD algorithm to estimate the correspondence between the points as part of the algorithm. Comparing the results from these algorithms will demonstrate the importance of the correspondence between two point sets in the hand–eye calibration scenario.

4.3.2. Validation phase

The implemented algorithms were assessed and compared using the validation data sets. To this end, the reconstruction accuracy error (RAE) and the root mean error of the combined rotation and translation error (RMCE) were used to compare the results. RAE is calculated as the RMSE (root mean square error) of the Euclidean distance between the estimated points (i.e., $\{{}^{B}\hat{P}\}$) and the truth points (i.e., $\{{}^{B}P\}$) as,

$$RAE = \left(\frac{1}{n}\sum_{i=1}^{n}\| {}^{B}\hat{P}_i - {}^{B}P_i\|^2\right)^{\frac{1}{2}} \tag{23}$$

where n is the number of points in the validation set. Also, RMCE is defined as,

$$RMCE = \frac{1}{m}\left(\sum_{i=1}^{m}\|A_iX - XB_i\|^2\right)^{\frac{1}{2}} \tag{24}$$

where A_i and B_i are the robot and camera motion matrices described in (1), and m is the number of matrices in the validation set.

4.4. Results and discussion

In this section, the results from all experiments are provided and compared.

4.4.1. Results of the training phase

We conducted the training phase following the procedure described in Section 4.3.1. The training results for HI-SO(3)R3, GD-SE(3), ICP, and CPD using two different cameras (Intel RealSense and KYT) are shown in Fig. 4. The training errors and the training time of these algorithms are listed in Tables 1 and 2. As shown in Fig. 4, $\{{}^{B}\hat{P}\}$ (the estimated points) obtained using these algorithms seems to match well with $\{{}^{B}P\}$ (the truth points). However, for the CPD algorithm, its RAE value is much larger than other algorithms. The reason is that the CPD algorithm cannot estimate the correspondence between $\{{}^{C}P\}$ and $\{{}^{B}P\}$ effectively. This problem was exacerbated because of the rectangular arrangement of $\{{}^{C}P\}$ and $\{{}^{B}P\}$, which made the CPD algorithm more-likely to ill-estimate the correspondence between the two sets of points.

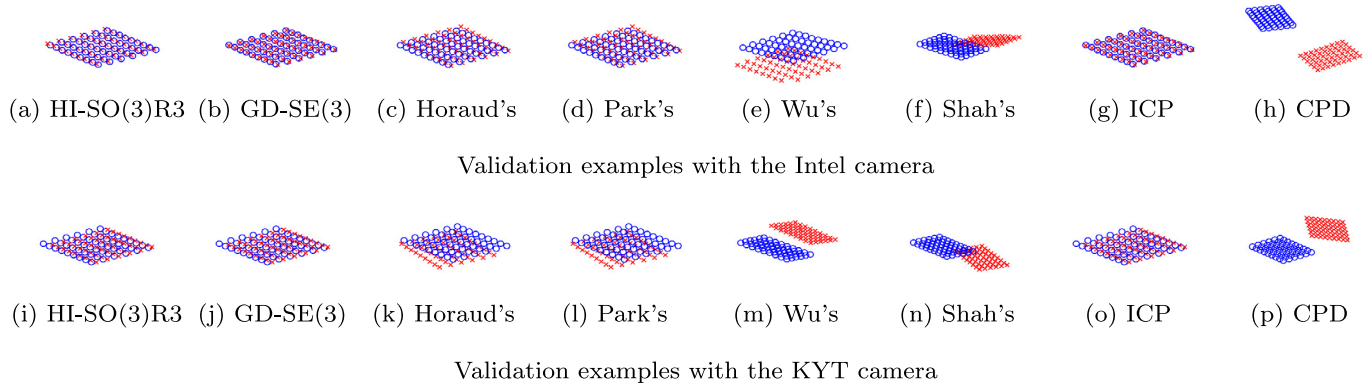

Fig. 5. Validation examples with different cameras: $\{{}^{B}\hat{P}\}$ (estimated points, denoted by red crosses) vs. $\{{}^{B}P\}$ (truth points, denoted by blue circles).

