

A New Formulation for Hand-Eye Calibrations as Point Set Matching

Shuwei Qiu, Miaomiao Wang, and Mehrdad R. Kermani

Abstract—Conventional methods formulate the calibration problem between a robot hand and a camera, also known as hand-eye-calibration problem, as $AX = XB$. However, in practice, these methods have limited accuracy as will be shown in our experimental results. In this paper we formulated the hand-eye calibration problem as a point set matching problem and propose a new approach to solve this problem. The proposed approach is particularly suitable for robotic applications and offers good accuracy. We obtain a solution for the said problem using gradient descent (GD) technique on the Special Euclidean group $SE(3)$. We call this approach GD-SE(3). To prove the validity of the proposed approach and to demonstrate its advantages, experimental results are provided where we compare the performance of GD-SE(3) with both conventional solutions for hand-eye calibration problem as well as those based on point set matching. The results show that the accuracy of GD-SE(3) is comparable with those based on other point-set matching algorithms. Yet, it outperforms conventional formulations based on $AX = XB$ while offering a more suitable approach for the hand-eye calibration problem of robot manipulators.

Index Terms—Hand-Eye Calibration; $AX = XB$; point set matching; Gradient Descent; Special Euclidean Group $SE(3)$

I. INTRODUCTION

Vision cameras are among the most common sensors for robotic applications such as object detection [1][2], pose estimation [3][4][5], and simultaneous localization and mapping (SLAM) [6][7][8]. To effectively use cameras, it is necessary to calibrate the camera and this topic has been rigorously studied in the literature [9][10][11]. A camera can be mounted near the robot (eye-to-hand) or on the robot as part of the robot end-effector (eye-in-hand). Regardless of the camera's placement, the location of the target object is initially obtained within the camera's coordinate system. It is necessary to represent the information within the robot's coordinate system. This requires the estimation of the relationship (transformation matrix) between the two coordinate systems. A *precise estimation* of such a transformation matrix for robot manipulators, a.k.a. hand-eye calibration, is a challenging task [12]. This problem presents itself in other multi-sensor calibration problems in applications such as integrated 3D scanning system [13], cameras and inertial/magnetic sensors calibration [14], visual-inertial integration for human navigation [15], and estimation of head-eye parameters of bionic eyes [16].

Shuwei Qiu, Miaomiao Wang, and Mehrdad R. Kermani are with Faculty of Electrical and Computer Engineering, Western University, London, Ontario, Canada sqiu47@uwo.ca, mwang448@uwo.ca, mkerman2@uwo.ca

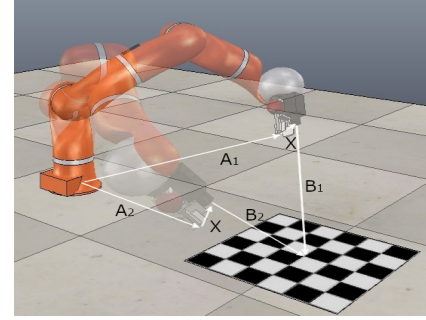


Fig. 1: A typical configuration for hand-eye calibration

The hand-eye calibration problem is conventionally formulated as $AX = XB$ [17]. A typical configuration for hand-eye calibration is shown in Fig. 1 where A_i ($i = 1, 2$) represents the transformation matrix from the robot base to the robot gripper for each pose, B_i ($i = 1, 2$) represents the transformation matrix from the camera frame to the world's frame attached to the calibration pattern, and X represents the unknown transformation matrix from the robot gripper to the camera. In this scenario, one can easily show that

$$A_1XB_1 = A_2XB_2 \Rightarrow A_2^{-1}A_1X = XB_2B_1^{-1} \Rightarrow AX = XB \quad (1)$$

where $A = A_2^{-1}A_1$ and $B = B_2B_1^{-1}$. In order to keep our notation consistent with conventional notations, the transformation matrices A_i , B_i , and X are renamed as B_GT_i (the robot base frame to the gripper frame), C_WT_i (the camera frame to the world's frame), and C_GT (the gripper frame to the camera frame), respectively. There are different methods in the literature to obtain a solution for C_GT (or X) [18][19]. Unfortunately, these methods, as will be shown experimentally, do not yield good accuracy in practice. To address this issue, we propose a new strategy in which we formulate the hand-eye calibration problem as a point set matching problem and obtain a solution for the problem using matrix operations. The rationale for formulating the hand-eye calibration problem as a point set matching problem is as follows. If we assume that the transformation matrix from the robot base frame to the camera frame, i.e., B_GT , is known, then the relationship among the transformation matrix from the robot base frame to the camera frame (B_GT), the transformation matrix from the robot base frame to the gripper frame (B_GT), and the transformation matrix from the gripper frame to the camera frame (C_GT) can be expressed as,

$${}^B_GT = {}^B_GT {}^C_GT \quad (2)$$

Subsequently, the unknown transformation matrix from the robotic gripper to the camera (${}^C_C T$) can be derived as,

$$X = {}^C_C T = {}^B_C T^{-1} {}^B_C T \quad (3)$$

where, ${}^B_C T$ is known from the robot kinematics. To estimate ${}^B_C T$, one can analyze the points from different frames. In this way, the problem of hand-eye calibration lends itself to the problem of rigid transformation estimation between two point sets. The problem of estimating the transform between two point sets is known as point set matching and will be briefly reviewed in section II.

The contributions of the current work are as follows:

- We formulate and solve the problem of hand-eye calibration for robot manipulators as a point set matching problem.
- We propose a gradient descent algorithm on the Special Euclidean group SE(3), GD-SE(3), for estimating the homogeneous transformation matrix between two rigidly related point sets.
- We show that the proposed point set matching formulation is more accurate than conventional hand-eye calibration formulation ($\mathbf{AX} = \mathbf{XB}$).
- The proposed approach takes advantage of known robot kinematics and is more convenient to perform hand-eye calibration for robot manipulators.

The rest of the paper is organized as follows: Section II briefly reviews related works in hand-eye calibration and point set matching. Section III introduces the proposed algorithm and provides corresponding mathematical derivations. Section IV experimentally evaluates the validity of the proposed formulation and its solution. Finally, Section V concludes the paper.

