A noise-tolerant algorithm for robot-sensor calibration using a planar disk of arbitrary 3D orientation

Wenyu Chen, Jia Du, Wei Xiong, Yue Wang, Shueching Chia, Bingbing Liu, Jierong Cheng, Ying Gu

***Abstract*—In a 3D scanning task, a robot-sensor system controls a robotic arm to move a laser sensor. In order to align the coordinate system of the robotic arm and laser sensor, prior calibration is required to derive the transformation between both coordinate systems. This paper proposes a new calibration method in three steps: manual data collection, sensing data calculation, and transformation solution. Firstly, at least four data are required to be collected by the user. The sensing data is then calculated from the collected data and adopted to provide the desired transformation. The proposed algorithm has two features: arbitrary placement of planar disk and noise tolerant. Using a planar disk, the algorithm will automatically derive the angular relationship between the disk and the sensor plane, enabling arbitrary orientation placement. Noise tolerant is guaranteed by fitting ellipses during the sensing data calculation and using a single set of sensing data in transformation solution. Experiments and comparisons are given to demonstrate the efficiency of the proposed calibration algorithm.**

Note to Practitioners:

***Abstract*—This article was motivated by the problem of calibrating a laser sensor and a positioning device (robot arm, CMM, etc.) in a robust and fast manner. Specifically, the calibration is to derive the transformation by aligning the sensors coordinate system to the positioning devices coordinate system. The proposed calibration procedure consists of two parts: manual data collection and automatic transformation calculation. During manual data collection, users only need to select four different data; whereby each data contains of two positions with the same orientation. Then, the desired transformation will be derived automatically. The calibration is designed in an efficient and robust way whereby: 1) data collection is done using a simple planar disk placed in arbitrary orientations; 2) minimum human interaction required; 3) tolerant to noise in the sensor data; and 4) easy implementation by following a proven and standard protocol.**

***Keywords*—Robot-sensor calibration, noise-tolerant, calibration disk, elliptic fitting.**

I. Introduction

CALIBRATION is to determine the relationship among different measuring devices [1]–[3]. In any tool/flange or hand/eye integrated system involving moving sensor [4]–[6], finding the geometry transformation between two coordinate systems is crucial for achieving accurate three-dimensional geometry. Such calibration is widely used in reverse engineering [7], robot-assisted medical applications [8], visually guided robot grinding [9], and robotic visual inspection [10]. Calibration methods vary in different aspects, such as calibration targets, data acquisition, and ways to derive the transformation.

All calibrations require some form of calibration targets such as 2D boards or 3D sphere-shaped balls. Usually, different types of sensors use different calibration targets. For example, cameras uses 2D calibration boards [5] while the profile laser sensors use 3D calibration balls [11]–[15]. In comparison, a 2D disk is cheaper and convenient, but has not been used as calibration target for laser sensors. The challenge is to account for the orientation of the 2D disk in the base coordinate system during the calibration, and estimation of the orientation has not been discussed before. The fixed-point calibration algorithm was introduced in [16] where the calibration target is a fixed cross-point printed on a 2D disk rather than the entire disk.

A profile laser sensor projects a laser line on the object and provides a 2D profile in the sensor plane. Different calibration methods have been developed for profile laser sensors, which require users to manually move the laser line to certain positions to acquire the sensing data. For example, the fixed-point calibration algorithm [16] required the users to manually move the laser line until it passes through the specific fixed point. On the other hand, the ball-based calibration algorithms [11]–[14] perform better as users can project a laser line on any part of the ball. Normally, one set of sensing data is required for the desired transformation [15]. But in two-step calibration methods [11]–[14], users need to acquire two different sets of sensing data: one set uses robotic arm with a fixed orientation while the other set uses robotic arm with different orientations. The two sets of data are used to solve the rotation and translation aspects of the transformation, respectively.

Once the sensing data for calibration are obtained, the desired transformation is solved in different manners. Traditional calibration methods are by means of homogeneous transformation. The well-known hand/eye calibration was formulated by Shiu and Ahmad [4] and Tsai and Lenz [5] by solving a homogeneous transform equation using nonlinear minimization [17], dual quaternions [18], etc. Another solution is the two-step calibration algorithm which derives the rotation and translation portions separately [11]–[14]. Two limitations arise from the two-step algorithms. Firstly, noise in the sensing data and errors caused in the first step will be propagated to the second step [19]. Secondly, when more sensing data is required, the amount of intensive manual labour work is also increased. Minimal manual work in data collection for the

W. Chen, J. Du, W. Xiong, Y. Wang, S. Chia, B. Liu, J. Cheng, and Y. Gu are with Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore.(e-mail:chenw@i2r.a-star.edu.sg)

calibration is desired for better accuracy and faster processing.

For the calibration of a profile laser sensor mounted onto a robotic arm, this paper targets to develop a new noise-tolerant calibration algorithm with a simple calibration target, easy data acquisition procedure, and lesser sensing data compared to existing techniques. A planar disk is chosen as the calibration target. To acquire the sensing data for the calibration, users can project the laser line onto the 2D disk without the need to pass through any specific points, including the disk center. The proposed algorithm will be based on a single set of sensing data. Not only the amount of manual effort in data collection is reduced, but the amount of error propagation is also reduced. Most importantly, the calibration algorithm is robust to noise in laser sensing data and mechanical vibrations during arm movement.

This paper is structured as follows. Section I introduces the research background, and Section II gives the preliminaries. After that, the proposed calibration method is presented in Section III. Section IV proves the formulae used during calibration. Experimental results and comparisons are given in Section V. Finally, Section VI concludes this paper.

II. System and Notations

A. Robot-sensor system

A robot-sensor system uses a robotic arm connected to a profile sensor to perform a 3D scanning on target objects. The coordinate systems are shown in Fig.1. The robotic arm has a fixed base coordinate system Φ. At the i-th position and orientation, the Tool Center Point (TCP, tool0) coordinate system and the sensor coordinate system are denoted as Ψ_i and Γ_i, respectively. Notations are as follows

- $R_{tcp,i}$ is the rotation matrix from Ψ_i to Φ,
- $T_{tcp,i}$ is the translation vector from Ψ_i to Φ,
- R_{sensor} is the rotation matrix from Γ_i to Ψ_i, and
- T_{sensor} is the translation vector from Γ_i to Ψ_i.

Since the profile sensor is mounted on TCP, for TCP at different positions with different orientations, R_{sensor} and T_{sensor} remain unchanged. The calibration aims to derive R_{sensor} and T_{sensor} based on the data from the robotic arm and the sensor. At the i-th position and orientation, the profile sensor generates a planar profile in the sensor plane π_i. Its coordinate system $\Gamma_i = o_i x_i y_i z_i$ is defined such that the origin o_i locates in the plane π_i and y_i is the normal of π_i. Following such notations, the second coordinate of the 2D profile from the sensor is fixed as zero, i.e. $y = 0$ for the profile data.

B. Calibration target

A calibration target is an object where laser lights will be shone during data collection. Usually, 3D balls are preferred, because the sensor has a uniform view from any position and orientation [15]. A ball with known radius has been used as the calibration target for robot-laser sensor calibration [11]–[14]. Different from previous methods, we plan to calibrate the system using a 2D disk other than a 3D object. Fig.2 presents two forms of planar disks with known radius.

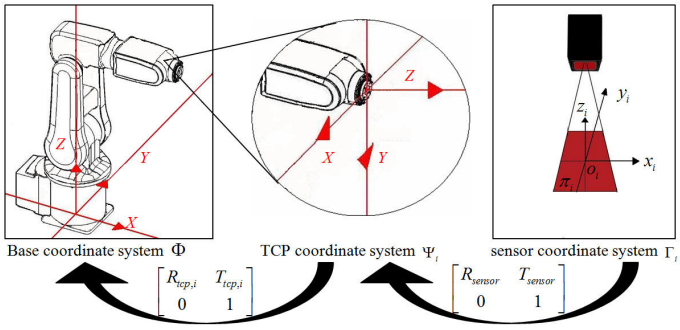
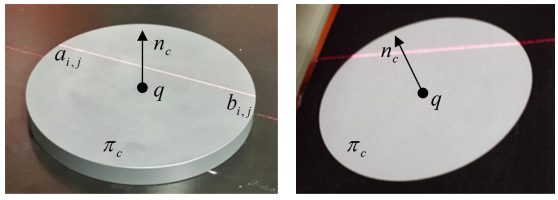


Fig. 1. Three different coordinate systems.

Profile data from the laser-scanning sensor can be divided into two types: chord data and non-chord data. On each profile, a chord is the line segment inside the disk, and the remnant data are non-chord data. Data preprocessing can remove the non-chord data in the profile. In Fig.2(a), non-chord data and chord data are on two different lines. Thus, non-chord data can be easily removed based on the profile's geometry. In Fig.2(b), non-chord data in black regions can be removed based on the profile's image. In this paper, experiments will be conducted using the 2D disk shown in Fig.2(a) and chords are derived using the split-and-merge algorithm [20]. Take the following notation:

- π_c is the plane where the disk locates,
- q is the center of the disk,
- r_c is the radius of the disk,
- n_c is the normal of π_c, and
- $c_{i,j} = [a_{i,j}, b_{i,j}]$ is a chord with two tips $a_{i,j}$ and $b_{i,j}$.

The new calibration method is dependent of q and n_c. Namely, the position and the orientation of the planar disk do not affect the calibration. Therefore, the planar disk can be placed at any position with any orientation for the calibration.



(a) (b)

Fig. 2. Planar disks with $100mm$ diameter: (a) a planar disk on the top of a cylinder, and (b) a planar white disk on a black paper.

C. Notations

For TCP at a position $T_{tcp,i}$ with orientation $R_{tcp,i}$, the sensor plane is π_i with the corresponding sensor coordinate system Γ_i. For disk-based calibration, n sensing data are required as

$$\left\{T_{tcp,i}, R_{tcp,i}, q^i, D_i^i, D_i^b\right\}, i = 1, 2, \cdots, n,$$

where

- $R_{tcp,i}$ and $T_{tcp,i}$ are TCP orientation and position for π_i passing through the disk center q,
- $q^i = (q^i_x, 0, q^i_z)$ gives the coordinates of q in Γ_i, and
- D_i is a TCP offset with coordinates D^b_i in Φ, and coordinates D^i_i in Γ_i.

Users can manually control the robotic arm and visually check the laser until it passes through the disk center q. However, due to human error and noise from the scanning system, it is difficult to achieve high accuracy. Thus, in our system, users only need to select a TCP orientation $R_{tcp,i}$ and two TCP positions. The system will automatically calculate the desired $T_{tcp,i}$.