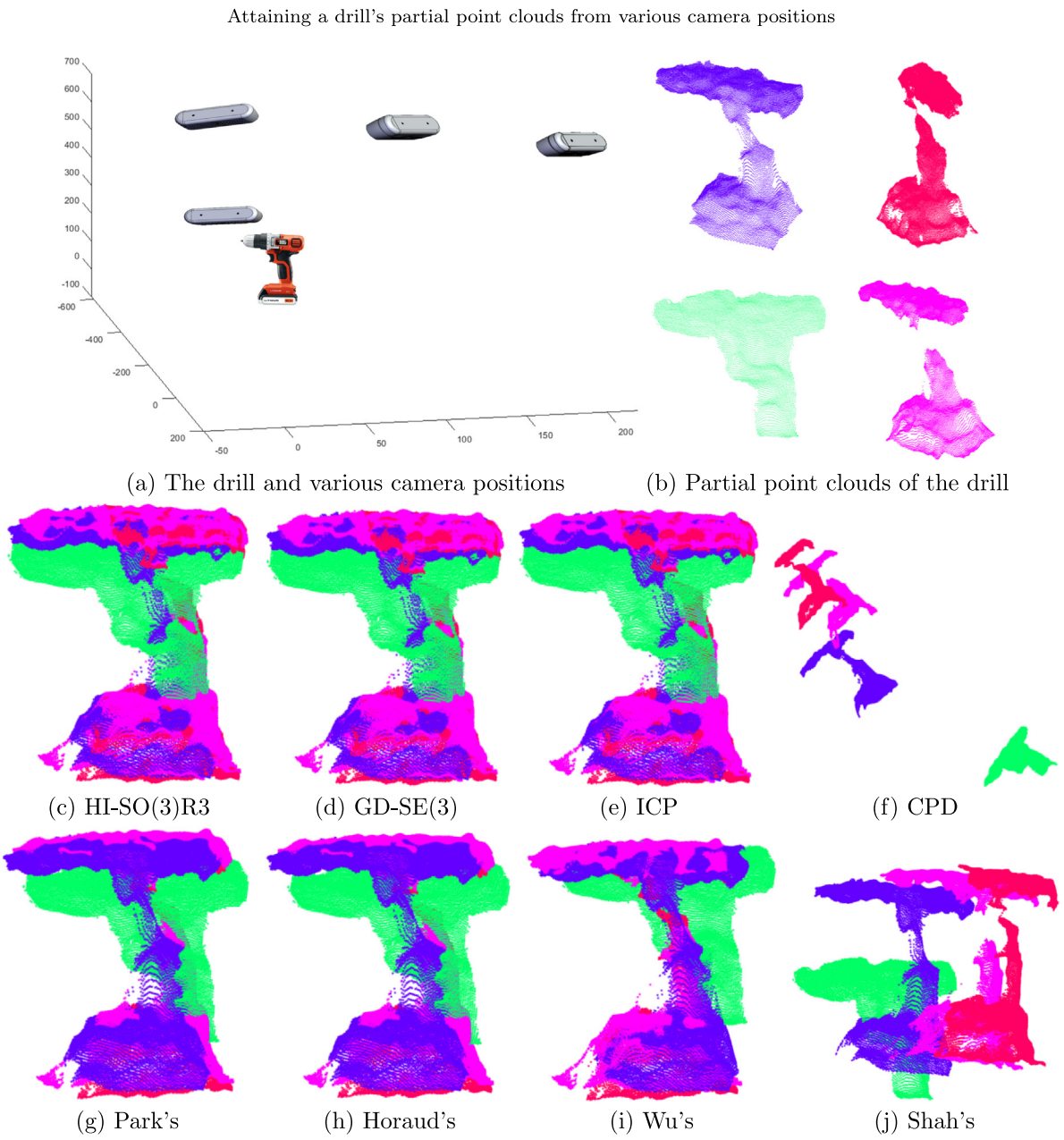

Fig. 6. Point cloud reconstruction using hand–eye calibration results.

Table 1
Experimental results with the Intel RealSense camera.