II. RELATED WORKS

A. $AX = XB$ Calibration

The hand-eye calibration problem is conventionally formulated as,

$$AX = XB \quad (4)$$

This well-known equation was first formulated in [20] and [21]. There are three main strategies to solve for X , known as separable, simultaneous, and iterative strategies [17].

In separable strategy, the rotation and the translation portion of the homogeneous transformation matrices are considered separately, i.e.,

$$AX = XB \Rightarrow \begin{bmatrix} R_A & P_A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_X & P_X \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_X & P_X \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_B & P_B \\ 0 & 1 \end{bmatrix} \quad (5)$$

where R is the rotation matrix and P is the translation vector. From (5), it is not difficult to show that,

$$R_A R_X = R_X R_B \quad (6)$$

$$R_A P_X + P_A = R_X P_B + P_X \quad (7)$$

Shiu and Ahmad [20] derived a general solution for this problem as,

$$R_X = Rot(k_A, \beta) Rot(v, \omega) \quad (8)$$

where $v = k_B \times k_A$ and $\omega = \text{atan2}(|k_B \times k_A|, k_B \cdot k_A)$ in which k_A and k_B denote (nonparallel) rotational axes of R_A and R_B , respectively, and β is any arbitrary angle that can be obtained by associating the general solutions of the two equations, $R_{A_1} R_X = R_X R_{B_1}$ and $R_{A_2} R_X = R_X R_{B_2}$. Having obtained R_X , one can find the translation component as,

$$(R_A - I)P_X = R_X P_B - P_A \quad (9)$$

Park and Martin [19] formed a closed-form exact solution for R_X by exploiting Lie theory. This solution can be expressed explicitly and visualized geometrically [19].

A common problem associated with separable strategy is that the error during the calculation of the rotational component is brought into the calculation of the translation part [14][17]. The simultaneous strategy overcome this problem by solving for the rotational and transnational components, well, simultaneously. To this end, Horaud and Dornaika [18] represented the rotational part using a unit quaternion (q) and applied a nonlinear technique to solve for the rotation and translation parts simultaneously. They were the first to apply a simultaneous nonlinear minimization with respect to the rotation quaternion and the translation vector [12]. Wu *et al.* formed an universal analytical solution using 4-dimensional Procrustes analysis [22].

The iterative strategy as the name suggests solves for the rotation and the translation parts iteratively using optimization techniques [14][17]. Unlike the separable strategy, the use of optimization technique avoids the error propagation from the rotational part into the transnational part. Zhuang and Shiu [23] obtained a solution by minimizing $\|AX - XB\|$ with the Levenberg-Marquardt algorithm. Tabb and Yousef [24] explored a collection of iterative methods consisting of different cost functions, different parameterizations of rotational components, and different formulations on both real and simulated data sets.

The use of gradient descent on the Special Euclidean group SE(3) for ultrasound sensors calibration was reported in [25]. In this approach, a gradient on SE(3) is defined using the standard basis of the Lie algebra. In this approach, a gradient on SE(3) using the standard basis of the Lie algebra E_n is defined as,

$$\nabla f(g) = \begin{bmatrix} \frac{d}{dt} f(g \circ \exp(tE_1))|_{t=0} \\ \frac{d}{dt} f(g \circ \exp(tE_2))|_{t=0} \\ \vdots \\ \frac{d}{dt} f(g \circ \exp(tE_6))|_{t=0} \end{bmatrix} \quad (10)$$

where $g \in \text{SE}(3)$. The update equation is given by,

$$g_{k+1} \approx g_k \exp(\Delta t V_g^r) \quad (11)$$

in that $V_g^r = g^{-1} \dot{g} = -\alpha \widehat{\nabla f(g)}$, and the cost function is defined as,

$$U_{AX=XB} = \|AX - XB\| \quad (12)$$

For more exhaustive review of $\mathbf{AX} = \mathbf{XB}$ problem, the readers are referred to [17].

B. Point Set Matching

Point set matching, also known as point set registration, is the problem of finding a spatial transformation that aligns two sets of points, known as target points and moving points with known correspondence between them. The transformation can be either rigid with no change of distance between corresponding points, or non-rigid which involves scaling and shear mapping [26]. Besl and McKay [27], and Zhang [28] proposed separately the Iterative Closest Point (ICP) algorithm. The algorithm assigns correspondence between the pair of points in two sets based on their distance. The algorithm then iteratively minimizes the mean squared error of the distance between all corresponding points until it converges. The use of ICP algorithm and its variations for point set matching has been widely reported in the literature [29][30][31].

Myronenko and Song [32] proposed a probabilistic approach termed Coherent Point Drift (CPD) to register two point sets for both rigid and non-rigid transformations. In this approach, the problem of point sets alignment is considered as a probability density estimation problem. The moving set is represented as a Gaussian mixture model (GMM) centroids, which are iteratively fitted to the target set by maximizing the likelihood. The GMM centroids are forced to move coherently so that the topological structure of the point sets could be preserved. There are also a number of reports on the use of CPD algorithm and its variations [33][34].

Due to space limitations, we leave an exhaustive review of all point set matching algorithms to [26]. In the next section, we present the details of our proposed point set matching algorithm.

III. THE GRADIENT DESCENT ALGORITHM GD-SE(3)

In this section, we formulated the hand-eye calibration problem as a point set matching and provide a solution to this problem using gradient descent algorithm on the Special Euclidean group SE(3). We refer to this algorithm as GD-SE(3).