For any physical point p, p^{oxyz} marks its coordinates in the coordinate system $oxyz$. Following this notation, $p^b = p^{base}$ denotes the coordinates of p in the base coordinate system Φ. A TCP position $T_{tcp,i}$ is always in the base coordinate system, i.e. $T_{tcp,i} = T^b_{tcp,i}$. For the i-th sensing data, when the sensor plane passes through the disk center, the sensor coordinate system is $\Gamma_i = o_i x_i y_i z_i$. Then, $p^{o_i x_i y_i z_i}$ becomes the coordinates of p in Γ_i. During the calibration, Γ_i will be rotated to the rotated coordinate system $\Gamma_{i,r}$.

The point pair p^{base} and $p^{o_i x_i y_i z_i}$ refers to the same physical point. Each pair is named as a conjugate pair. Multiple conjugate pairs are required to enable the calibration [15]. Referring to Fig.1, for the i-th sensing data, a conjugate pair q^b and q^i satisfies

$$\begin{bmatrix} q^b \\ 1 \end{bmatrix} = \begin{bmatrix} R_{tcp,i} & T_{tcp,i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{sensor} & T_{sensor} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q^i \\ 1 \end{bmatrix},$$

i.e.

$$q^b = R_{tcp,i} R_{sensor} q^i + R_{tcp,i} T_{sensor} + T_{tcp,i}. \quad (1)$$

In this equation, the coordinate q^b is unknown before the calibration, as the disk is freely placed in the scanning system. The rotation matrix $R_{tcp,i}$ can be read from the robotic arm. The translation $T_{tcp,i}$ and the coordinate q^i will be calculated according to the chords. The aim of the calibration is to obtain the rotation matrix R_{sensor} and the translation T_{sensor}.

III. Noise-tolerant calibration algorithm

Fig.3 is the flowchart for the calibration process, where

- n is the number of sensing data to be collected,
- m is the number of parallel chords to be scanned for the i-th sensing data,
- $\bar{T}_{tcp,i}$ and $\tilde{T}_{tcp,i}$ are two TCP positions manually selected for the i-th sensing data,
- s_i is of value 1 or -1,
- $c_{i,j}$ are m parallel chords scanned when TCP moves from $\bar{T}_{tcp,i}$ to $\tilde{T}_{tcp,i}$,
- π_i is the sensor plane for TCP at the i-th position $T_{tcp,i}$ with orientation $R_{tcp,i}$, and
- α_i is an angle between the sensor plane π_i and the disk plane π_c.

According to the flowchart, the calibration algorithm contains three parts: manual data collection, sensing data calculation, and transformation solution. For the i-th sensing data, users only need to manually move TCP to two different positions $\bar{T}_{tcp,i}$ and $\tilde{T}_{tcp,i}$ with the same orientation $R_{tcp,i}$. During sensing data calculation, m chords are obtained and the i-th sensing data $\{T_{tcp,i}, R_{tcp,i}, q^i, D^i_i, D^b_i\}$ is calculated based on these chords. Finally, all sensing data are combined together to derive R_{sensor} and T_{sensor}. This section will detail the key formulae and steps in the flowchart. Section IV will show how to deduct these formulae.

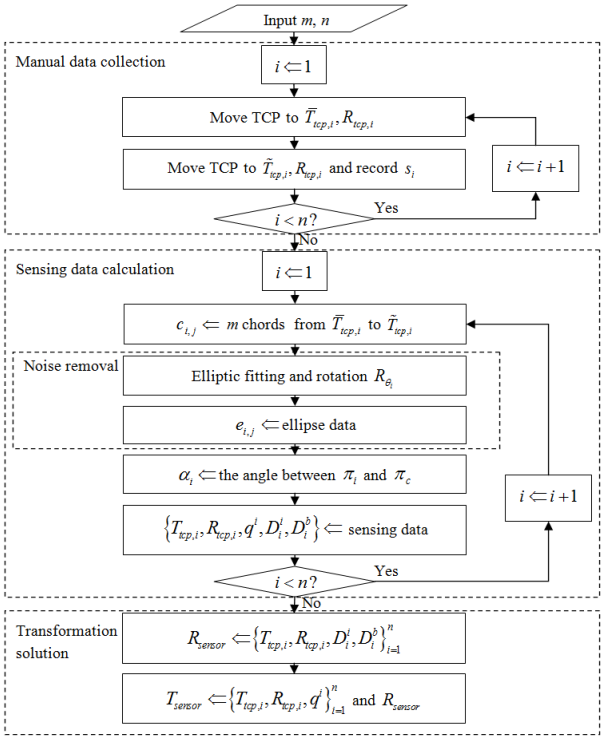


Fig. 3. Flowchart of the noise-tolerant calibration.

A. Manual data collection

In each scan, the 3D ball-based calibration methods [11]–[14] obtained a circular arc from the sensor and located the ball center directly from this arc. In our case, only one single chord can be obtained in each scan (Fig.2(a)). This chord alone is not sufficient to derive the coordinates of the disk center. Multiple chords are required in order to derive the i-th sensing data (see details in Section IV-B). However, users do not need to manually scan each chord, which will be painful and time consuming. According to Fig.3, users only need to manually select two different TCP positions $\bar{T}_{tcp,i}$ and $\tilde{T}_{tcp,i}$ with a fixed orientation $R_{tcp,i}$. With $M_i = \tilde{T}_{tcp,i} - \bar{T}_{tcp,i}$, the scanning system can automatically translate TCP from $\bar{T}_{tcp,i}$ by $M_i/(m-1)$ for $(m-1)$ times to obtain m chords $\{c_{i,j}\}_{j=1}^m$ (the black chords in Fig.4). To achieve a better calibration result, $\tilde{T}_{tcp,i}$ is selected such that the TCP's moving direction M_i is neither parallel nor perpendicular to the calibration disk, i.e.

$$M_i \nparallel \pi_c, M_i \not\perp \pi_c. \quad (2)$$

With the above constraints, users can choose different $\bar{T}_{tcp,i}$ and $\tilde{T}_{tcp,i}$. The scanning can be within half of the disk (Fig.4(a)) or the whole disk (Fig.4(b)).

Besides $\bar{T}_{tcp,i}$, $\tilde{T}_{tcp,i}$, and $R_{tcp,i}$, users also need to collect a value s_i, which is defined according to the TCP movement from $\bar{T}_{tcp,i}$ to $\tilde{T}_{tcp,i}$. If it follows the y^+ direction in the sensor coordinate system (Fig.5(a)), then $s_i = 1$. Otherwise (Fig.5(b)), $s_i = -1$.

Referring to Fig.5, suppose $\bar{\Gamma}_i$ is the sensor coordinate system in the sensor plane $\bar{\pi}_i$. If $\tilde{T}_{tcp,i}$ is formulated in $\bar{\Gamma}_i$ as $\tilde{T}^{\bar{\Gamma}_i}_{tcp,i} = (T_{i,x}, T_{i,y}, T_{i,z})$, s_i can also be defined as

$$s_i = \begin{cases} 1, & T_{i,y} > 0, \\ -1, & T_{i,y} < 0. \end{cases} \tag{3}$$

The value s_i is necessary for ball-based calibration algorithms as well (see details in Section IV-A). The value s_i gives users more degrees of freedom during data collection. Users can easily have $s_i = 1$ for all sensing data.

B. Sensing data calculation

For the i-th sensing data, sensor automatically scan each chord $c_{i,j} = [a_{i,j}, b_{i,j}]$ (Fig.2(a)) and provide 2D chord data as

$$a_{i,j} = (a_{i,j,x}, 0, a_{i,j,z}), \quad b_{i,j} = (b_{i,j,x}, 0, b_{i,j,z}). \tag{4}$$

These chord data can be proved to be on the same ellipse in the plane $\bar{\pi}_i$ (Fig.6, see details in Section IV-C), and there may be noise within these chord data (see details in Section IV-D). Therefore, according to Fig.3, these chord data are not directly used for sensing data calculation. Instead, ellipse fitting algorithm [21] is adopted for noise removal. The fitted ellipse will be used for calculating the desired sensing data.

There is an angle between x_i and π_c (Fig.7(a)). Applying a rotation matrix R_{θ_i} in the sensor coordinate system $\bar{\Gamma}_i$ in the sensor plane $\bar{\pi}_i$, $x_i y_i z_i$ becomes the rotated coordinate system $\bar{x}_i y_i \bar{z}_i$ (Fig.7(b)). The rotation matrix R_{θ_i} can be selected such that $\bar{x}_i \parallel \pi_c$. In order to get the desired R_{θ_i}, the chord direction for the i-th sensing data is defined as (Fig.7(a))

$$c_i = \frac{1}{m}\sum_{j=1}^{m}(b_{i,j} - a_{i,j}).$$

Then, the rotation matrix R_{θ_i} is defined by rotating x_i to c_i along y_i with angle θ_i. After rotation, points $\{e_{i,j}\}_{j=1}^{5}$ can be derived from the ellipse under the rotated coordinate system $\bar{\Gamma}_{i,r}$ in the plane $\bar{\pi}_i$ as follows (Fig.6, see details in Section IV-E):

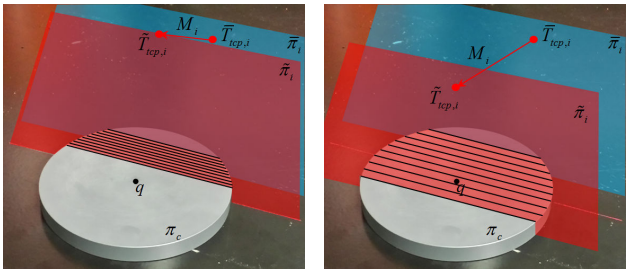
- $e_{i,1}$ and $e_{i,2}$ are two ellipse points whose tangent directions are parallel to $\bar{x}_i$ axis,
- $e_{i,3}$ is the center of the ellipse, and
- $e_{i,4}/e_{i,5}$ is the intersection of the chord $e_{i,1}e_{i,2}$ with the first/last chord.

The coordinates for $e_{i,j}$ in $\bar{\Gamma}_{i,r}$ and $\bar{\Gamma}_i$ can be calculated as

$$e^{\bar{\Gamma}_{i,r}}_{i,j} = (e_{i,j,x}, 0, e_{i,j,z}), \tag{5}$$

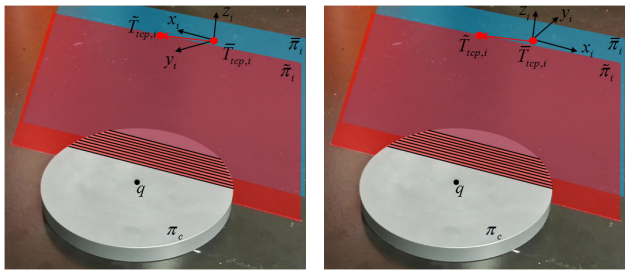
$$e^{\bar{\Gamma}_i}_{i,j} = R^{-1}_{\theta_i}(e_{i,j,x}, 0, e_{i,j,z}). \tag{6}$$

Without loss of generality, as presented in Fig.6, $e_{i,1}$ and $e_{i,2}$ can be selected such that $\overrightarrow{e_{i,1}e_{i,2}}$ and $\overrightarrow{e_{i,4}e_{i,5}}$ are of the same direction.