Methods		Average RAE In training (mm)	Training time (s)	Average RAE In validation (mm)	RMCE
HI-SO(3)R3		0.9436	**0.3368**	4.2517	1.1×10^{-3}
GD-SE(3) (Qiu et al., 2020)		0.9436	121.29	4.2516	1.1×10^{-3}
Traditional algorithms	Horaud's method (Horaud & Dornaika, 1995)	–	0.0083	30.273	1.1×10^{-3}
	Park's method (Park & Martin, 1994)	–	0.0042	30.273	1.1×10^{-3}
	Wu's method (Wu et al., 2019)	–	0.0024	34.724	1.1×10^{-3}
	Shah's method (Shah, 2013)	–	0.0026	134.06	1.3×10^{-3}
Point set matching algorithms	ICP (Besl & McKay, 1992)	0.9436	0.5711	4.2517	1.1×10^{-3}
	CPD (Myronenko & Song, 2010)	102.97	2.0650	204.92	6.9×10^{-3}

It is clear that the HI-SO(3)R3 algorithm outperformed all other point set matching algorithms with respect to the training time. In comparison to our previous algorithm GD-SE(3), the speedup achieved in HI-SO(3)R3 is due to the fact that the rotational estimation error is decoupled from the translational estimation error as discussed in Section 3.3. Even though the ICP algorithm required less number of

Table 2
Experimental results with the KYT camera.

Methods		Average RAE In training (mm)	Training time (s)	Average RAE In validation (mm)	RMCE
HI-SO(3)R3		1.7002	**0.3643**	20.4847	2.4×10^{-3}
GD-SE(3) (Qiu et al., 2020)		1.7001	229.60	20.4808	2.4×10^{-3}
Traditional algorithms	Horaud's method (Horaud & Dornaika, 1995)	–	0.0110	35.4475	2.4×10^{-3}
	Park's method (Park & Martin, 1994)	–	0.0090	35.4484	2.4×10^{-3}
	Wu's method (Wu et al., 2019)	–	0.0026	118.575	2.4×10^{-3}
	Shah's method (Shah, 2013)	–	0.0028	162.282	2.4×10^{-3}
Point set matching algorithms	ICP (Besl & McKay, 1992)	1.7001	0.6183	20.4837	2.4×10^{-3}
	CPD (Myronenko & Song, 2010)	60.357	2.3733	267.5175	7.8×10^{-3}

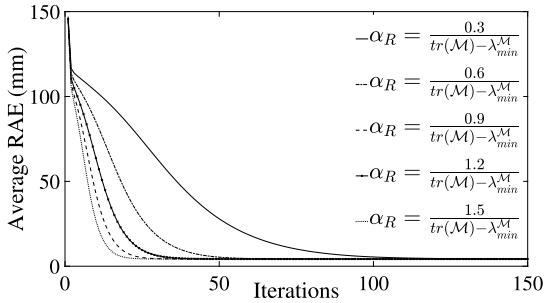

Fig. 7. Performance of HI-SO(3)R3 for different α_R values and $\alpha_p = \frac{0.9}{n}$.

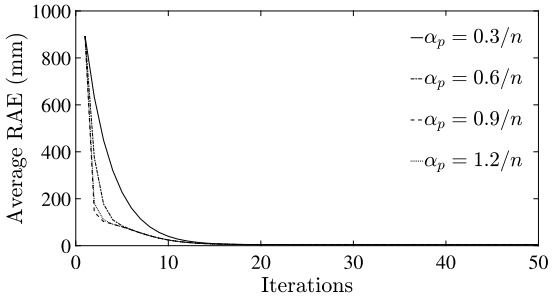

Fig. 8. Performance of HI-SO(3)R3 for different α_p values and $\alpha_R = \frac{1.5}{tr(\mathcal{M})-\lambda^{\mathcal{M}}_{min}}$.

iterations, it took longer to train due to its reliance on the singular value decomposition (SVD) for estimating the rotation matrix in each iteration. It is also important to note that the implemented conventional algorithms (Horaud & Dornaika, 1995; Park & Martin, 1994; Shah, 2013; Wu et al., 2019) required much less training time. This is due to the fact that these algorithms use closed-form solutions for the results. The trade-off here was in much larger validation error (RAE) values as shown in the results of these algorithms.