A. Objective Formulation

Unlike conventional formulations of $\mathbf{AX} = \mathbf{XB}$ problem, the objective of GD-SE(3) algorithm is formulated as a point set matching within the context of hand-eye calibration problem. Let us denote two point sets in the camera frame and the robot base frame as $\{^C X_i\}$ and $\{^B X_i\}$ ($i = 1, 2, \dots, n$), respectively. We assume the correspondence between the points in $\{^C X_i\}$ and $\{^B X_i\}$ are available. The two point sets can be related as follows,

$$^B P_i = {}^B T {}^C P_i, \quad \forall i = 1, 2, \dots, n \quad (13)$$

where ${}^B T = \begin{bmatrix} {}^B R & {}^B P \\ 0 & 1 \end{bmatrix}$ is the homogeneous transformation matrix from the robot base frame to the camera frame, in

which ${}^B R$ and ${}^B P$ are the rotation matrix and translation vector of ${}^B T$, and ${}^B P_i = [{}^B X_i \ 1]^T$ and ${}^C P_i = [{}^C X_i \ 1]^T$. Let us also denote ${}^B \hat{T}$ as the estimated ${}^B T$. The objective of GD-SE(3) is to find a ${}^B \hat{T}$ that minimizes the following cost function

$$\begin{aligned} f({}^B T, {}^C P, {}^B P) &:= \frac{1}{n} \sum_{i=1}^n \|{}^B T {}^C P_i - {}^B P_i\|^2 \\ &= \frac{1}{n} \sum_{i=1}^n \|{}^B R {}^C X_i + {}^B P - {}^B X_i\|^2 \end{aligned} \quad (14)$$

Using (3) and the estimation of ${}^B \hat{T}$ obtained in (14), one can obtain ${}^G T$ as follows,

$${}^G T = {}^G T {}^B \hat{T} = {}^G T^{-1} {}^B \hat{T} \quad (15)$$

where ${}^B T$ represents the robot kinematics from the robot base to the gripper.

Thus, the problem of hand-eye calibration, i.e., estimating ${}^G T$ is now converted to the problem of matching the two points sets of $\{^C X_i\}$ and $\{^B X_i\}$ using the transformation ${}^B T$.

B. Algorithm Derivation

In this section we present the solution to the problem defined in (14) and discuss the details of GD-SE(3) algorithm. Let us consider the 3-dimensional *Special Euclidean group*, defined as

$$\begin{aligned} SE(3) &:= \left\{ g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \right. \\ &\quad \left. R^{-1} = R^T \in \mathbb{R}^{3 \times 3}, \det(R) = 1, p \in \mathbb{R}^3 \right\} \end{aligned} \quad (16)$$

On any Lie group the tangent space at the group identity has the structure of a *Lie algebra*. The Lie algebra of SE(3) is given by,

$$\begin{aligned} SE(3) &:= \left\{ U = \begin{bmatrix} \Omega & v \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \right. \\ &\quad \left. \Omega^T = -\Omega \in \mathbb{R}^{3 \times 3}, v \in \mathbb{R}^3 \right\} \end{aligned} \quad (17)$$

For all X, U_1 , and $U_2 \in SE(3)$ the right invariant Riemannian metric $\langle \cdot, \cdot \rangle_X$ is defined as, $\langle XU_1, XU_2 \rangle_X := \langle U_1, U_2 \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the Euclidean metric on $\mathbb{R}^{4 \times 4}$. Given a differentiable smooth function $f: SE(3) \rightarrow \mathbb{R}$, the gradient of f , i.e., $\text{grad}_X f \in T_X SE(3)$ relative to the Riemannian metric is uniquely defined as,

$$df \cdot XU = \langle \text{grad}_X f, XU \rangle_X = \langle X^{-1} \text{grad}_X f, U \rangle. \quad (18)$$

In view of the cost function defined in (14) and the definition of the gradient in (18), one can obtain $\text{grad}_{{}^B T} f({}^B T, {}^C P, {}^B P)$ as follows,

$$\begin{aligned} & df({}^B_C T, {}^C P, {}^B P) \cdot {}^B_C T \cdot U \\ &= \langle {}^B_C T^{-1} \text{grad}_{{}^B_C T} f({}^B_C T, {}^C P, {}^B P), U \rangle \end{aligned} \quad (19)$$

with some $U \in \text{SE}(3)$. On the other hand, in view of (14) we have,

$$\begin{aligned} & df({}^B_C T, {}^C P, {}^B P) \cdot {}^B_C T \cdot U \\ &= \frac{1}{n} \sum_{i=1}^n ({}^B_C T {}^C P_i - {}^B P_i)^\top {}^B_C T \cdot U \cdot {}^C P_i \\ &= \frac{1}{n} \sum_{i=1}^n \langle {}^B_C T^\top ({}^B_C T {}^C P_i - {}^B P_i) {}^C P_i^\top, U \rangle \end{aligned} \quad (20)$$

where we make use of the facts that $y^\top x = \text{tr}(xy^\top)$ and $\langle A^\top, B \rangle = \text{tr}(AB)$. Defining $\mathbb{P} : \mathbb{R}^{4 \times 4} \rightarrow \text{SE}(3)$ as a projection of \mathbb{A} on the Lie algebra $\text{SE}(3)$, then for all $A \in \mathbb{R}^{3 \times 3}, b, c^\top \in \mathbb{R}^3$ and $d \in \mathbb{R}$, we have

$$\mathbb{P} \left(\begin{bmatrix} A & b \\ c & d \end{bmatrix} \right) = \begin{bmatrix} \mathbb{P}_a(A) & b \\ 0_{1 \times 3} & 0 \end{bmatrix}. \quad (21)$$

where $\mathbb{P}_a(A) := (A - A^\top)/2$ denotes the anti-symmetric projection of A . It could be shown that for any $U \in \text{SE}(3)$ and $\mathbb{A} \in \mathbb{R}^{4 \times 4}$

$$\langle \langle \mathbb{A}, U \rangle \rangle = \langle \langle \mathbb{P}(\mathbb{A}), U \rangle \rangle = \langle \langle U, \mathbb{P}(\mathbb{A}) \rangle \rangle. \quad (22)$$

From (20) and (22), we obtain,

$$\begin{aligned} & df({}^B_C T, {}^C P, {}^B P) \cdot {}^B_C T \cdot U \\ &= \frac{1}{n} \sum_{i=1}^n \langle \mathbb{P} ({}^B_C T^\top ({}^B_C T {}^C P_i - {}^B P_i) {}^C P_i^\top), U \rangle \\ &= \frac{1}{n} \sum_{i=1}^n \langle \mathbb{P} (({}^C P_i - {}^B_C T^{-1} {}^B P_i) {}^C P_i^\top), U \rangle \end{aligned} \quad (23)$$

where we make use of the fact that $\mathbb{P}(X^\top y z^\top) = \mathbb{P}(X^{-1} y z^\top)$ for all $y, z \in \mathbb{R}^4$ and $X \in \text{SE}(3)$. Then, in view of (19) and (23), it can be concluded that,

$$\begin{aligned} & \text{grad}_{{}^B_C T} f({}^B_C T, {}^C P, {}^B P) \\ &= {}^B_C T \mathbb{P} \left(\frac{1}{n} \sum_{i=1}^n ({}^C P_i - {}^B_C T^{-1} {}^B P_i) {}^C P_i^\top \right) \end{aligned} \quad (24)$$

Moreover, motivated by [4] [5], the update equation for ${}^B_C T$ can be defined as,

$${}^B_C T_{k+1} = {}^B_C T_k \exp(-\alpha {}^B_C T_k^{-1} \text{grad}_{{}^B_C T} f({}^B_C T, {}^C P, {}^B P)) \quad (25)$$

where $\alpha > 0$ is the learning rate, and the initial value of ${}^B_C T$ is ${}^B_C T_0 \in \text{SE}(3)$.