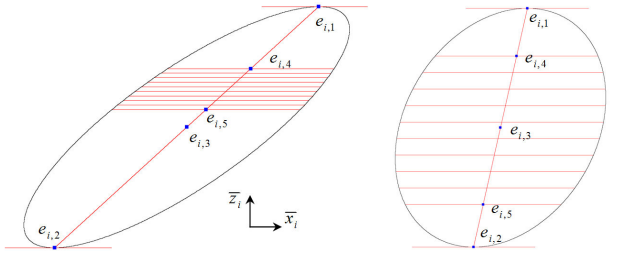
(a) (b)

Fig. 4. Multiple chords scanning with $m = 10$: (a) scan before q, and (b) scan over q.



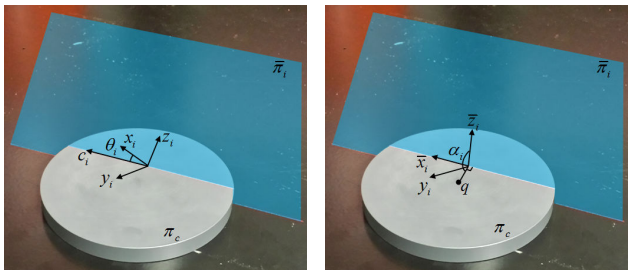
(a) (b)

Fig. 5. Two possibilities: (a) $s_i = 1$, and (b) $s_i = -1$.



(a) (b)

Fig. 6. Different ellipses in $\bar{\pi}_i$: (a) Fig.4(a), and (b) Fig.4(b).



(a) (b)

Fig. 7. Two coordinate systems in the sensor plane $\bar{\pi}_i$: (a) $\bar{\Gamma}_i$ and (b) $\bar{\Gamma}_{i,r}$.

For $j = 1, 2, \cdots, 5$, denote

$$\eta_{i,j} = \frac{(e_{i,j} - e_{i,5}) \cdot (e_{i,4} - e_{i,5})}{|e_{i,4} - e_{i,5}|^2}, \tag{7}$$

and

$$T_{i,j} = \eta_{i,j} \bar{T}_{tcp,i} + (1 - \eta_{i,j}) \tilde{T}_{tcp,i}. \tag{8}$$

Then, $T_{i,1}$ and $T_{i,2}$ are two TCP positions where the sensor plane tangent to the disk, and $T_{i,3}$ is the TCP position where the sensor plane passes through the disk center (see details in Section IV-F).

With Eq.(5) and Eq.(8), an angle α_i can be obtained as (Fig.7(b), see details in Section IV-G)

$$\alpha_i = \arccos\left(\frac{\|T_{i,1} - T_{i,2}\|^2 - \|e_{i,1} - e_{i,2}\|^2 - 4r_c^2}{4r_c\,(e_{i,1,z} - e_{i,2,z})}\right). \tag{9}$$

Then, the i-th sensing data $\left\{T_{tcp,i}, R_{tcp,i}, q^i, D_i^i, D_i^b\right\}$ is calculated as

$$T_{tcp,i} = T_{i,3}, \tag{10}$$

$$q^i = R_{\theta_i}^{-1}\left(e_{i,3,x}, 0, e_{i,3,y}\right), \tag{11}$$

$$D_i^i = R_{\theta_i}^{-1}\left(e_{i,1,x} - e_{i,2,x}, 2r_c s_i \sin\alpha_i, e_{i,1,z} - e_{i,2,z} + 2r_c \cos a_i\right), \tag{12}$$

$$D_i^b = T_{i,2} - T_{i,1}. \tag{13}$$

Details on how to derive these formulae will be presented in Section IV-G.

C. Transformation solution

Robot-sensor calibration is based on all sensing data $\left\{T_{tcp,i}, R_{tcp,i}, q^i, D_i^i, D_i^b\right\}_{i=1}^n$. Define two $3 \times n$ matrices as

$$\begin{aligned} X &= \left[D_1^1, D_2^2, \cdots, D_n^n\right], \\ Y &= \left[R_{tcp,1}^{-1} D_1^b, R_{tcp,2}^{-1} D_2^b, \cdots, R_{tcp,n}^{-1} D_n^b\right]. \end{aligned} \tag{14}$$

Apply singular value decomposition (SVD [22]) on the 3×3 matrix $H = XY^T$ as

$$H = USV^T, \tag{15}$$

where U and V are 3×3 orthogonal matrices while S is an 3×3 diagonal matrix with nonnegative real numbers on the diagonal. Then, the rotation matrix can be derived as

$$R_{sensor} = VU^T. \tag{16}$$

Coupling with

$$\begin{aligned} R_i &= R_{tcp,i} - R_{tcp,i+1}, \\ N_i &= R_{tcp,i+1} R_{sensor} q^{i+1} - R_{tcp,i} R_{sensor} q^i + T_{tcp,i+1} - T_{tcp,i}, \end{aligned} \tag{17}$$

a $3\,(n-1) \times 3$ matrix M and a $3\,(n-1) \times 1$ matrix N can be defined as

$$M = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{n-1} \end{bmatrix}, N = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_{n-1} \end{bmatrix}. \tag{18}$$

Then, the translation vector can be formulated as

$$T_{sensor} = \left(M^T M\right)^{-1}\left(M^T N\right), \tag{19}$$

where M^T is the transpose of M. Section IV-H will detail how to formulate Eq.(16) and Eq.(19).

D. Calibration accuracy

In [13], the accuracy is estimated as the difference between the radius of the calibration target and the radius of the reconstructed target. In fact, as discussed during T_{sensor} calculation in [12], a ball can be reconstructed only if R_{sensor} is known. Thus, T_{sensor} will not affect the radius difference. In fact, radius difference only reflect the accuracy of R_{sensor}. Instead of the radius difference, the position difference should be used to estimate the calibration accuracy [23].

After calibration, $\left\{T_{tcp,i}, R_{tcp,i}, q^i\right\}$ can be used to derive a disk center following Eq.(1) as

$$q_i^b = R_{tcp,i} R_{sensor} q^i + R_{tcp,i} T_{sensor} + T_{tcp,i}.$$

The disk center is estimated as

$$q^b = \sum_{i=1}^{n} q_i^b \Big/ n. \tag{20}$$

The calibration accuracy is

$$\max_k \left\|q^b - q_k^b\right\|.$$

The calibration accuracy is affected by the following factors:

- the resolution and accuracy of the robotic arm,
- the resolution and accuracy of the sensor, and
- the accuracy of the ellipse fitting on the training data.

IV. Algorithm Formulation

During manual data collection, $R_{tcp,i}$, $\bar{T}_{tcp,i}$, $\tilde{T}_{tcp,i}$, and s_i are collected. After that, chord data $c_{i,j}$ are acquired from the sensor. This section will detail how the calibration algorithm is developed based on these data.

A. Necessity of s_i

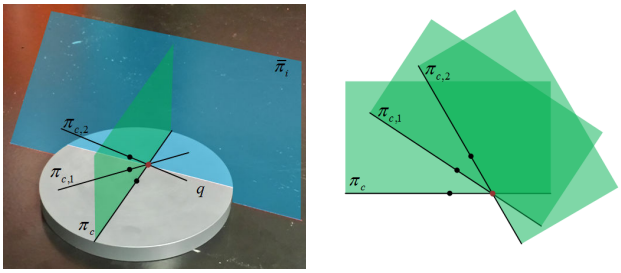
The sign s_i in Eq.(3) is recorded during manual data collection, and used in Eq.(12) during sensing data calculation. The ball-based calibration methods [11]–[14] also need such a value when formulating the ball center in the sensor coordinate system. Suppose the ball centers at c_b with radius r_b. In each scan, the sensor acquires a circular arc from the ball. Circle fitting algorithm is adopted to get the radius r_d and the center $c_d = (c_{d,x}, 0, c_{d,z})$ of the circle in the sensor plane. Then, the ball center can be formulated in the sensor coordinate system as

$$c_b = \left(c_{d,x}, s_i\sqrt{r_b^2 - r_d^2}, c_{d,z}\right),$$

where s_i can be -1 or 1. If the disk center locates on y^+ direction in the sensor coordinate system, $s_i = 1$. Otherwise, $s_i = -1$.

B. Multiple chords based calculation

For TCP at any position, only a single chord is provided from the sensor (Fig.8(a)). If the chord passes through the disk center q, the coordinates q^i in the sensor coordinate system can be obtained as the chord center. Otherwise, q^i cannot be directly derived from the single chord. Consider a scenario where the robotic arm and the sensor are fixed while the circular disk rotates along the chord from π_c to $\pi_{c,1}$ and $\pi_{c,2}$ (Fig.8(b)). The chord data from the sensor remains unchanged; because the geometric distance from the chord to the sensor is fixed. Meanwhile, the coordinates q^i keep changing during the rotation. This means that a single chord alone is not sufficient to define q^i. The following sections will present that minimally two chords are required to locate the disk center. Moreover, to tolerate the noise during data acquisition, multiple chords should be collected.



(a) (b)

Fig. 8. Disk rotation along the chord: (a) the sensor scan one chord on the intersection between the disk plane and $\bar{\pi}_i$, and (b) rotating the disk plane along the chord leads to the same chord from the sensor. (The green plane is perpendicular to both the disk plane and the chord)

C. Chord projection

Chords $c_{i,j}$ in Fig.4 are on the disk. When the sensor scans these chords, the sensor coordinate system move. Thus, the readings from the sensor in Eq.(4) are in different coordinate systems. This section will prove that the chord data in Eq.(4) are on the same ellipse in the plane $\bar{\pi}_i$.