4.4.2. Result of the validation phase

After the training phase, the validation phase was conducted following the procedure described in Section 4.3.2. Examples of the validation results obtained from different implemented algorithms are shown in Fig. 5 and the results are listed in the last two columns of Tables 1 and 2. As seen, the results are consistent with those obtained during the training phase. Once again, the degraded performance of the CPD algorithm highlights the adverse effect of not having the correct correspondence between the points. Another important observation is the inconsistent outcome seen in the values of RMCE vs. RAE. The RMCE cannot represent the effectiveness of various methods and separate the results properly. On the other hand, the RAE clearly shows the superiority of point set matching algorithms to traditional methods. This is due to the fact that the point set matching algorithms benefit from the availability of the point correspondence. The conventional algorithms, on the other hand, define a cost function such as $\frac{1}{n}\sum_{i=1}^{n}\|A_iX - XB_i\|^2$ to solve for X through optimization. In reality, such optimization, as also seen in our result in Tables 1 and 2 does not necessarily minimize the RAE values.

It should be pointed out that Shah's method (Shah, 2013) is designed for robot-world and hand–eye calibration using $AX = YB$ formulation to provide a solution for both X and Y. The results in Tables 1 and 2 only present the accuracy of the X matrix for this algorithm. Since in Shah's algorithm, both X and Y matrices are estimated together, the errors from one matrix propagate into another matrix resulting in the least accurate results among conventional methods.

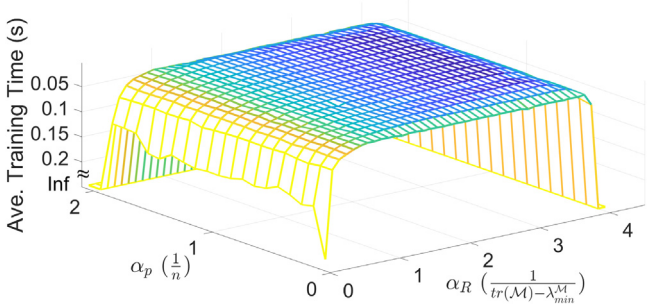

Fig. 9. Average training time (s) of HI-SO(3)R3 with different values of α_R and α_p (The darker the color is, the less the training time).

It is clear that our proposed algorithm, in comparison to other state-of-the-art algorithms, offers a much better computation speed without compromising the accuracy of the results.

To further assess the performance of implemented methods, we applied the hand–eye calibration results (i.e., the estimated transformation matrix from the gripper frame to the camera frame, ${}^{G}_{C}\hat{T}$) to reconstruct a complete point cloud of a power drill from its partial point clouds (see Fig. 6). Point Cloud Reconstruction (also known as Point Cloud Stitching) is the process of constructing a complete point cloud of a scene/object through combining/stitching several partial point clouds belonging to the same scene/object. As shown in Fig. 6(a)(b), partial point clouds of a power drill were acquired with various camera poses using the Intel RealSense stereo camera. From ${}^{G}_{C}\hat{T}$ and ${}^{B}_{G}T$ (computed from robot kinematics), the estimated transformation matrix from the robot base frame to the camera frame (i.e., ${}^{B}_{C}\hat{T}$) can be calculated as ${}^{B}_{C}\hat{T} = {}^{B}_{G}T\,{}^{G}_{C}\hat{T}$. Using ${}^{B}_{C}\hat{T}$, we can convert the partial point clouds represented in different camera frames into the same robot base frame. If ${}^{G}_{C}\hat{T}$ is estimated accurately, we can attain a more complete point cloud with high quality expressed in the robot base frame. As shown in Fig. 6, the methods with lower RAE values perform better than those with higher RAE values.

