Robotic Hand-eye Calibration with Depth Camera: A Sphere Model Approach

Lixin Yang[1], Qixin Cao[2], Minjie Lin[3], Haoruo Zhang[4], Zhuoming Ma[5]
School of Mechanical Engineering
Shanghai Jiao Tong University
Shanghai, People's Republic of China
e-mail: {siriusyang, qxcao, linminjie180, zhr, mzmsjtu2016}@sjtu.edu.cn

***Abstract*—This paper presents a new depth-based hand-eye calibration method to find the transformation between depth camera and robot wrist. Owing to the true hand-eye transformation is hard to obtained, indirect measurements depending on robot movements and calibration tools is adopted. In our method, A sphere, which has simple and elegant parameterized presentation, is used as calibration tool. The points cloud acquired from the depth camera viewing the calibration tool are aligned to a virtual sphere model by RANSAC to find the translation part of the homogeneous transformation between depth camera and calibration tool. The rotational part of the transformation is ambiguous thus eliminated by reformulating the hand-eye calibration problem. Then a non-linear optimization method is proposed to solve the hand-eye rotation and translation simultaneously. Due to the ground truth hand-eye transformation is not available, a quantitative error measurement is adopted to measure the accuracy.**

Keywords-hand-eye calibration; geometric alignment; depth cameras; non-linear optimization

I. INTRODUCTION

When cameras are mounted on robots, robotics hand-eye calibration address the task of computation the relative orientation and position between the robot's base/wrist coordinate system and the camera optical coordinate system. Based on the camera's mounting position, the hand-eye calibration setups can be categorized as: *eye-in-hand* and *eye-to-hand*. The first one specified that the camera are mounted on the robot's wrist (last link of arm), while the other specified the mounting position are independent to arm's position (Fig. 1). Owing to in both categories cameras are rigidly mounted, the relative orientation and position between any two of frames can be addressed as homogeneous transformation matrix in 3D Euclidean space.

For 2D color camera, well researched tasks have been proposed to tackle hand-eye calibration [1][2][3]. Hand-eye calibration problem has been formulated as homogeneous equation with the form:

$$\boldsymbol{AX} = \boldsymbol{XB} \tag{1}$$

where $\boldsymbol{X}$ is the unknown hand-eye transformation, $\boldsymbol{A}$ the robot hand movement. $\boldsymbol{B}$ is the corresponding camera movement (*eye-in-hand*) or calibration tool movement (*eye-to-hand*). $\boldsymbol{A}$ can be evaluated by encoder readings of robot's joints. And $\boldsymbol{B}$, referred as camera extrinsic, usually obtained by using camera viewing patterned calibration tools (chessboard).

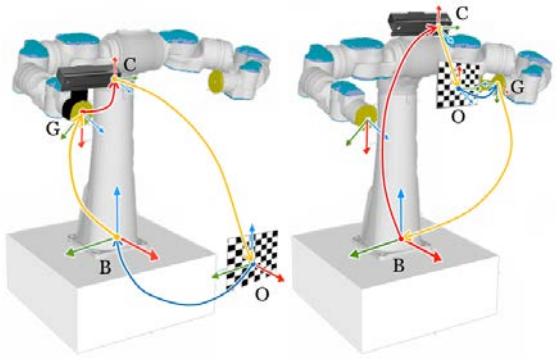

Figure 1. The eye-in-hand and eye-to-hand calibtraion setups.

Recently, the technological advancement brought up a variety of 3D cameras like Kinect, RealSense and SwissRanger, all enable capturing points cloud in real time. While this advancement enables 3D points-based robot bin-picking, SLAM (simultaneously localization and mapping) and 3D reconstruction, it also brings up the question that how to calibrate depth sensor mounted on the robots. Owing to majority of the depth sensors stream depth intensity image and color image simultaneously. A straight-forward way is to use the 2D image-based algorithm [4][5]. However, for situations that depth and color frame are not accurately aligned, or the resolutions of color image are lower than depth intensity image, and for those cameras that only have depth sensor, a color-independent hand-eye calibration method is necessary. Fuchs [6] proposed a method used depth measurement of a calibration plane to restore relative orientation and position. Khan [7] addressed 3D hand-eye calibration by geometrically aligning 3D measurements with a virtual model using iterative closest point algorithm (ICP) [8]. Khan's method is similar to the image-based approach. The difference is that, $\boldsymbol{B}$ in (1) is evaluated by tracking the color frame for image-based approach, while by aligning 3D point for Khan's geometric approach.