C. The Gradient Descent Algorithm

Having derived the definition of gradient and the update equation, we now describe the proposed GD-SE(3) algorithm (see Algorithm 1).

It is proven that gradient descent methods can converge to the global minimum from any initial condition on the Special Euclidean group $\text{SE}(3)$ [4]. Hence, the initial value of ${}^B_C \hat{T}$ when minimizing the cost function (14) can be randomly chosen in $\text{SE}(3)$. To terminate the iteration properly, a simple convergence detection method can be employed. The mean difference in the rotation matrix and the translation vector from the recent three iterations (denoted as "dR" and "dp" respectively) are computed for convergence detection. If dR and dp are both less than a tolerance (denoted as " σ "), the iteration will be terminated.

Contrary to terminating condition, selection of the learning rate (i.e., α) is often more important and difficult when using gradient descent in an optimization problem. A large α prohibits the algorithm to converge, while a small α slows down the converge of the algorithm, even for off-line calculations. Hence, setting α as a constant value is not the best option. To select an adaptive α , we employ a method called ADADELTA [35]. This method is a per-dimension learning rate method for gradient descent that utilizes the gradient accumulation (denoted as $E[g^2]$) and the step size of the accumulation (denoted as $E[\Delta x^2]$) to dynamically calculate the learning rate as follows,

$$E[g^2] = \rho E[g^2] + (1 - \rho)g^2 \quad (26)$$

$$E[\Delta x^2] = \rho E[\Delta x^2] + (1 - \rho)\Delta x^2 \quad (27)$$

$$\alpha = \left(\frac{E[\Delta x^2] + \tau}{E[g^2] + \tau} \right)^{\frac{1}{2}} \quad (28)$$

where g is the gradient, $\Delta x = -\alpha g$ is the step size, ρ denotes a decay constant that is similar to the decay constant commonly used in momentum methods, and τ is a smoothing term to avoid the denominator from becoming zero. The details of the GD-SE(3) algorithm is given below,

Algorithm The GD-SE(3) algorithm

-
- Input:** two point sets with known correspondence ($\{{}^B P_i\}$ and $\{{}^C P_i\}$, $i = 1, 2, \dots, n$), maximum iteration (N).
Output: the best estimation of the homogeneous transformation matrix between the input point sets (${}^B_C \hat{T}_k$, $k = 1, \dots, N$)
Initialization: randomly generate a rotation matrix, $R_0 \in \text{SO}(3)$, and a translation vector, $p_0 \in \mathbb{R}^3$, to get the initial guess of ${}^B_C \hat{T}_0$, which is $(R_0, p_0) \in \text{SO}(3)$
- 2: For convergence detection: define the tolerance (σ).
For adaptive learning rate: initialize accumulation variables $E[g^2] = 0, E[\Delta x^2] = 0$; define the decay rate ρ and the value of τ .
 - 4: $k = 0$
 - repeat**
 - 6: Convergence detection: $\{$
 ${}^B_C R_k = {}^B_C T_k(1 : 3, 1 : 3)$
 - 8: ${}^B_C p_k = {}^B_C T_k(1 : 3, 4)$
 $q_k = \text{rot2qua}({}^B_C R_k)$ ▷ Convert the rotation matrix into quaternion
 - 10: **if** $k > 0$ **then**
-

```

12:   if  $k \geq 3$  then
13:      $dR = \frac{1}{3} \sum_{j=k-3}^{k-1} \text{acos}(\frac{q_j \cdot q_{j+1}}{\|q_j\| \|q_{j+1}\|})$ 
14:      $dp = \frac{1}{3} \sum_{j=k-3}^{k-1} \|{}^B_C P_j - {}^B_C P_{j+1}\|$ 
15:   else
16:      $dR = \frac{1}{k+1} \sum_{j=0}^k \text{acos}(\frac{q_j \cdot q_{j+1}}{\|q_j\| \|q_{j+1}\|})$ ,
17:      $dp = \frac{1}{k+1} \sum_{j=0}^k \|{}^B_C P_j - {}^B_C P_{j+1}\|$ 
18:   end if
19:   end if
20:   if  $dR < \sigma$  and  $dp < \sigma$  then
21:     Return  ${}^B_C T_k$ 
22:   end if}
23:   Calculate  $U = \frac{1}{n} \sum_{i=1}^n ({}^C P_i - {}^B_C \hat{T}_k^{-1} {}^B P_i) {}^C P_i^\top$ : {
24:    $U = 0$ 
25:   for all measurements  $i$  do
26:      $U = U + \frac{1}{n} ({}^C P_i - {}^B_C \hat{T}_k^{-1} {}^B P_i) {}^C P_i^\top$ 
27:   end for}
28:   Get the gradient:  $\text{grad}_{{}^B_C \hat{T}_k} f = {}^B_C \hat{T}_k \mathbb{P}(U)$ 
29:   Calculate the learning rate (ADADELTA): {
30:    $E[g^2] = \rho E[g^2] + (1 - \rho) \|\text{grad}_{{}^B_C \hat{T}_k} f\|^2$ 
31:    $\alpha = (\frac{E[\Delta x^2] + \tau}{E[g^2] + \tau})^{\frac{1}{2}}$ 
32:    $E[\Delta x^2] = \rho E[\Delta x^2] + (1 - \rho) \Delta x^2$ 
33:   } where  $\Delta x = -\alpha \|\text{grad}_{{}^B_C \hat{T}_k} f\|$  }
34:   Update:  ${}^B_C \hat{T}_{k+1} = {}^B_C \hat{T}_k \exp(-\alpha {}^B_C \hat{T}_k^{-1} \text{grad}_{{}^B_C \hat{T}_k} f)$ 
35:    $k = k + 1$ 
36: until  $k \equiv N$ 
37: return  ${}^B_C \hat{T}_k$ 

```

IV. EXPERIMENTAL RESULTS

In this section, the hardware setup used to evaluate the algorithm and the experimental results that demonstrate the performance of hand-eye calibration are discussed. The experiments are designed to verify the performance of the GD-SE(3) while comparing the results with those obtained from conventional hand-eye calibration algorithms as well as point set matching methods. The hardware setup used in these experiments is described in the next section.