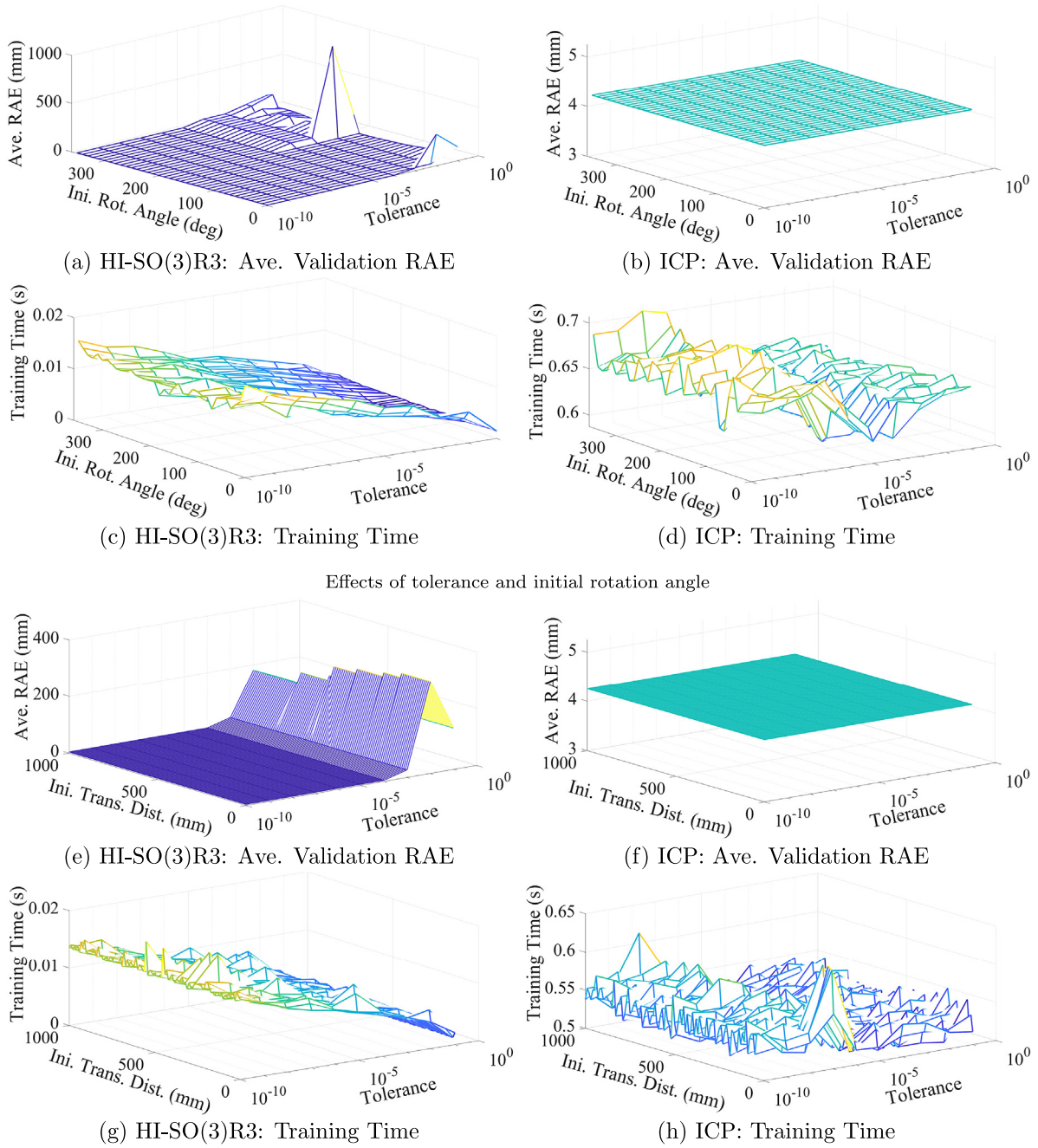

Fig. 10. Effects of the selection of tolerance and initial guess to HI-SO(3)R3 and ICP.

To show the effect of α_R and α_p, additional experiments were conducted using HI-SO(3)R3 with the data acquired from the Intel RealSense stereo camera. The results are shown in Figs. 7, 8, and 9. Figs. 7 and 8 show that larger values of α_R and α_p result in faster convergence as long as these values satisfy the conditions described in (18) and (22). Fig. 9 shows the average training time of HI-SO(3)R3 with different choices of learning rates while achieving the same accuracy as that listed in Table 2. The infinite training time in Fig. 9 means HI-SO(3)R3 did not converge using the selected learning rate. As mentioned in Section 3.3, with slightly larger learning rates α_R, α_p than the bounds given in (18) and (22), the estimated error may still converge to its minimum (see Figs. 7, 8, and 9), but the convergence is not guaranteed for other systems. Hence, α_R and α_p should be selected according to (18) and (22) when no prior information is available.