For the i-th sensing data, referring to Fig.4, TCP is translated by $M_i/(m-1)$ for $(m-1)$ times to move from $\bar{T}_{tcp,i}$ to $\tilde{T}_{tcp,i}$. Simultaneously, m parallel sensor planes $\{\pi_{i,j}\}_{j=1}^m$ are derived satisfying $\pi_{i,1} = \bar{\pi}_i$ and $\pi_{i,m} = \tilde{\pi}_i$. In these sensor planes, m sensor coordinate systems $\{\Gamma_{i,j} = o_{i,j}x_iy_iz_i\}_{j=1}^m$ also translate uniformly with (Fig.9)

$$o_{i,j} = o_{i,1} + \frac{j-1}{m-1}M_i, \tag{21}$$

where $\bar{\Gamma}_i = \Gamma_{i,1} = o_{i,1}x_iy_iz_i$ is the sensor coordinate system in the sensor plane $\bar{\pi}_i$. Since there is only one translation between the two coordinate systems $\Gamma_{i,1}$ and $\Gamma_{i,j}$, the coordinates of a vector in $\Gamma_{i,1}$ equal to its coordinates in $\Gamma_{i,j}$, i.e.

$$M_i^{o_{i,1}x_iy_iz_i} = M_i^{o_{i,j}x_iy_iz_i},$$

and the coordinates of any point p in $\Gamma_{i,1}$ and $\Gamma_{i,j}$ satisfy

$$\begin{aligned} p^{o_{i,1}x_iy_iz_i} &= p^{o_{i,j}x_iy_iz_i} + \tfrac{j-1}{m-1}M_i^{o_{i,1}x_iy_iz_i} \\ &= p^{o_{i,j}x_iy_iz_i} + \tfrac{j-1}{m-1}M_i^{o_{i,j}x_iy_iz_i}. \end{aligned} \tag{22}$$

Reformulate sensor data in Eq.(4) for each chord $c_{i,j}$ in $\Gamma_{i,j}$ as (black parallel line segments in Fig.9(a))

$$c_{i,j}^{o_{i,j}x_iy_iz_i} = (c_{i,j,x}, 0, c_{i,j,z}). \tag{23}$$

Define a line segment $\tilde{c}_{i,j}$ paralleling to $c_{i,j}$ as

$$\tilde{c}_{i,j} = c_{i,j} - \frac{j-1}{m-1}M_i, \tag{24}$$

and formulate it in $\Gamma_{i,j}$ as

$$\tilde{c}_{i,j}^{o_{i,j}x_iy_iz_i} = c_{i,j}^{o_{i,j}x_iy_iz_i} - \frac{j-1}{m-1}M_i^{o_{i,j}x_iy_iz_i}. \tag{25}$$

On the other hand, substituting $p = \tilde{c}_{i,j}$ into Eq.(22) gives

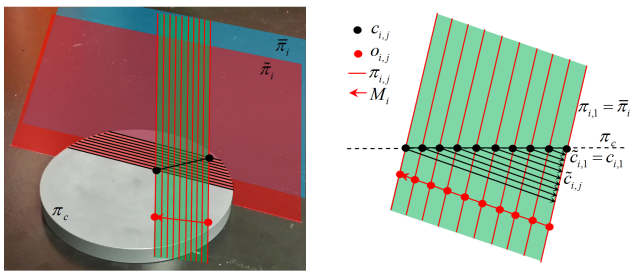
$$\tilde{c}_{i,j}^{o_{i,1}x_iy_iz_i} = \tilde{c}_{i,j}^{o_{i,j}x_iy_iz_i} + \frac{j-1}{m-1}M_i^{o_{i,j}x_iy_iz_i}. \tag{26}$$

Substituting Eq.(25) into Eq.(26) leads to

$$\tilde{c}_{i,j}^{o_{i,1}x_iy_iz_i} = c_{i,j}^{o_{i,j}x_iy_iz_i} = (c_{i,j,x}, 0, c_{i,j,z}). \tag{27}$$

Since the y coordinate of $\tilde{c}_{i,j}^{o_{i,1}x_iy_iz_i}$ is zero, $\tilde{c}_{i,j}$ locates in the plane $\bar{\pi}_i$. In fact, $\{\tilde{c}_{i,j}\}_{j=1}^m$ are the projection of $\{c_{i,j}\}_{j=1}^m$ to the plane $\bar{\pi}_i$ following a direction parallel to M_i (black arrows in Fig.9(b)). Thus, the input sensor data $\{(c_{i,j,x}, 0, c_{i,j,z})\}_{j=1}^m$ are not only the coordinates of chords $\{c_{i,j}\}_{j=1}^m$ in different coordinate systems $\{\Gamma_{i,j}\}_{j=1}^m$, i.e. $\{c_{i,j}^{o_{i,j}x_iy_iz_i}\}_{j=1}^m$, but also the coordinates of line segments $\{\tilde{c}_{i,j}\}_{j=1}^m$ in the same coordinate system $\bar{\Gamma}_i$, i.e. $\{\tilde{c}_{i,j}^{o_{i,1}x_iy_iz_i}\}_{j=1}^m$.

Geometrically, all $\{c_{i,j}\}_{j=1}^m$ are on the same disk. However, not all sensor data in Eq.(23) are on a disk, because they are under different coordinate systems $\{\Gamma_{i,j}\}_{j=1}^m$. With the constraints in Eq.(2), the projection of a circular disk to a plane following a direction parallel to M_i is an elliptic disk. Thus, $\{(c_{i,j,x}, 0, c_{i,j,z})\}_{j=1}^m$ are chords of this elliptic disk. The projected ellipse in $\bar{\pi}_i$ can be acquired using ellipse fitting algorithm [21], and the chord data in Eq.(4) are on the projected ellipse.



(a) (b)

Fig. 9. A plane perpendicular to π_c and passing through $o_{i,j}$: (a) The plane in 3D with red lines as its intersection with $\pi_{i,j}$ and red arrow as the moving direction, and (b) the plane in 2D with black dots as $c_{i,j}$, red dots as $o_{i,j}$, black line as π_c, red lines as $\pi_{i,j}$, red arrow as M_i, and black arrows as projection directions.

D. Noise analysis in sensor data

In the sensor plane $\bar{\pi}_i$, the chords $\{\tilde{c}_{i,j}\}_{j=1}^{m}$ in Eq.(27) center at $\{s_{i,j}\}_{j=1}^{m}$. According to the construction of $e_{i,1}$ and $e_{i,2}$, all $s_{i,j}$ are geometrically on the chord $e_{i,1}e_{i,2}$. Formulate $s_{i,j}$ in $\bar{\Gamma}_i$ as $s_{i,j}^{o_{i,1}x_iy_iz_i} = (s_{i,j,x}, 0, s_{i,j,z})$. Then, ideally, $(s_{i,j,x}, 0, s_{i,j,z})$ should be uniformly distributed on a line. However, in the actual scanned data, $(s_{i,j,x}, 0, s_{i,j,z})$ may not be on a same line (Fig.10). This is due to the noise introduced by the surface condition of the calibration target, the resolution of the sensor, the vibration of the robotic arm during sensing data acquisition, and so on.

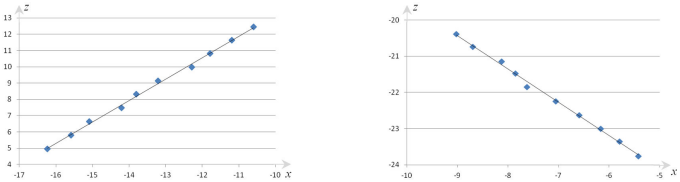


Fig. 10. Data $(s_{i,j,x}, 0, s_{i,j,z})$ from the sensor do not lie exactly on the same line.

To tolerate these noises, the proposed algorithm uses the fitted ellipse instead of the original chords for sensing data calculation. Theoretically, minimally two chords are required to enable the ellipse fitting. However, to better tolerate the noises, $m \geq 100$ chords are collected for sensing data calculation (see details in Section V-B). For illustration purpose, figures in this paper adopts $m = 10$.

E. Elliptic points

During sensing data calculation, five elliptic points $\{e_{i,j}\}_{j=1}^{5}$ on the projected ellipse instead of the raw sensor data are used. This section will introduce how these points are formulated as Eq.(5).

Referring to Fig.11(a), on the circular disk, points

$$q_{i,j}, j = 1, 2, \cdots, 5 \tag{28}$$

are defined in the plane π_c such that

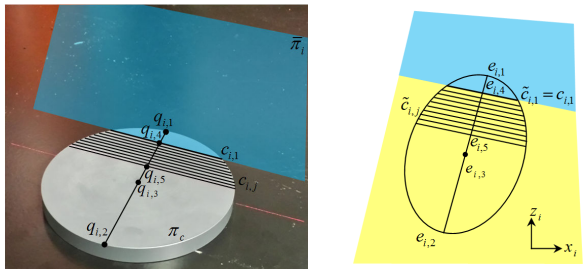
- $q_{i,1}q_{i,2}$ is the diameter geometrically perpendicular to the chords $\{c_{i,j}\}_{j=1}^{m}$,
- $q_{i,3} = q$ is the disk center, and
- $q_{i,4}$ and $q_{i,5}$ are intersections of $q_{i,1}q_{i,2}$ to $c_{i,1}$ and $c_{i,m}$, respectively.

Projecting $q_{i,j}$ to the plane $\bar{\pi}_i$ following a direction parallel to M_i (Fig.11(b)) leads to the projected points

$$e_{i,j}, j = 1, 2, \cdots, 5. \tag{29}$$

According to the definition,

- ellipse chords $\{\tilde{c}_{i,j}\}_{j=1}^{m}$ are the projections of circular chords $\{c_{i,j}\}_{j=1}^{m}$,
- ellipse chord $e_{i,1}e_{i,2}$ is the projection of the diameter $q_{i,1}q_{i,2}$,
- center $e_{i,3}$ is the projection of the center $q = q_{i,3}$, and
- points $e_{i,4}$ and $e_{i,5}$ are intersections of $e_{i,1}e_{i,2}$ to $\tilde{c}_{i,1}$ and $\tilde{c}_{i,m}$, respectively.



(a) (b)

Fig. 11. Elliptic points via projection: (a) $q_{i,j}$ in π_c, and (b) $e_{i,j}$ in $\bar{\pi}_i$. (The blue area is above π_c and the yellow area is below π_c).

In the plane $\bar{\pi}_i$, the chord $e_{i,1}e_{i,2}$ passes through the centers of all ellipse chords parallel to $\tilde{c}_{i,j}$. Therefore, the tangent directions at $e_{i,1}$ and $e_{i,2}$ are parallel to $\tilde{c}_{i,j}$ as well. After the rotation R_{θ_i}, Fig.11(b) becomes Fig.6(a), and the tangent directions at $e_{i,1}$ and $e_{i,2}$ are parallel to the axis $\bar{x}_i$. In the rotated coordinate system $\bar{\Gamma}_{i,r}$, each point $e_{i,j}$ is formulated in Eq.(5).