A. Hardware Setup

A KUKA Light-Weight Robot (LWR) IV and a CRS Robotics underactuated gripper [36] were used for conducting our experiments. A screw was fixed to the gripper using a 3D-printed adapter. The sharp tip of the screw was placed in the middle of the gripper or the point of contacts of the gripper's two fingertips (see Fig. 2(e)). The gripper frame is defined to be at the tip of the screw. We used two different calibration devices in conducting the experiments that are shown in Fig. 2(a) and Fig. 2(d). One calibration device consists of a 3D-printed bed with 25 screws of different heights (see Fig. 2(c)). The other calibration device is the typical checkerboard pattern that contains 54 corners (see Fig. 2(d)). We also used two different stereo cameras (KYT-U100-960R301 and Intel RealSense D435) to conduct concurrent experiments and compare the sensitivity of results to the camera quality. The two calibration devices and the two cameras were used in three different combinations: the 3D-printed calibration device with KYT camera (Setup 1), the checkerboard pattern with KYT camera (Setup 2), and the checkerboard pattern with Intel camera (Setup 3). These

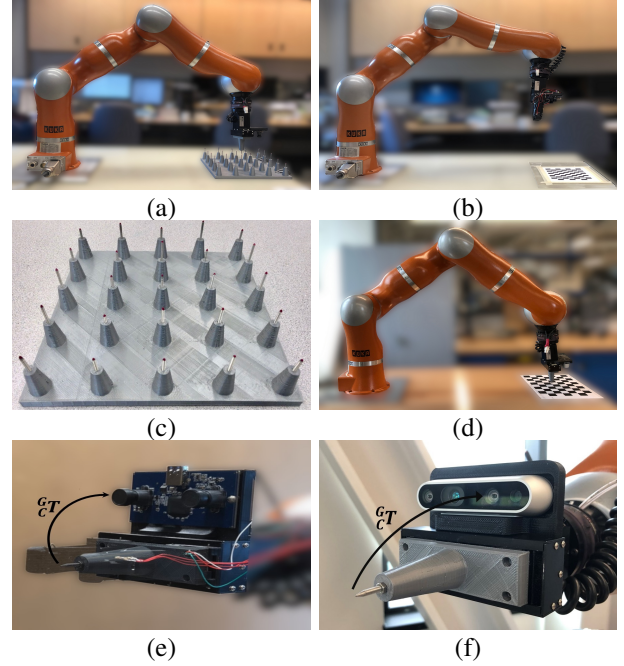


Fig. 2: Experimental setup. (a) The 3D-printed setup for performing calibration using contact. (b) The checkerboard setup for non-contact calibration. (c) The 3D-printed calibration device consisting of 25 screws. (d) The black and white checkerboard with 54 corners. (e) The gripper with the KYT stereo camera. (f) The gripper with the Intel RealSense D435 stereo camera.

setups result in different levels of noises in the experimental data allowing to evaluate and compare the performance of all various algorithms. All computations and visualizations were performed on a DELL personal computer with i5 CPU along with MATLAB r2018b software.

B. Experiment Data Collection

In this section, the details of each experiment are described.

1) Conventional Formulation

Our first set of experiments aimed at evaluating the performance of conventional methods of hand-eye calibration, i.e., $\mathbf{AX} = \mathbf{XB}$. To this end, multiple images of the checkerboard calibration pattern were captured. The corresponding poses of the gripper (i.e., position + orientation) were also recorded for each captured image using the robot encoders. As described in (1), the transformation matrix between the robot base frame to the gripper frame (i.e., ${}^B_C T$) for two different poses were used to calculate the \mathbf{A} matrix, i.e., $\mathbf{A} = \mathbf{A}_2^{-1} \mathbf{A}_1 = {}^B_C T_2^{-1} {}^B_C T_1$. Moreover, the transformation matrix between the camera frame and the world's frame (i.e., ${}^C_W T$) for these poses were used to calculate the \mathbf{B} matrix, i.e., $\mathbf{B} = \mathbf{B}_2 \mathbf{B}_1^{-1} = {}^C_W T_2 {}^C_W T_1^{-1}$. As an alternative approach the matrix \mathbf{B} can be also estimated by matching the 3D coordinates of the checkerboard corners associated with the two poses. Two different implementations of traditional formulation reported in [18][19] were evaluated. We captured

30 images using Setup 1 and 100 images using Setup 2 and 3 for these purposes.

2) Proposed Formulation: Point Set Matching

Next, we evaluated the performance of point set matching algorithms. To this end, two types of target points were selected. For the 3D-printed calibration device, the tips of the screws were selected as target points while for the checkerboard, the target points were the corners of the checkerboard patterns. The actual locations of the target points were obtained by guiding the robot's gripper (the tip of the screw attached to the gripper) to make a physical contact with each target point as shown in Fig. 2(a) and (d). Referring to (13), the data obtained in this part constituted $\{\{^B P\}\}$. Several stereo images of the target points were also captured from multiple poses and the location of these points in the camera frame were estimated using triangulation. In the case of 3D-printed calibration device the target points in each stereo images were selected manually for triangulation while in the checkerboard case, these points were detected automatically with image processing. The data obtained in this part constituted $\{\{^C P\}\}$. The manual selection of target points in 3D-printed calibration device introduces more error in $\{\{^C P\}\}$ values. Yet, this error for automatic detection is limited to 1 millimeter.

We compared the performance of ICP, CPD, and GD-SE(3) algorithms using this data. To this effect, 10 sets of data using Setup 1 and 100 sets of data using Setup 2 and 3 were obtained for $\{^C P\}$. Of this data, 70% were used for training while 30% were used for validation.