To show the effect of the selection of tolerance and initial guess, additional experiments were conducted using HI-SO(3)R3 and ICP with the data acquired using the Intel RealSense stereo camera. In these experiments, the maximum number of iterations for HI-SO(3)R3 and ICP were chosen to be 1×10^4. The learning rates in HI-SO(3)R3 were selected as $\alpha_R = \frac{1.5}{tr(\mathcal{M}) - \lambda^{\mathcal{M}}_{min}}$ and $\alpha_p = \frac{0.9}{n}$. During these experiments, the initial rotation matrix was obtained from various rotation angles (ranging from 0° to 350°) around the rotation axis computed from the best calibration results of the previous tests. Similarly, the initial translation vector was obtained from various translation distances (ranging

Table 3
The performance of HI-SO(3)R3 and ICP with SVD initialization.

	HI-SO(3)R3	ICP
Training time (s)	0.001044	0.4724
Ave. RAE (mm)	4.2517	4.2517

from 0 to 1000 mm) along the translation direction computed from the best calibration results of the previous tests. The experiments regarding different tolerances and initial guesses were divided into two parts: (1) the experiments about various tolerances and initial rotation angles with the initial translation vector being $\underline{0}_3$, and (2) the experiments about various tolerances and initial translation distances with the initial rotation matrix being I. The results from these experiments are shown in Fig. 10. Although the accuracy of ICP was less sensitive to the selection of tolerance and initial guess, its training time was much longer than that of HI-SO(3)R3. On the other hand, the accuracy of HI-SO(3)R3 is sensitive to the selection of tolerance and the best result was obtained when the tolerance was chosen to be less or equal to 1×10^{-5} for both the rotation matrix and translation vector. Hence the best value of tolerance for HI-SO(3)R3 is suggested to be 1×10^{-5} to achieve the best accuracy and computational efficiency. As for the selection of the initial guess, however, no clear relation between the initial guess and the accuracy and efficiency of HI-SO(3)R3 can be deducted from Fig. 10. In practice, using singular value decomposition (SVD) is a common approach to estimate the rigid transformation matrix between two sets of points. When the initial guess was computed using SVD, the performances of HI-SO(3)R3 and ICP with the data acquired from the Intel RealSense stereo camera are listed in Table 3. During the tests with SVD initialization, the tolerance was set to 1×10^{-5} for both the rotation matrix and the translation vector, and the learning rates of HI-SO(3)R3 were $\alpha_R = \frac{1.5}{tr(\mathcal{M})-\lambda^{\mathcal{M}}_{min}}$ and $\alpha_p = \frac{0.9}{n}$. With SVD initialization, the initial guess is expected to be close to the optimal solution. In this case, the computational efficiency of HI-SO(3)R3 is even more superb compared to the ICP algorithm.

Remark 5. To achieve the best accuracy and computational efficiency, the suggested tolerance for HI-SO(3)R3 algorithm is 1×10^{-5} and HI-SO(3)R3 algorithm is best to be initialized using the SVD method.

5. Conclusions

In this paper, we formulated and solved the hand–eye calibration problem as a problem of point set matching. In this light, we proposed a new algorithm on manifold $SO(3)\times\mathbb{R}^3$ with a nice decoupling property between the rotational estimation error and translational estimation error. As a result, the convergence speed of the algorithm was significantly increased. We called our algorithm HI-SO(3)R3 for its **hi**gh convergence speed. The performance of HI-SO(3)R3 was evaluated and compared with some conventional algorithms for hand–eye calibration and some widely used point set matching algorithms. The results manifested that better accuracy in practice could be achieved by formulating the problem of hand–eye calibration as a problem of point set matching. HI-SO(3)R3 offers a superior and convenient alternative to conduct hand–eye calibration for robot manipulators. The experimental results also highlighted a drawback of the algorithm that the robot is required to physically move and make contact with the calibration object to attain the coordinates in the robot frame. Our future work will focus on eliminating this requirement and also solve the robot-world/hand–eye calibration problem (i.e., $AX = YB$).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Allibert, G., Hua, M.-D., Krupínski, S., & Hamel, T. (2019). Pipeline following by visual servoing for autonomous underwater vehicles. *Control Engineering Practice*, *82*, 151–160.

Berkane, S., & Tayebi, A. (2019). Attitude estimation with intermittent measurements. *Automatica*, *105*, 415–421.

Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures, Vol. 1611* (pp. 586–607). International Society for Optics and Photonics.