When using geometric approach to solve hand-eye calibration, several requirements should be satisfied. First, while using depth camera to measure calibration tool's position and orientation, the measurements accuracy is poor at jump edges and vertex [9], to avoid this drawback, a sophisticated artificial object with curved surface need to be

978-1-5386-6338-7/18/$31.00 ©2018 IEEE

used. Second, when using ICP to track camera motions, the shape of calibration tools is required to be neither symmetries nor periodicities [7]. Both requirements impose the geometrical complexity on tools. Next, depth camera only has a limited view, and the view vary from one to another with different camera position. Furthermore, depth camera can't have a *priori* to tell that which points in the points cloud are belonged to the model. All measurement points will be used by ICP. In both situations, without giving ICP a proper initial value, misalignment is possible.

This paper contributes to finding a more elegant way to tackle depth-based hand-eye calibration problem with three main contributions. First, we propose using sphere model as calibration tool. Among all the 3D geometry model, a sphere has the simplest parameterized expression: x, y, z and radius. Another advantage of using sphere model is that the measurements on the surface of sphere are invariant in different camera views, meaning whichever orientation the observation is made, the shape of measurement remains unchanged. Instead of using ICP, we adopt random sample consensus algorithm (RANSAC) [10] to calculate the parameters of the sphere model inside camera's view. When a sphere model is given as ground truth, RANSAC can find the best-matched batch of points (inliers) that fit the model in the whole points cloud, leaving the remained points (outliers) not involved in calculation. Second, we propose a new homogeneous formulation for the depth-based hand-eye calibration problem and simultaneously solve unknowns in non-linear optimization. Third, we provide a depth-based error criterion and use this criterion to quantitatively evaluate the solution.

This paper is structured as follows. The problem formulation will be given in Section 2, including an introduction of how the previous hand-eye calibration model were formulated and solved. Section 3 constructs non-linear optimization model to the problem and describes its quaternions-based representation. Error matrix are also introduced to evaluate hand-eye calibration. Numerical simulation is performed in Section 4 which also illustrates the noise sensitivity. In Section 5, experiment results, accuracy evaluation and comparation with color-based hand-eye calibration are presented. This paper concludes in Section 6.

II. PROBLEM FOMULATION

As in Fig. 2, the relative positions and orientations among the robot base coordinates, gripper, camera, and calibration tool can be presented as a closed loop. The vertex **B**, **G** {1, 2…}, **C** {1, 2…}, **O** are the origin of the reference frame of the base, gripper, camera, and the calibration tool, respectively. Those directed edges illustrate the homogeneous transformations between the two frames' coordinate system. For instance, edge $\boldsymbol{T}_{GC}$ stand for relative transformation of the frame **C** in respect of frame **G**. In mathematical, it is obvious that if three edges in the quadrangle ***BGCO*** is specified, the other edge is analytically solved by closing the loop:

$$\boldsymbol{T}_{BG}\boldsymbol{T}_{GC}\boldsymbol{T}_{CO}\boldsymbol{T}_{OB} = \boldsymbol{I} \tag{2}$$

With two unknowns $\boldsymbol{T}_{GC}$ and $\boldsymbol{T}_{OB}$ existed in (2), the classical hand-eye solution considered eliminating $\boldsymbol{T}_{OB}$. The problem is reformulated by closing the middle loop $\boldsymbol{G_1G_2C_1C_2}$ as:

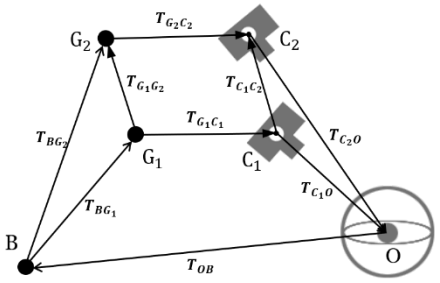

Figure 2. Relative coordinate transformations of hand-eye calibration.

$$\boldsymbol{T}_{G_1G_2}\boldsymbol{T}_{G_2C_2} = \boldsymbol{T}_{G_1C_1}\boldsymbol{T}_{C_1C_2} \tag{3}$$

The camera is rigidly mounted on the gripper, thus:

$$\boldsymbol{T}_{G_2C_2} = \boldsymbol{T}_{G_1C_1} \tag{4}$$

Equation (3) represent the classical hand-eye calibration formulation: $\boldsymbol{AX} = \boldsymbol{XB}$. Tsai and Shiu [1][11] proved both rotational and translation part of $\boldsymbol{X}$ have one degree of freedom, so to yield $\boldsymbol{X}$ uniquely, at least two robot movements (three positions) should be performed