C. Performance Evaluation

The data acquired previously was used to train and assess the performance of various algorithms. The data was partly used for training and partly for assessment.

1) Training Phase

During the training phase, two most common hand-eye calibration algorithms [18][19] were selected for estimating $^C T$ (i.e., the X matrix in $AX = XB$) using the A and B matrices obtained previously.

Similarly, we used the data $\{\{^B P\}\}$ and $\{\{^C P\}\}$ to estimate $^B T$ (in (13)) using ICP [27], CPD [32]), and GD-SE(3) algorithms. We then used the estimated $^B \hat{T}$ following the approach (15) to obtain $^C \hat{T}$.

Since multiple data sets were used in estimating $^C T$, to reduce the effect of measurement noises on the result, the average value of $^C T$ was regarded as the training result.

In order to evaluate the performance of each algorithm, the reconstruction accuracy error (RAE) was used. RAE is defined as the average root mean square error (RMSE) of the Euclidean distance between the estimated points in the robot base frame (i.e., $\{\{^B \hat{P}\}\}$) and the truth points (i.e., $\{\{^B P\}\}$) as,

$$RAE = \frac{1}{n} \left(\sum_{i=1}^n \|{}^B \hat{P}_i - {}^B P_i\|^2 \right)^{\frac{1}{2}} \quad (29)$$

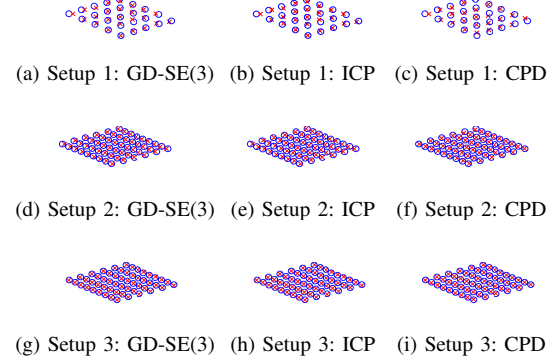


Fig. 3: Examples of the training results: estimated $\{^B P\}$ (red cross) vs measured $\{^B P\}$ (blue circle)

The training time for each algorithm was also measured so as to further compare the performance of the algorithms.

It should be pointed out that it is necessary to provide the correspondence between $\{^C P\}$ and $\{^B P\}$ for the GD-SE(3) and ICP algorithms. The CPD algorithm, on the other hand, estimates the correspondence automatically. Comparing the results from GD-SE(3), ICP and CPD algorithms later in this section, will show the importance of the correspondence in the performance of each algorithm.

2) Validation Phase

The performance of implemented algorithms were evaluated and compared using the remaining data not used for the training. This evaluation was based on RAE defined in (29) as well as root mean combined rotation and translation error (RMCE) defined as,

$$RMCE = \frac{1}{n} \left(\sum_{i=1}^n \|A_i X - X B_i\|^2 \right)^{\frac{1}{2}} \quad (30)$$

where A_i and B_i are the robot and camera motion matrices described in (1).

D. Results and Discussion

In this part, the results from all experiments are provided and discussed.

1) Results of training phase

The training phase was conducted following the procedure described in Section IV-C1. Examples of training results obtained using GD-SE(3), ICP, and CPD algorithms are shown in Fig. 3, and the values of training errors for each algorithm are listed in Table I, II, and III. As shown in Fig. 3, the estimated values $\{^B P\}$ obtained using GD-SE(3), ICP and CPD algorithms seem to match well with the actual value of $\{^B P\}$ (ground truth) for all three setups. The average RAE values for each algorithm during the training phase are also listed in the tables. The RAE for the CPD algorithm is much bigger than the RAE of GD-SE(3) and ICP algorithm, since the former estimates the correspondence between $\{^C P\}$ and $\{^B P\}$ automatically. The problem was exacerbated since

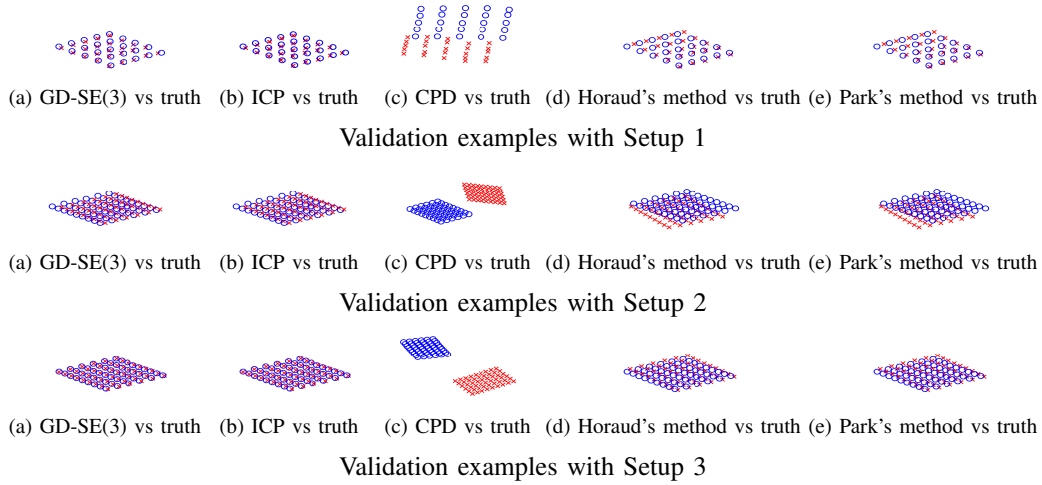


Fig. 4: Validation examples with all three setups: estimated $\{^B P\}$ (red cross) vs measured $\{^B P\}$ (blue circle)

TABLE I: Experimental Results With Setup 1

Methods		Average RAE During Training (mm)	Training Time (seconds)	Average RAE During Validation (mm)	RMCE
GD-SE(3)		6.2419	5.213	12.7078	1.9×10^{-3}
Traditional Algorithms	Horaud's method [18]	—	0.019	34.3691	1.9×10^{-3}
	Park's method [19]	—	0.004	34.3690	1.9×10^{-3}
Point Set Matching Algorithms	Iterative Closest Point (ICP) [27]	6.2419	0.057	12.7078	1.9×10^{-3}
	Coherent Point Drift (CPD) [32]	152.8428	0.059	191.3983	7.7×10^{-3}