F. TCP positions

During data collection, users only need to manually collect two TCP positions $\bar{T}_{tcp,i}$ and $\tilde{T}_{tcp,i}$. With the orientation $R_{tcp,i}$ fixed, the proposed algorithm calculates other TCP positions where the sensor plane is passing through the disk center or tangent to the disk. Referring to Fig.12(a), suppose that TCP positions $\{T_{i,j}\}_{j=1}^{5}$ are linear combinations of $\bar{T}_{tcp,i}$ and $\tilde{T}_{tcp,i}$. When TCP moves to $T_{i,j}$, the sensor plane is $\phi_{i,j}$ and the sensor coordinate system is $\Phi_{i,j}$. With $q_{i,j}$ defined in Eq.(28), the positions $\{T_{i,j}\}_{j=1}^{5}$ are taken such that:

- $\phi_{i,1}$ and $\phi_{i,2}$ are tangent to the disk at the points $q_{i,1}$ and $q_{i,2}$, respectively.
- $\phi_{i,3}$ passes through the disk center $q = q_{i,3}$.
- $\phi_{i,4} = \bar{\pi}_i$ passes through the point $q_{i,4}$.
- $\phi_{i,5} = \tilde{\pi}_i$ passes through the point $q_{i,5}$.

This section is to prove that $T_{i,j}$ satisfies Eq.(8).

Referring to Fig.12(b), since $\{e_{i,j}\}_{j=1}^{5}$ are parallel projection of $\{q_{i,j}\}_{j=1}^{5}$, similar with the chord projection in Eq.(27), the coordinates of $q_{i,j}$ in the sensor coordinate system $\Phi_{i,j}$ equal to the coordinates of $e_{i,j}$ in the sensor coordinate system $\Phi_{i,4}$, i.e.

$$q_{i,j}^{\Phi_{i,j}} = e_{i,j}^{\bar{\Gamma}_i} = e_{i,j}^{\Phi_{i,4}}.$$

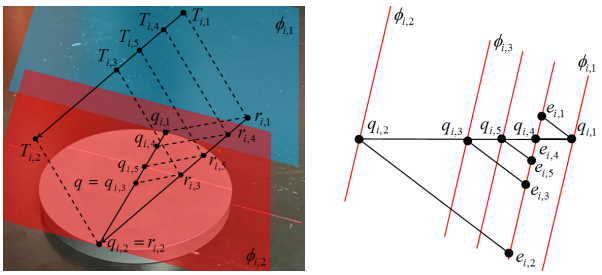
Since $\phi_{i,4} = \bar{\pi}_i$ and $\Phi_{i,4} = \bar{\Gamma}_i$, coupling with Eq.(5) and Eq.(6), the coordinates of $q_{i,j}$ in the coordinate system $\Phi_{i,j}$ are

$$q_{i,j}^{\Phi_{i,j}} = e_{i,j}^{\Phi_{i,4}} = e_{i,j}^{\bar{\Gamma}_i} = R_{\theta_i}^{-1}\left(e_{i,1,x}, 0, e_{i,1,z}\right), \tag{30}$$

and its coordinates in the rotated coordinate system $\Phi_{i,j,r}$ are

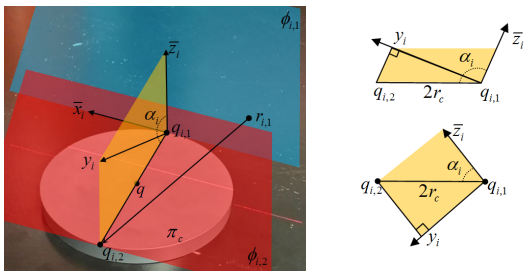
$$q_{i,j}^{\Phi_{i,j,r}} = e_{i,j}^{\Phi_{i,4,r}} = e_{i,j}^{\bar{\Gamma}_{i,r}} = \left(e_{i,1,x}, 0, e_{i,1,z}\right). \tag{31}$$

When TCP moves linearly, the sensor plane and the rotated coordinate system move linearly as well. Consider a point



(a) (b)

Fig. 12. $T_{i,j}$ formulation: (a) relationship among $T_{i,j}$,$r_{i,j}$, and $q_{i,j}$, and (b) parallel projection from $q_{i,j}$ to $e_{i,j}$.



(a) (b)

Fig. 13. α_i formulation: (a) the rotated coordinate system in $\phi_{i,1}$, and (b) different cases for α_i.

r_i in the sensor plane with fixed coordinates in the rotated coordinate system as $(e_{i,1,x}, 0, e_{i,1,z})$. Then, point r_i moves linearly as well. Referring to Fig.12(a), when TCP moves to $T_{i,j}$, r_i moves to $r_{i,j}$ in $\phi_{i,j}$ with

$$r_{i,j}^{\Phi_{i,j}} = R_{\theta_i}^{-1}\left(e_{i,1,x}, 0, e_{i,1,z}\right). \tag{32}$$

As such, all $r_{i,j}$ are on the same line, and

$$T_{i,j}r_{i,j} \parallel T_{i,1}r_{i,1}, \qquad q_{i,j}r_{i,j} \parallel q_{i,1}r_{i,1}. \tag{33}$$

Coupling with Eq.(7), $e_{i,j}$ can be formulated as

$$e_{i,j} = \eta_{i,j}e_{i,4} + (1-\eta_{i,j})\, e_{i,5}.$$

Referring to Fig.12(b), since $\{e_{i,j}\}_{j=1}^{5}$ are parallel projection of $\{q_{i,j}\}_{j=1}^{5}$, line segments $\{e_{i,j}e_{i,5}\}_{j=1}^{4}$ and $\{q_{i,j}q_{i,5}\}_{j=1}^{4}$ are proportional segments, i.e.

$$\frac{\|e_{i,j}e_{i,5}\|}{\|e_{i,4}e_{i,5}\|} = \frac{\|q_{i,j}q_{i,5}\|}{\|q_{i,4}q_{i,5}\|}.$$

Thus

$$q_{i,j} = \eta_{i,j}q_{i,4} + (1-\eta_{i,j})\, q_{i,5}.$$

Similarly, Eq.(33) leads to

$$r_{i,j} = \eta_{i,j}r_{i,4} + (1-\eta_{i,j})\, r_{i,5},$$
$$T_{i,j} = \eta_{i,j}T_{i,4} + (1-\eta_{i,j})\, T_{i,5},$$

which proves Eq.(8).

G. Sensing data formation

The i-th sensing data is $\left\{T_{tcp,i}, R_{tcp,i}, q^i, D_i^i, D_i^b\right\}$, where $R_{tcp,i}$ is from the robotic arm. According to the definition of $T_{i,j}$ in Section IV-F, $T_{i,3}$ is the TCP position where the sensor plane passes through q. Thus, $T_{tcp,i}$ can be defined as Eq.(10). According to Eq.(30), $q^i = q_{i,3}^i = q_{i,3}^{\Phi_{i,3}} = e_{i,j}^{\bar{\Gamma}_i}$, which proves Eq.(11). The TCP offset D_i in the base coordinate system is given by Eq.(13). This section will show how to derive Eq.(12).

Γ_i is the sensor coordinate system for the sensor plane passing through the disk center. According to the definition, $\Phi_{i,3} = \Gamma_i$. Suppose $\Phi_{i,j,r}$ is the rotated coordinate system after applying R_{θ_i} on $\Phi_{i,j}$. To derive the sensing data, different points need to be formulated in the same coordinate system $\Phi_{i,1,r}$.

Referring to Eq.(1), two different points $r_{i,1}$ and $q_{i,2}$ satisfy

$$\begin{aligned} r_{i,1}^b &= R_{tcp,i}R_{sensor}r_{i,1}^{\Gamma_i} + R_{tcp,i}T_{sensor} + T_{tcp,i}, \\ q_{i,2}^b &= R_{tcp,i}R_{sensor}q_{i,2}^{\Gamma_i} + R_{tcp,i}T_{sensor} + T_{tcp,i}. \end{aligned} \tag{34}$$

Referring to Fig.12(a), the TCP offset is

$$D_i = T_{i,2} - T_{i,1} = q_{i,2} - r_{i,1}. \tag{35}$$

Coupling with Eq.(34), D_i can be formulated as

$$D_i^b = q_{i,2}^b - r_{i,1}^b = R_{tcp,i}R_{sensor}D_i^{\Gamma_i} = R_{tcp,i}R_{sensor}D_i^i. \tag{36}$$

Since $p^{\Gamma_i} = R_{\theta_i}^{-1}p^{\Gamma_{i,r}}$ holds for any point p, the above equation becomes

$$D_i^b = R_{tcp,i}R_{sensor}R_{\theta_i}^{-1}D_i^{\Gamma_{i,r}}. \tag{37}$$

The coordinates of a vector remain unchanged in $\Phi_{i,1,r}$ and $\Phi_{i,3,r}$, because there is only a translation between the two coordinate systems. Substituting

$$D_i^{\Gamma_{i,r}} = D_i^{\Phi_{i,3,r}} = D_i^{\Phi_{i,1,r}} \tag{38}$$

into Eq.(37) gives

$$D_i^b = R_{tcp,i}R_{sensor}R_{\theta_i}^{-1}D_i^{\Phi_{i,1,r}}. \tag{39}$$

In Fig.13, $\bar{x}_i y_i \bar{z}_i$ is the rotated coordinate system, and α_i is an angle from $\bar{z}_i$ to $\overrightarrow{q_{i,1}q_{i,2}}$. Noted that $\pi_c \parallel \bar{x}_i$, $y_i \perp \bar{x}_i$, $\bar{z}_i \perp \bar{x}_i$ and $q_{i,1}q_{i,2} \perp \bar{x}_i$. The yellow plane spanned by y_i, $\bar{z}_i$ and $q_{i,1}q_{i,2}$ is perpendicular to π_c and $\bar{x}_i$ (Fig.13(a)). Thus, α_i is an angle between π_c and $\phi_{i,j}$. Fig.13(b) shows two possibilities for α_i: $\alpha_i \geq 90°$ and $\alpha_i < 90°$. According to Eq.(2), although TCP movement is not perpendicular to π_c, $\phi_{i,j}$ may be perpendicular to π_c, i.e. $\alpha_i = 90°$.