At the early stage of hand-eye calibration, solving (3) is equivalent to solving two separate equations by decoupling rotation and translation apart (*R then t*). Several representations of rotation are adopted: Euler axis-angle parameters by Tsai & Lenz [1] and Shiu & Ahmad [11]; unit quaternion by Chou & Kamel [12]; Lie algebra in *SO3* by Park & Martin [13]. Owing to noises inevitably dwell among those $\boldsymbol{A}$ and $\boldsymbol{B}$. A more practical approach is preforming several robot movements and estimating $\boldsymbol{X}$ by adopting least-square or non-linear optimization method [14][15].

However, '*R then t*' approach augments the translation error by passing the rotation error in to it. Chen [16] suggested a *'R and t*' method that solves rotation and translation simultaneously should be adopted. Lu & Chou [17] introduced quaternions based eight-space formulation to linearly optimize *R* and *t*. Horaud & Dornaika [15] proposed a non-linear optimization to the solution. Daniilidis [18] parameterize transformation as dual-quaternions. Strobl & Hirzinger [19] represented transformations on the group of *SE3*. In order to improve the accuracy, methods for selecting well-suited batches from whole set of hand-eye movements were presented by Schmidt & Niemann [20].

In our depth-based calibration method, A sphere model are adopted to obtain camera extrinsic. However, sphere model, with the most elegant geometry representation, inevitably losses the uniqueness on orientation. When using RANSAC to restore the expression of sphere from camera's measurement, the outputs are the sphere's position and radius with respect to camera coordinates frame. Lacking observation on rotation, the homogeneous transformation

$\boldsymbol{T_{CO}}$ in Fig. 2 only guarantees the correctness on relative translation. Thus, the classic mathematical model $\boldsymbol{AX=XB}$ cannot be formulated to solve for $\boldsymbol{X}$.

Then the hand-eye calibration is reformulated by closing the original loop $\boldsymbol{G_1G_2C_2C_1}$ in Fig. 2. Each edge in this loop stands for a homogeneous transformation matrix. The equation can be rewritten as:

$$\boldsymbol{T_{C_1G_1}T_{G_1B}T_{BO}} = \boldsymbol{T_{C_1O}} \tag{5}$$

the subscript shift represents inverse matrix. Let's rename $\boldsymbol{T_{C_1G_1}}$ by unknow $\boldsymbol{T_X}$, $\boldsymbol{T_{G_1B}}$ by $\boldsymbol{T_i}$, $\boldsymbol{T_{BO}}$ by $\boldsymbol{T_b}$, and $\boldsymbol{T_{C_1O}}$ by $\boldsymbol{T_j}$ for continence. The Equation # can be expanded as:

$$\begin{pmatrix} \boldsymbol{R_x} & \boldsymbol{t_x} \\ \boldsymbol{0} & 1 \end{pmatrix}\begin{pmatrix} \boldsymbol{R_i} & \boldsymbol{t_i} \\ \boldsymbol{0} & 1 \end{pmatrix}\begin{pmatrix} \boldsymbol{R_b} & \boldsymbol{t_b} \\ \boldsymbol{0} & 1 \end{pmatrix}=\begin{pmatrix} \boldsymbol{R_j} & \boldsymbol{t_j} \\ 0 & 1 \end{pmatrix} \tag{6}$$

where $\boldsymbol{R}$ is a 3 × 3 orthogonal matrix representing a rotation, and $\boldsymbol{t}$ is 3 by 1 vector representing a translation. Merging the left three matrices into one, the two sub-equations of first row are:

$$\boldsymbol{R_xR_iR_b} = \boldsymbol{R_j} \tag{7}$$

$$\boldsymbol{R_xR_it_b} + \boldsymbol{R_xt_i} + \boldsymbol{t_x} = \boldsymbol{t_j} \tag{8}$$

Owing to RANSAC of a sphere model can only tell translation, while leaving $\boldsymbol{R_j}$ uncertain, only the second equation is adopted to solve $\boldsymbol{R_x}$, $\boldsymbol{t_x}$ and $\boldsymbol{t_b}$.

III. Quaternion-based Non-linear Optimization

In the case of *n* robot motions, solving (8) is casted into minimizing a non-linear error function. For each robot position, an error term can be written as:

$$e^{(k)} = \left\| \left(\boldsymbol{R_x R_i}^{(k)} \boldsymbol{t_b} + \boldsymbol{R_x t_i}^{(k)} + \boldsymbol{t_x} \right) - \boldsymbol{t_j}^{(k)} \right\|^2 \tag{9}$$

where superscript *k* stands for the *k-th* robot position.