TABLE II: Experimental Results With Setup 2

Methods		Average RAE During Training (mm)	Training Time (seconds)	Average RAE During Validation (mm)	RMCE
GD-SE(3)		1.7001	229.6	20.4808	2.4×10^{-3}
Traditional Algorithms	Horaud's method [18]	—	0.011	35.4475	2.4×10^{-3}
	Park's method [19]	—	0.009	35.4484	2.4×10^{-3}
Point Set Matching Algorithms	Iterative Closest Point (ICP) [27]	1.7001	0.447	20.4837	2.4×10^{-3}
	Coherent Point Drift (CPD) [32]	60.3571	2.250	267.5175	7.8×10^{-3}

TABLE III: Experimental Results With Setup 3

Methods		Average RAE During Training (mm)	Training Time (seconds)	Average RAE During Validation (mm)	RMCE
GD-SE(3)		0.9422	169.7	4.9198	1.2×10^{-3}
Traditional Algorithms	Horaud's method [18]	—	0.011	30.3736	1.1×10^{-3}
	Park's method [19]	—	0.009	30.3738	1.1×10^{-3}
Point Set Matching Algorithms	Iterative Closest Point (ICP) [27]	0.9422	0.448	4.9193	1.1×10^{-3}
	Coherent Point Drift (CPD) [32]	105.4434	1.904	256.8278	6.8×10^{-3}

$\{^C P\}$ and $\{^B P\}$ used as target points were arranged in a rectangular pattern, making them more likely to be matched incorrectly.

One of the shortcoming of the proposed GD-SE(3) algorithm is the computational time of the algorithm. However, in practice, hand-eye calibrations are often performed off line, making the computational time of the algorithm of less importance. Moreover, the computational speed of the GD-SE(3) algorithm is dependant and hence can be improved significantly with better selection of the starting point and the learning rate (α).

Remark 1: The initial transformation matrix in the GD-SE(3) algorithm is randomly selected from the special Euclidean group SE(3) considering no prior information. The computational time of the GD-SE(3) algorithm becomes comparable

with that of the ICP algorithm, should one uses identical initial transformation matrix and identical terminating conditions in both algorithms.

To validate Remark 1, more experiments were conducted for GD-SE(3) and ICP. A better initial transformation matrix between $\{^C P\}$ and $\{^B P\}$ was estimated through singular value decomposition (SVD) instead of random selection. In these experiments, we chose an identical method for detecting convergence (termination condition) in GD-SE(3) and ICP (see Algorithm 1). The iteration was terminated when both dR and dT were less than 1×10^{-7} . The results from these experiments are shown in Table IV, V and VI. As observed, the GD-SE(3) algorithm in all three setups outperforms the ICP algorithm in terms of the computational time if the initial transformation matrix is selected using SVD. Relaxing this

TABLE IV: Experimental Results With Setup 1 And SVD Initialization

Methods	Average RAE During Training (mm)	Training Time (seconds)	Average RAE During Validation (mm)	RMCE
GD-SE(3)	6.2419	0.0089	12.7078	1.9×10^{-3}
ICP	6.2419	0.0408	12.7078	1.9×10^{-3}

TABLE V: Experimental Results With Setup 2 And SVD Initialization

Methods	Average RAE During Training (mm)	Training Time (seconds)	Average RAE During Validation (mm)	RMCE
GD-SE(3)	1.7001	0.0534	20.4837	2.4×10^{-3}
ICP	1.7001	0.3791	20.4837	2.4×10^{-3}

TABLE VI: Experimental Results With Setup 3 And SVD Initialization

Methods	Average RAE During Training (mm)	Training Time (seconds)	Average RAE During Validation (mm)	RMCE
GD-SE(3)	0.9422	0.0774	4.9193	1.1×10^{-3}
ICP	0.9422	0.3793	4.9193	1.1×10^{-3}

condition can significantly simplify the GD-SE(3) algorithm at the cost of increasing its computational time.

2) Result of validation phase

The validation phase was conducted after the training phase following the procedure described in Section IV-C2. Examples of validation results obtained from different implemented algorithms are shown in Fig. 4 and results for all three setups are listed in Table I, II, and III. As seen, the results are consistent with those obtained during training. Once again, the adverse effect of not providing correspondence between points in the performance of the CPD algorithm is clear. Another importance observation in the obtained results is the inconsistency of the values of RMCE with those from RAE. As observed, the RMCE cannot precisely describe the accuracy of the hand-eye calibration results where such differences are reflected clearly in RAE. By comparing these results, it can be concluded that those algorithms based on point set matching with given correspondence (e.g., GD-SE(3) and ICP) can outperform conventional approaches based on $AX = XB$ formulation. The reason for this can be explained as such. Conventional algorithms often define a cost function similar to $\frac{1}{n} \sum_{i=1}^n \|A_i X - X B_i\|^2$ and solve for the X matrix (i.e., $\mathcal{C}T$) by minimizing the cost function. In reality, such optimization as also ascertained with the result in Table I, II, and III does not necessary minimize the RAE values.

To evaluate the effects of position and orientation changes on the positioning accuracy of the camera, the estimated value of $\mathcal{C}T$ using Setup 3 was tested experimentally. A target point was captured by the Intel camera from 29 different poses (see Fig.5). The mean error (ME) and the root mean square error (RMSE) of the positioning error are shown in Table VII. The CPD algorithm was excluded from these experiments due to its poor performance. The results, once again demonstrate the advantage of our proposed formulation.