To derive α_i, points $q_{i,1}$, $q_{i,2}$ and $r_{i,1}$ also need to be formulated in the same coordinate system $\Phi_{i,1,r}$. Eq.(31) and Eq.(32) gives

$$q_{i,1}^{\Phi_{i,1,r}} = \left(e_{i,1,x}, 0, e_{i,1,z}\right). \tag{40}$$

$$r_{i,1}^{\Phi_{i,1,r}} = \left(e_{i,2,x}, 0, e_{i,2,z}\right). \tag{41}$$

According to the definition of $q_{i,j}$ in Eq.(28) and s_i in Eq.(3), if $s_i = 1$ (Fig.13(b)), $q_{i,2}$ is on y^+ direction of $\Phi_{i,1,r}$. Otherwise, $q_{i,2}$ is on y^- direction of $\Phi_{i,1,r}$. Thus, in the yellow plane

perpendicular to $\bar{x}_i$ (Fig.13), the offset $q_{i,2} - q_{i,1}$ depends on s_i as

$$\left(q_{i,2} - q_{i,1}\right)^{\Phi_{i,1,r}} = 2r_c\left(0, s_i \cdot \sin\alpha_i, \cos\alpha_i\right).$$

Coupling with Eq.(40), $q_{i,2}$ can be formulated as

$$\begin{aligned} q_{i,2}^{\Phi_{i,1,r}} &= q_{i,1}^{\Phi_{i,1,r}} + 2r_c\left(0, s_i \cdot \sin\alpha_i, \cos\alpha_i\right) \\ &= \left(e_{i,1,x}, 2r_c \cdot s_i \cdot \sin\alpha_i, e_{i,1,z} + 2r_c \cdot \cos\alpha_i\right). \end{aligned} \tag{42}$$

Combining Eq.(41) and Eq.(42), D_i defined in Eq.(35) can be formulated in $\Phi_{i,1,r}$ as

$$\begin{aligned} &D_i^{\Phi_{i,1,r}} = q_{i,2}^{\Phi_{i,1,r}} - r_{i,1}^{\Phi_{i,1,r}} \\ &= \left(e_{i,1,x} - e_{i,2,x}, 2r_c \cdot s_i \cdot \sin\alpha_i, e_{i,1,z} - e_{i,2,z} + 2r_c \cdot \cos\alpha_i\right). \end{aligned}$$

According to Eq.(38), Eq.(12) can be obtained from

$$D_i^i = D_i^{\Gamma_i} = R_{\theta_i}^{-1} D_i^{\Gamma_{i,r}} = R_{\theta_i}^{-1} D_i^{\Phi_{i,1,r}}.$$

On the other hand, Eq.(39) gives $\left\|D_i^b\right\| = \left\|D_i^{\Phi_{i,1,r}}\right\|$, i.e.

$$\begin{aligned} \left\|T_{i,2} - T_{i,1}\right\|^2 = \; & \left(e_{i,1,x} - e_{i,2,x}\right)^2 + \left(e_{i,1,z} - e_{i,2,z}\right)^2 \\ & + 4r_c^2 + 4r_c\left(e_{i,1,z} - e_{i,2,z}\right)\cos\alpha_i. \end{aligned}$$

According to Eq.(2), the sensor plane does not move parallel to π_c, i.e., $e_{i,1,z} \neq e_{i,2,z}$. Therefore,

$$\cos\alpha_i = \frac{\left\|T_{i,2} - T_{i,1}\right\|^2 - \left(e_{i,1,x} - e_{i,2,x}\right)^2 - \left(e_{i,1,z} - e_{i,2,z}\right)^2 - 4r_c^2}{4r_c\left(e_{i,1,z} - e_{i,2,z}\right)}, \tag{43}$$

which gives Eq.(9).

H. Algorithm design

Robot-sensor calibration is to derive the desired R_{sensor} and T_{sensor} from all sensing data. Eq.(36) gives

$$R_{sensor} D_i^i = R_{tcp,i}^{-1} D_i^b, i = 1, 2, \cdots, n,$$

which can be reformulated into

$$R_{sensor}\left[D_1^1, D_2^2, \cdots, D_n^n\right] = \left[R_{tcp,1}^{-1} D_1^b, R_{tcp,2}^{-1} D_2^b, \cdots, R_{tcp,n}^{-1} D_n^b\right]. \tag{44}$$

Coupling with notations in Eq.(14), the above equations become

$$R_{sensor} X = Y. \tag{45}$$

The unknown R_{sensor} in Eq.(45) can be solved using the singular value decomposition (SVD) [22], which gives the solution as Eq.(16).

On the other hand, according to Eq.(1), for $i = 1, \cdots, n-1$,

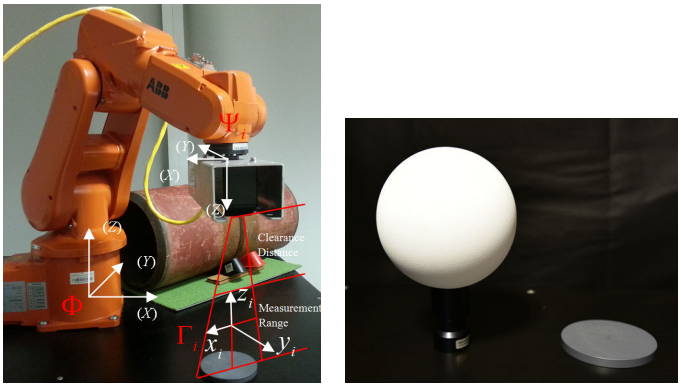
$$\begin{aligned} q^b &= R_{tcp,i+1} R_{sensor} q^{i+1} + R_{tcp,i+1} T_{sensor} + T_{tcp,i+1}, \\ q^b &= R_{tcp,i} R_{sensor} q^i + R_{tcp,i} T_{sensor} + T_{tcp,i}. \end{aligned}$$

The substraction of the two equations provides

$$\begin{aligned} &\left(R_{tcp,i+1} R_{sensor} q^{i+1} + R_{tcp,i+1} T_{sensor} + T_{tcp,i+1}\right) \\ &-\left(R_{tcp,i} R_{sensor} q^i + R_{tcp,i} T_{sensor} + T_{tcp,i}\right) = 0, \end{aligned}$$

which gives

$$\begin{aligned} \left(R_{tcp,i} - R_{tcp,i+1}\right) T_{sensor} = \; & R_{tcp,i+1} R_{sensor} q^{i+1} \\ & - R_{tcp,i} R_{sensor} q^i + T_{tcp,i+1} - T_{tcp,i}. \end{aligned}$$



(a) (b)

Fig. 14. Experimental setup: (a) the scanning system, and (b) calibration ball with radius 69.5mm and calibration disk with radius 50mm.

With notations in Eq.(17), the above equations can be reformulated into

$$R_i T_{sensor} = N_i, i = 1, 2, \cdots, n-1,$$

or equivalently

$$\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{n-1} \end{bmatrix} T_{sensor} = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_{n-1} \end{bmatrix}. \tag{46}$$

Substituting Eq.(18) into Eq.(46) leads to

$$M T_{sensor} = N. \tag{47}$$

The unknown T_{sensor} can be solved using the least square method with the result presented in Eq.(19).

V. Experimental Results

A. System setup and calibration

In the scanning system shown in Fig.14(a), an LMI GOCATOR 2340 [24] laser profile sensor is mounted onto an ABB IRB 120 robotic arm [25]. The unit for the coordinate system is millimeter (mm).

Denote the manually collected TCP positions and orientations used for the calibration as

$$(\bar{R}_i, \bar{T}_i, \tilde{R}_i, \tilde{T}_i), i \in \{1, 2, 3, 4\}. \tag{48}$$

Table I lists the four sensing data derived using the scanning system in Fig.14(a). The proposed noise-tolerant calibration algorithm derives the final transformation matrix as

$$\Lambda = \begin{bmatrix} 0.99996 & -0.00547 & -0.00716 & 6.0252 \\ -0.00544 & -0.99998 & 0.00288 & 54.18211 \\ -0.00718 & -0.00284 & -0.99997 & 364.25534 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{49}$$

Following Section III-D, the disk center in Eq.(20) is calculated as

$$q^b = \left(-319.622, -304.086, 11.48\right), \tag{50}$$

and the calibration accuracy is estimated as

$$\max_{i=1}^{4} \left\| q^b - q_i^b \right\| = 0.212mm, \tag{51}$$

which is jointly affected by the x-resolution of the sensor, which is 0.1mm, the robotic arm's positioning accuracy, and the ellipse fitting accuracy.

B. Simulation and validation

A simulated robotic-laser scanning system is set up with a simulated calibration matrix $\bar{\Lambda} = \Lambda$ in Eq.(49). A virtual disk centered at $\bar{q}^b = q^b$ in Eq.(50) with orientation $\bar{n}_c^b = (0, 0, 1)$ is used for collecting simulation data. TCP data in Eq.(48) are adopted for the simulation. The simulated chord data $\bar{a}_{i,j}$ and $\bar{b}_{i,j}$ are collected as the intersections of the virtual disk and the moving laser planes. Then, the calibration based on the simulated chord data derives the transformation matrix $\tilde{\Lambda}$ and the disk center $\tilde{q}^b$. Simulation results show that

$$\tilde{\Lambda} = \bar{\Lambda}, \quad \tilde{q}^b = \bar{q}^b,$$

which validates the proposed calibration algorithm.

The above simulation and validation do not take into account the noise in the chord data. In fact, the real chord data $a_{i,j}$ and $b_{i,j}$ in Eq.(4) may contain noise introduced by the accuracy of the robotic arm, the x-resolution of the sensor, the accuracy of the chord segmentation, and so on. Assume that such noise is the Gaussian noise $G(\sigma)$ with mean value 0 and standard deviation σ. A set of noise chord data from the simulated chord data is derived as

$$\begin{aligned} \tilde{a}_{i,j} &= \bar{a}_{i,j} + \frac{\tau_{i,j,1}}{\|\bar{b}_{i,j} - \bar{a}_{i,j}\|}(\bar{b}_{i,j} - \bar{a}_{i,j}), \\ \tilde{b}_{i,j} &= \bar{b}_{i,j} + \frac{\tau_{i,j,2}}{\|\bar{b}_{i,j} - \bar{a}_{i,j}\|}(\bar{b}_{i,j} - \bar{a}_{i,j}), \end{aligned} \tag{52}$$

where $i = 1, 2, 3, 4, j = 1, 2, \ldots, m$, and $\tau_{i,j,k}$ are Gaussian noise $G(\sigma)$.

For each standard deviation $\sigma \in \{0.1, 0.2, \ldots, 0.9\}$, 500 sets of noise chord data $\{\tilde{a}_{i,j,l}, \tilde{b}_{i,j,l}\}_{l=1}^{500}$ are simulated according to Eq.(52). The calibration based on the l-th set $\{\tilde{a}_{i,j,l}, \tilde{b}_{i,j,l}\}$ gives a transformation matrix, a disk center $\tilde{q}_l^b$, and a calibration accuracy $\varepsilon_l = \|\tilde{q}_l^b - \bar{q}^b\|$ (the error in reconstructing the disk center). The standard deviation of $\{\varepsilon_i\}_{l=1}^{500}$ is denoted as e, which indicates the calibration accuracy with respect to σ and m.