Bhargavapuri, M., Shastry, A. K., Sinha, H., Sahoo, S. R., & Kothari, M. (2019). Vision-based autonomous tracking and landing of a fully-actuated rotorcraft. *Control Engineering Practice*, *89*, 113–129.

Bishop, B. E., & Spong, M. W. (1999). Adaptive calibration and control of 2D monocular visual servo systems. *Control Engineering Practice*, *7*(3), 423–430.

Daniilidis, K. (1999). Hand-eye calibration using dual quaternions. *International Journal of Robotics Research*, *18*(3), 286–298.

Figueroa, F., & Mahajan, A. (1994). A robust navigation system for autonomous vehicles using ultrasonics. *Control Engineering Practice*, *2*(1), 49–59.

Harris, C., & Teeder, A. (1993). Geometric camera calibration for vision-based navigation. *IFAC Proceedings Volumes*, *26*(1), 77–82.

He, Y., Liang, B., Yang, J., Li, S., & He, J. (2017). An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. *Sensors*, *17*(8), 1862.

Horaud, R., & Dornaika, F. (1995). Hand-eye calibration. *The International Journal of Robotics Research*, *14*(3), 195–210.

Maiseli, B., Gu, Y., & Gao, H. (2017). Recent developments and trends in point set registration methods. *Journal of Visual Communication and Image Representation*, *46*, 95–106.

Myronenko, A., & Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(12), 2262–2275.

Park, F. C., & Martin, B. J. (1994). Robot sensor calibration: solving ax = xb on the euclidean group. *IEEE Transactions on Robotics and Automation*, *10*(5), 717–721.

Qiu, S., Wang, M., & Kermani, M. R. (2020). A new formulation for hand-eye calibrations as point set matching. *IEEE Transactions on Instrumentation and Measurement*, *69*(9), 6490–6498.

Shah, M. (2013). Solving the robot-world/hand-eye calibration problem using the kronecker product. *Journal of Mechanisms and Robotics*, *5*(3).

Shah, M., Eastman, R. D., & Hong, T. (2012). An overview of robot-sensor calibration methods for evaluation of perception systems. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems* (pp. 15–20). ACM.

Shiu, Y. C., & Ahmad, S. (1989). Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax = xb. *IEEE Transactions on robotics and automation*, *5*(1), 16–29.

Tabb, A., & Yousef, K. M. A. (2017). Solving the robot-world hand-eye (s) calibration problem with iterative methods. *Machine Vision and Applications*, *28*(5–6), 569–590.

Tihonkih, D., Makovetskii, A., & Kuznetsov, V. (2016). A modified iterative closest point algorithm for shape registration. In *Applications of Digital Image Processing XXXIX, Vol. 9971* (p. 99712D). International Society for Optics and Photonics.

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

Tykkälä, T., Audras, C., & Comport, A. I. (2011). Direct iterative closest point for real-time visual odometry. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (pp. 2050–2056). Citeseer.

Wang, M., & Tayebi, A. (2019). Hybrid pose and velocity-bias estimation on se(3) using inertial and landmark measurements. *IEEE Transactions on Automatic Control*, *64*(8), 3399–3406.

Wang, M., & Tayebi, A. (2020a). Hybrid nonlinear observers for inertial navigation using landmark measurements. *IEEE Transactions on Automatic Control*, http://dx.doi.org/10.1109/TAC.2020.2972213.

Wang, M., & Tayebi, A. (2020b). Nonlinear state estimation for inertial navigation systems with intermittent measurements. *Automatica*, *122*, Article 109244.

Wu, J., Sun, Y., Wang, M., & Liu, M. (2019). Hand-eye calibration: 4D procrustes analysis approach. *IEEE Transactions on Instrumentation and Measurement*, *69*(6), 2966–2981.

Xiao, D., Song, M., Ghosh, B. K., Xi, N., Tarn, T. J., & Yu, Z. (2004). Real-time integration of sensing, planning and control in robotic work-cells. *Control Engineering Practice*, *12*(6), 653–663.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, *13*(2), 119–152.

Zhang, Z.-Q. (2016). Cameras and inertial/magnetic sensor units alignment calibration. *IEEE Transactions on Instrumentation and Measurement*, *65*(6), 1495–1502.