V. CONCLUSIONS

In this paper, the hand-eye calibration problem was formulated and solved as a point set matching problem. A new gradient descent algorithm on Special Euclidean group SE(3)

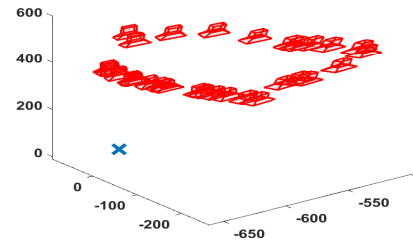


Fig. 5: The locations of the camera and the target point during the experimental test

TABLE VII: Experimental results using the robot manipulator

Methods	ME (mm)	RMSE (mm)
GD-SE(3)	5.0344	6.8989
ICP	5.0344	6.8989
Horaud's	30.5941	30.8792
Park's	30.5943	30.8794

(termed as GD-SE(3)) was proposed to solve the hand-eye calibration problem. To validate the new formulation based on point set matching, the performance of the proposed GD-SE(3) algorithm was thoroughly examined and compared with conventional hand-eye calibration algorithms as well as other widely used point set matching algorithms. The results clearly showed that formulating the hand-eye calibration problem as a point set matching problem can achieve better accuracy in practice. The proposed algorithm provide and alternative and convenient approach to perform hand-eye calibration in robotic manipulator. The GD-SE(3) algorithm provides a consistent and repeatable process for performing hand-eye calibration as it relies on the motion of the robot arm as opposed to the motion of the captured image. The results also showed a drawback of GD-SE(3) in terms of longer computational time. However, this shortcoming is of less importance in off-line calibration procedures.

REFERENCES

- [1] Y. Guo, M. Bennamoun, F. Soheli, M. Lu, and J. Wan, "An integrated framework for 3-d modeling, object detection, and pose estimation from point-clouds," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 683–693, 2014.
- [2] Y. Yin, X. Wang, D. Xu, F. Liu, Y. Wang, and W. Wu, "Robust visual detection-learning-tracking framework for autonomous aerial refueling of uavs," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 510–521, 2016.
- [3] H. Bazargani and R. Laganière, "Camera calibration and pose estimation from planes," *IEEE Instrumentation & Measurement Magazine*, vol. 18, no. 6, pp. 20–27, 2015.
- [4] M.-D. Hua, M. Zamani, J. Trunpf, R. Mahony, and T. Hamel, "Observer design on the Special Euclidean group SE (3)," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 8169–8175.
- [5] M. Wang and A. Tayebi, "Hybrid pose and velocity-bias estimation on se(3) using inertial and landmark measurements," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3399 – 3406, 2019.
- [6] J. Jung, S.-M. Lee, and H. Myung, "Indoor mobile robot localization and mapping based on ambient magnetic fields and aiding radio sources," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp. 1922–1934, 2014.
- [7] X. Wang, C. Zhang, F. Liu, Y. Dong, and X. Xu, "Exponentially weighted particle filter for simultaneous localization and mapping based on magnetic field measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 7, pp. 1658–1667, 2017.
- [8] M. Wang and A. Tayebi, "Geometric nonlinear observer design for slam on a matrix lie group," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1488–1493.
- [9] J. Liu, Y. Li, and S. Chen, "Robust camera calibration by optimal localization of spatial control points," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 12, pp. 3076–3087, 2014.
- [10] C. Heinze, S. Spyropoulos, S. Hussmann, and C. Perwass, "Automated robust metric calibration algorithm for multifocus plenoptic cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 5, pp. 1197–1205, 2016.
- [11] Z. Hu, Y. Li, N. Li, and B. Zhao, "Extrinsic calibration of 2-d laser rangefinder and camera from single shot based on minimal solution," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 915–929, 2016.
- [12] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.
- [13] S. Yin, Y. Ren, Y. Guo, J. Zhu, S. Yang, and S. Ye, "Development and calibration of an integrated 3d scanning system for high-accuracy large-scale metrology," *Measurement*, vol. 54, pp. 65–76, 2014.
- [14] Z.-Q. Zhang, "Cameras and inertial/magnetic sensor units alignment calibration," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 6, pp. 1495–1502, 2016.
- [15] Y. Tian, W. R. Hamel, and J. Tan, "Accurate human navigation using wearable monocular visual and inertial sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 1, pp. 203–213, 2013.
- [16] X. Chen, C. Wang, W. Zhang, K. Lan, and Q. Huang, "An integrated two-pose calibration method for estimating head-eye parameters of a robotic bionic eye," *IEEE Transactions on Instrumentation and Measurement*, 2019.
- [17] M. Shah, R. D. Eastman, and T. Hong, "An overview of robot-sensor calibration methods for evaluation of perception systems," in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. ACM, 2012, pp. 15–20.
- [18] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] J. Wu, Y. Sun, M. Wang, and M. Liu, "Hand-eye calibration: 4d procrustes analysis approach," *IEEE Transactions on Instrumentation and Measurement*, 2019.
- [23] H. Zhuang and Y. C. Shiu, "A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 4, pp. 1168–1175, 1993.
- [24] A. Tabb and K. M. A. Yousef, "Solving the robot-world hand-eye (s) calibration problem with iterative methods," *Machine Vision and Applications*, vol. 28, no. 5-6, pp. 569–590, 2017.
- [25] M. K. Ackerman, A. Cheng, E. Bector, and G. Chirikjian, "Online ultrasound sensor calibration using gradient descent on the euclidean group," in *Proceeding of IEEE international conference on Robotics and automation (ICRA)*. IEEE, 2014, pp. 4900–4905.
- [26] B. Maiseli, Y. Gu, and H. Gao, "Recent developments and trends in point set registration methods," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 95–106, 2017.
- [27] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–607.
- [28] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International journal of computer vision*, vol. 13, no. 2, pp. 119–152, 1994.
- [29] T. Tykkälä, C. Audras, and A. I. Comport, "Direct iterative closest point for real-time visual odometry," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. Citeseer, 2011, pp. 2050–2056.
- [30] D. Tihonkih, A. Makovetskii, and V. Kuznetsov, "A modified iterative closest point algorithm for shape registration," in *Applications of Digital Image Processing XXXIX*, vol. 9971. International Society for Optics and Photonics, 2016, p. 99712D.
- [31] Y. He, B. Liang, J. Yang, S. Li, and J. He, "An iterative closest points algorithm for registration of 3d laser scanner point clouds with geometric features," *Sensors*, vol. 17, no. 8, p. 1862, 2017.
- [32] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [33] V. Golyanik, B. Taetz, G. Reis, and D. Stricker, "Extended coherent point drift algorithm with correspondence priors and optimal subsampling," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.
- [34] M. Lu, J. Zhao, Y. Guo, and Y. Ma, "Accelerated coherent point drift for automatic three-dimensional point cloud registration," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 2, pp. 162–166, 2015.
- [35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [36] M. Abdeetelal and M. R. Kermani, "Development and grasp analysis of a sensorized underactuated finger," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 6331–6336.