Fig.15 shows the trend of the calibration accuracy e with respect to σ and m. In general, with m fixed, e increases as σ increases. For example, with $m = 100$, as σ increases from 0.1 to 0.9, e increases from 0.052 to 0.449. On the other hand, with σ fixed, bigger m gives better accuracy e. In this simulation, when m increases from 10 to 200 while σ fixed at 0.4, the accuracy e increases from 0.591 to 0.135 (the forth curve from the bottom in Fig.15). According to curves in Fig.15, bigger m provides better accuracy. For the experimental setup in Fig.14(a), we adopt $m = 100$.

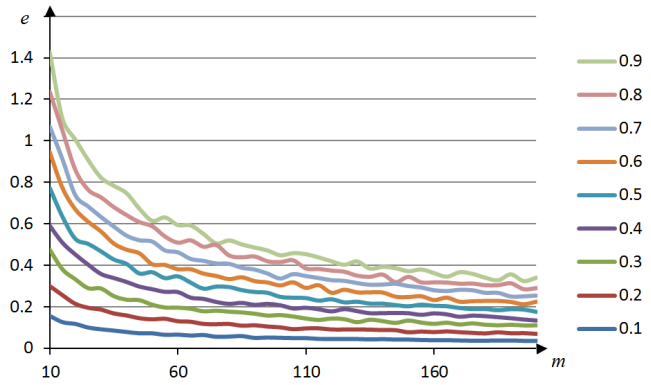


Fig. 15. Disk-based calibration accuracy with respect to σ and m (the horizontal coordinate is the number of chords m, the vertical coordinate is the calibration accuracy $e(mm)$, and curves in different colors correspond to different noise level $\sigma(mm)$).

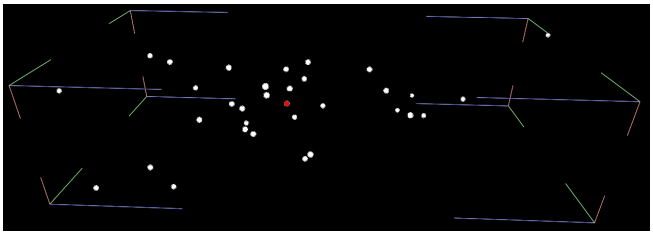


Fig. 16. Reconstructed disk centers in repeatability test (the white dots are disk centers from different tests; the red dot is the average of the 32 white dots; the size of the bounding box is $0.070mm \times 0.131mm \times 0.287mm$)

C. Repeatability

To analyze the repeatability of the proposed calibration algorithm, with position and orientation of the disk fixed, calibration data is collected based on the same set of TCP data in Eq.(48). The data collection is repeated for 32 times providing 32 calibration results. The disk centers $\left\{q_i^b\right\}^3 2_{i=1}$ are used to estimate the repeatability (the white dots in Fig.16). For $\left\{q_i^b\right\}^3 2_{i=1}$, the mean is denoted as (the red dot in Fig.16)

$$\dot{q} = \sum_{i=1}^{32} q_i^b \Big/ 32,$$

and the standard deviation is

$$\sum_{i=1}^{32} \|q_i^b - \dot{q}\| \Big/ 32 = 0.071.$$

Thus, the repeatability of the proposed algorithm based on the same set of TCP data is 0.071mm.

D. Comparisons

1) Different methods: The proposed method is different from the methods presented in [11]–[15] with different calibration targets and different calibration procedures (Table II). Balls had been used for the calibration between a CMM

TABLE I. THE SENSING DATA FOR DISK-BASED CALIBRATION.

	$i=1$	$i=2$	$i=3$	$i=4$
$T_{tcp,i}$	(-222.966,-295.994,360.786)	(-232.113,-298.165,344.196)	(-374.932,-347.138,338.836)	(-377.572,-373.611,352.241)
$R_{tcp,i}$ †	(3.057,-0.103,-2.008)	(3.115,-0.2,-2.269)	(-2.981,0.256,-2.491)	(3.092,0.208,3.11)
q^i	(5.947,0,5.97)	(-12.315,0,24.716)	(9.328,0,34.277)	(9.675,0,15.973)
D_i^i ‡	(0.042,0.336,-0.941)	(-0.146,0.424,-0.894)	(0.222,0.589,-0.777)	(-0.024,0.674,-0.738)
D_i^b ‡	(0.181,-0.246,-0.952)	(0.287,-0.294,-0.912)	(0.391,-0.593,-0.704)	(0.164,-0.631,-0.758)

† ZYX Euler angle.
‡ D_i^i and D_i^b are normalized since $\|D_i^i\| = \|D_i^b\|$.

TABLE II. COMPARISON OF DIFFERENT CALIBRATION METHODS.

	Calibration target	Data from a conjugate pair	Number of conjugate pairs	Method	Required data for each conjugate pair
[15]	Tetra-ball	$\left\{q^b, q^i\right\}$	More than 3	Constrained least-squares optimization	q^b and q^i
[11]–[14]	Ball	$\left\{T_{tcp,i}, R_{tcp,i}, q^i\right\}$	At least 7	At least 4 data for R_{sensor}, and at least 3 data for T_{sensor}	for R_{sensor}: one arc for T_{sensor}: m arcs
Ours	Planar disk	$\left\{T_{tcp,i}, R_{tcp,i}, q^i, D_i^i, D_i^b\right\}$	At least 4	$\left\{T_{tcp,i}, R_{tcp,i}, D_i^i, D_i^b\right\}$ for R_{sensor}, and $\left\{T_{tcp,i}, R_{tcp,i}, q^i\right\}$ for T_{sensor}	m chords

machine and a laser-stripe sensor [15]. The target is also to find out the transformation matrix

$$\mathrm{H} = \begin{bmatrix} R_{sensor} & T_{sensor} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & p_1 \\ t_{21} & t_{22} & t_{23} & p_2 \\ t_{31} & t_{32} & t_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Multiple conjugate pairs $\left\{q^b, q^i\right\}$ are collected and used to formulate a constrained least-squares optimization problem with augmented objective function

$$\begin{aligned} F = & \sum_{i=1}^{n} \left|q^b - H \cdot q^i\right|^2 + \sum_{j=1}^{3} \lambda_j \left[t_{j1}^2 + t_{j2}^2 + t_{j3}^2 - 1\right] \\ & + \lambda_4 \left[t_{12}t_{13} + t_{22}t_{23} + t_{32}t_{33}\right]. \end{aligned}$$

Solving such problem directly gives H. Theoretically, three conjugate pairs are sufficient. However, the constrained least-squares optimization problem may give more than one solution. Thus, more conjugate pairs are desired [15].

Another type of ball-based calibration methods [11]–[14] is different from that presented in [15]. First, the ball center q^b is fixed but is unknown, and data $\left\{T_{tcp,i}, R_{tcp,i}, q^i\right\}$ are collected satisfying

$$q^b = R_{tcp,i} R_{sensor} q^i + R_{tcp,i} T_{sensor} + T_{tcp,i}.$$

This is different from [15], where $\left\{q^b, q^i\right\}$ should be collected. R_{sensor} and T_{sensor} are solved separately. First, with TCP in translation mode, at least four data are collected and used to solve a set of linear equations for R_{sensor}. Then, as TCP moves to different positions with different orientations, at least three data are collected. Coupling with previously computed R_{sensor}, solving another set of linear equations leads to T_{sensor}.

Balls are preferred for the calibration, because the laser has a uniform view from any position and orientation [15]. In the proposed calibration algorithm, the position and orientation of the 2D planar disk do not affect the calibration. This enables the calibration target to be freely placed in the system like the balls. During calibration, R_{sensor} and T_{sensor} are also solved separately with at least four conjugate pairs in total. Different from previous methods [11]–[15], our method collects more data $\left\{T_{tcp,i}, R_{tcp,i}, q^i, D_i^i, D_i^b\right\}$ from one conjugate pair. First, data $\left\{T_{tcp,i}, R_{tcp,i}, D_i^i, D_i^b\right\}$ are used to formulate Eq.(45) to derive R_{sensor}. Then, Eq.(47) is formulated using R_{sensor} and data $\left\{T_{tcp,i}, R_{tcp,i}, q^i\right\}$ to obtain T_{sensor}. Sensing data are usually collected manually. Users are required to try different T_{tcp} and R_{tcp} to locate different conjugate pairs. Smaller number of conjugate pairs make the sensing data collection easier. The ball-based calibration methods [11]–[14] need to collect two sets of sensing data, which needs at least 7 conjugate pairs in total. The first four conjugate pairs used for solving rotation portion required at least four T_{tcp} with the same R_{tcp}. In the second set of data for solving the translation portion, two T_{tcp} with the same R_{tcp} are required for each conjugate pair. Compared with the ball-based calibration, the proposed algorithm only requires one set of the sensing data with at least 4 conjugate pairs in total.

2) Similar calibration accuracy: The ball in Fig.14(b) is used to illustrate that the proposed calibration method can achieve similar accuracy as the ball-based method does. In this comparison, the disk-based calibration and ball-based calibration are based on four and eight conjugate pairs, respectively. Table III lists the eight sensing data $\left\{T_{tcp,i}, R_{tcp,i}, q^i\right\}_{i=1}^{8}$ derived for the ball-based calibration, where the first four sensing data with the same orientation are used for solving R_{sensor} and the others with different orientations are used for deriving T_{sensor}. The ball-based calibration gives

$$\Upsilon = \begin{bmatrix} 0.99983 & -0.0115 & -0.01411 & 6.30066 \\ -0.01129 & -0.99983 & 0.01485 & 54.38647 \\ -0.01428 & -0.01468 & -0.99979 & 364.14357 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (53)$$

The deviation between Eq.(49) and Eq.(53) is

$$\Lambda^{-1}\Upsilon = \begin{bmatrix} 1.0000 & -0.0059 & -0.0070 & 0.2751 \\ 0.0059 & 0.9999 & -0.0119 & -0.2055 \\ 0.0071 & 0.0119 & 0.9999 & 0.1104 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (54)$$

TABLE III. THE SENSING DATA FOR BALL-BASED CALIBRATION.

	$T_{tcp,i}$	$R_{tcp,i}$ †	q^i
$i=1$	(-215.407,-287.69,559.998)	(3.08,-0.25,-2.611)	(-16.168,43.629,-58.079)
$i=2$	(-179.4,-289.3,550.402)	(3.08,-0.25,-2.611)	(15.42,23.498,-57.636)
$i=3$	(-187.799,-283.8,502.902)	(3.08,-0.25,-2.611)	(23.695,30.256,-10.06)
$i=4$	(-133.8,-268,564.099)	(3.08,-0.25,-2.611)	(60.036,19.653,-84.071)
$i=5$	(-317.98,-296.293,567.379)	(-3.074,-0.051,-2.689)	(-26.706,34.022,-52.304)
$i=6$	(-290.941,-319.017,496.149)	(3.116,-0.013,-3.027)	(12.521,45.942,16.945)
$i=7$	(-275.77,-326.73,480.747)	(3.09,-0.136,3.037)	(-8.488,54.055,29.974)
$i=8$	(-217.283,-327.968,569.943)	(2.994,-0.249,-2.467)	(-50.322,37.241,-64.803)

† ZYX Euler angle.

with Euler angles

$$(0.0119, -0.0071, 0.0059). \tag{55}$$

Following Section III-D, the ball center from each orientation is

$$\begin{bmatrix} \mu_i \\ 1 \end{bmatrix} = \begin{bmatrix} R_{tcp,i} & T_{tcp,i} \\ 0 & 1 \end{bmatrix} \Upsilon \begin{bmatrix} q^i \\ 1 \end{bmatrix}, i \in \{5,6,7,8\}.$$

The ball center is estimated as

$$\mu = \sum_{i=5}^{8} \mu_i \Big/ 4,$$

and the calibration accuracy is estimated as

$$\max_{i=5}^{8} \|\mu - \mu_i\| = 0.208mm. \tag{56}$$

Eq.(51) and Eq.(56) are similar. However, they are estimated according to different fixed points. In [23], calibration accuracy is evaluated as the standard deviation of the ball centers reconstructed from different scans. We follow the same way to compare the two calibration results Eq.(49) and Eq.(53). The scanning system scans the ball in Fig.14(b) with the robotic arm in different orientations. In each scan, 50 profiles are sampled with the robotic arm in a fixed orientation. For $i = 1, 2, \cdots, 8$, applying a transformation matrix (Eq.(49) or Eq.(53)) to the i-th scan gives one point cloud C_i. On average, there are 45000 points in each point cloud. The ninth point cloud is defined as $C_9 = \bigcup_{i=1}^{8} C_i$. A ball is fitted to the point cloud C_i to derive the ball center $\nu_i, i = 1, 2, \cdots, 9$ (Table IV). The standard deviation of the ball centers is calculated as

$$\sum_{i=1}^{8} \|\nu_i - \nu_9\| \Big/ 8.$$

The standard deviations for using Eq.(49) and Eq.(53) are $0.342mm$ and $0.355mm$, respectively. Thus, we conclude that the disk-based calibration can provide a similar calibration result as the ball-based method does.

3) Different noise-tolerance: The disk-based and ball-based calibrations are affected by different sources of noise. Disk-based calibration relies on the two tips on each chord, thus the x-resolution of the sensor affect the accuracy. One the other hand, the ball-based calibration depends on the arc data, which are affected by the z-accuracy of the sensor and the surface condition of the balls. The fitting algorithms adopted during the calibration may also introduce errors as well. Thus, instead of comparing their accuracies under the same noise level, we study their capability to tolerate a given noise. As presented in Fig. 15, increasing m is one way to tolerate the noise in disk-based calibration.

TABLE IV. RECONSTRUCTED BALL CENTERS.

	ν_i from Eq.(49)	ν_i from Eq.(53)
$i=1$	(-321.193,-303.203,150.141)	(-321.068,-302.984,150.372)
$i=2$	(-320.712,-302.869,150.7)	(-320.443,-302.813,150.796)
$i=3$	(-321.049,-302.838,150.485)	(-320.891,-302.673,150.707)
$i=4$	(-320.997,-302.84,150.181)	(-320.878,-302.691,150.583)
$i=5$	(-320.746,-302.522,150.15)	(-320.415,-302.855,150.755)
$i=6$	(-321.011,-302.605,150.254)	(-320.875,-302.618,150.433)
$i=7$	(-320.696,-302.671,150.477)	(-320.24,-302.909,150.836)
$i=8$	(-320.945,-303.067,150.352)	(-320.754,-302.99,150.688)
$i=9$	(-320.877,-302.896,150.311)	(-320.61,-302.785,150.665)

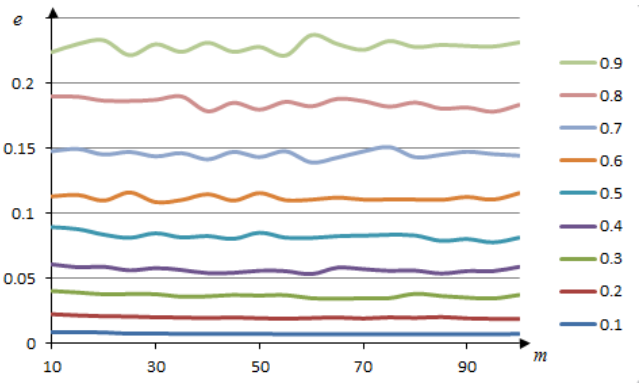


Fig. 17. Ball-based calibration accuracy with respect to σ and m (the horizontal coordinate is the number of arcs m, the vertical coordinate is the calibration accuracy $e(mm)$, and curves in different colors correspond to different noise level $\sigma(mm)$).

As presented in the last column of Table II, the ball-based calibration algorithms [11]–[14] collect one arc for each conjugate pair of R_{sensor}, and m arcs for each conjugate pair of T_{sensor}. Robotic data for deriving Table III are used to generate arcs. On each arc, points are generated in the sensor coordinate system with x-resolution of 0.1mm and Gaussian noise $G(\sigma)$ on the z value of each point. For each $m - \sigma$ pair, ball-based calibration is repeated for 500 times, and e is the standard deviation of the 500 ball centers reconstructed from the calibration. According to the simulation result in Fig. 17, in the same noise level σ, increasing m does not affect e, which is different from the disk-based calibration.

VI. Conclusion

For calibrating profile sensors to robotic arms, all existing methods require 3D calibration balls. Previously, 2D disks are not used for robot-sensor calibration because it is difficult to estimate its orientation during the calibration. In this paper, a new calibration algorithm is proposed to perform robot-sensor calibration based on a 2D disk. The novel feature is that, in the calibration procedures, the 2D disk can be placed in arbitrary orientation and this makes the placement of the disk as easy as that of a 3D ball. To acquire the sensing data for our calibration algorithm, users can project the line laser onto the 2D disk without passing any specific point. The proposed algorithm can derive the angular relationship between the 2D disk and the laser's plane, and automatically obtain the disk center for the calibration. A new two-step calibration is proposed based on a single set of sensing data. This can reduce not only the manual work in data collection but also the error propagation. Minimal manual work in data collection for the calibration is desired for better accuracy and faster processing. Simulations are performed to validate the proposed algorithm and analyze the effect of noise. Experiments show that the proposed algorithm can achieve similar accuracy as the ball-based method does.

Acknowledgements

This work is partially supported by SERC grant (1225100006) from A∗STAR of Singapore.

References

[1] I. Chen, S. Wang *et al.*, "An efficient approach for the calibration of multiple PTZ cameras," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 2, pp. 286–293, 2007.

[2] ——, "An efficient approach for dynamic calibration of multiple cameras," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 187–194, 2009.

[3] L. Wu and H. Ren, "Finding the kinematic base frame of a robot by hand-eye calibration using 3d position data," *IEEE Transactions on Automation Science and Engineering*, vol. PP, no. 99, pp. 1–11, 2016.

[4] 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.

[5] 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.

[6] H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX= YB," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549–554, 1994.

[7] S. Larsson and J. Kjellander, "Motion control and data capturing for laser scanning with an industrial robot," *Robotics and autonomous systems*, vol. 54, no. 6, pp. 453–460, 2006.

[8] F. Ernst, L. Richter, L. Matthäus, V. Martens, R. Bruder, A. Schlaefer, and A. Schweikard, "Non-orthogonal tool/flange and robot/world calibration," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 8, no. 4, pp. 407–420, 2012.

[9] W. L. Li, H. Xie, G. Zhang, S. J. Yan, and Z. P. Yin, "Hand-eye calibration in visually-guided robot grinding," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–9, 2015.

[10] S. Yin, Y. Guo, Y. Ren, J. Zhu, S. Yang, and S. Ye, "Real-time thermal error compensation method for robotic visual inspection system," *The International Journal of Advanced Manufacturing Technology*, vol. 75, no. 5, pp. 933–946, 2014.

[11] S. Zhu, P. Cui, C. Shen, and Y. Guo, "Scanner external calibration algorithm based on fixed point in robot remanufacturing system," *Journal of Central South University of Technology*, vol. 12, no. 2, pp. 133–137, 2005.

[12] J. Li, M. Chen, X. Jin, Y. Chen, Z. Dai, Z. Ou, and Q. Tang, "Calibration of a multiple axes 3-D laser scanning system consisting of robot, portable laser scanner and turntable," *Optik-International Journal for Light and Electron Optics*, vol. 122, no. 4, pp. 324–329, 2011.

[13] Y. Ren, S. Yin, and J. Zhu, "Calibration technology in application of robot-laser scanning system," *Optical Engineering*, vol. 51, no. 11, pp. 114 204–114 204, 2012.

[14] S. Yin, Y. Guo, Y. Ren, J. Zhu, S. Yang, and S. Ye, "A novel TCF calibration method for robotic visual measurement system," *Optik-International Journal for Light and Electron Optics*, vol. 23, no. 125, pp. 6920–6925, 2014.

[15] C. Che and J. Ni, "A ball-target-based extrinsic calibration technique for high-accuracy 3-D metrology using off-the-shelf laser-stripe sensors," *Precision Engineering*, vol. 24, no. 3, pp. 210–219, 2000.

[16] S. Yin, Y. Ren, J. Zhu, S. Yang, and S. Ye, "A vision-based self-calibration method for robotic visual inspection systems," *Sensors*, vol. 13, no. 12, pp. 16 565–16 582, 2013.

[17] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.

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

[19] H. Zuang 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.

[20] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, no. 2, pp. 97–111, 2007.

[21] M. Pilu, A. W. Fitzgibbon, and R. B. Fisher, "Ellipse-specific direct least-square fitting," in *Proceedings of International Conference on Image Processing*, vol. 3. IEEE, 1996, pp. 599–602.

[22] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[23] H. Xie, C. Pang, W. Li, Y. Li, and Z. Yin, "Hand-eye calibration and its accuracy analysis in robotic grinding," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 862–867.

[24] "LMI GOCATOR 2340," http://www.lmi3d.com/sites/default/files/products/gocator/datasheets/DATASHEET_Gocator_2300_WEB_EN.pdf, accessed: 2016-10-05.

[25] "ABB IRB 120," http://new.abb.com/products/robotics/industrial-robots/irb-120, accessed: 2016-10-05.