While $\boldsymbol{R}$ stands for rotation in 3D space, the 9 components only describe 3 degrees of rotational freedom. However, when using 3 elements (Euler angles) to represent rotation, it may encounter the Gimbal lock problem. Unit quaternions can eliminate those geometrical uncertainty. Quaternion algebra provide a unique and elegant description for orientation in 3D space [12][21]. A quaternion is defined by three imaginary parts and one real part:

$$q = q_0 + iq_x + jq_y + kq_z \tag{10}$$

and can be viewed as a 4 by 1 vector:

$$\boldsymbol{q} = \left[q_0, q_x, q_y, q_z \right]^T = \left[q_0, \boldsymbol{q_{im}}^T \right]^T \tag{11}$$

The multiplication of two quaternion $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ is given by:

$$\boldsymbol{q_1} * \boldsymbol{q_2} = \boldsymbol{Q(q_1)q_2} = \boldsymbol{W(q_2)q_1} \tag{12}$$

with:

$$\boldsymbol{Q(q)} = \begin{pmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & -q_z & q_y \\ q_y & q_z & q_0 & -q_x \\ q_z & -q_y & q_x & q_0 \end{pmatrix}; \boldsymbol{W(q)} = \begin{pmatrix} q_0 & -q_x & -q_y & -q_z \\ q_x & q_0 & q_z & -q_y \\ q_y & -q_z & q_0 & q_x \\ q_z & q_y & -q_x & q_0 \end{pmatrix} \tag{13}$$

The 3 × 3 orientation or rotation matrix $\boldsymbol{R}$ is equivalent to a quaternion $\boldsymbol{q}$ if:

$$q_0 = \cos\left(\frac{\theta}{2}\right); \boldsymbol{q_{im}} = \boldsymbol{n}\sin\left(\frac{\theta}{2}\right) \tag{14}$$

where $\boldsymbol{n}$ and θ are $\boldsymbol{R}$'s Euler axis-angle. This impose the constraint on quaternion-based rotation representation $\boldsymbol{q}$:

$$\boldsymbol{q} * \overline{\boldsymbol{q}} = \|\boldsymbol{q}\|^2 \triangleq 1 \tag{15}$$

The 3 × 1 position or translation vector $\boldsymbol{t}$ can be viewed as a purely imaginary quaternion:

$$\boldsymbol{t}_{(quaternion)} = \left[0, \boldsymbol{t}^T \right]^T = \left[0, t_x, t_y, t_z \right]^T \tag{16}$$

Then quaternion-based expression of '*rotating t by pre-multiplication of R'* can be written as:

$$\boldsymbol{Rt} = \boldsymbol{q} * \begin{pmatrix} 0 \\ \boldsymbol{t} \end{pmatrix} * \overline{\boldsymbol{q}} \tag{17}$$

In the following parts, we will still write $(0, \boldsymbol{t}^T)^T$ as $\boldsymbol{t}$ yet implicitly transfer $\boldsymbol{t}$ to purely imaginary quaternion. Then (9) can be written as:

$$e^{(k)} = \left\| \boldsymbol{q_x} * \left(\boldsymbol{q_i}^{(k)} * \boldsymbol{t_b} * \overline{\boldsymbol{q}}_{\boldsymbol{i}}^{(k)} \right) * \overline{\boldsymbol{q}}_{\boldsymbol{x}} + \boldsymbol{q_x} * \boldsymbol{t_i}^{(k)} * \overline{\boldsymbol{q}}_{\boldsymbol{x}} + \boldsymbol{t_x} - \boldsymbol{t_j}^{(k)} \right\|^2 \tag{18}$$

owing to $\boldsymbol{q_x}$ is unit quaternion:

$$\begin{aligned} e^{(k)} &= \left\| \boldsymbol{q_x} * \left(\boldsymbol{q_i}^{(k)} * \boldsymbol{t_b} * \overline{\boldsymbol{q}}_{\boldsymbol{i}}^{(k)} \right) * \overline{\boldsymbol{q}}_{\boldsymbol{x}} + \boldsymbol{q_x} * \boldsymbol{t_i}^{(k)} * \overline{\boldsymbol{q}}_{\boldsymbol{x}} + \boldsymbol{t_x} - \boldsymbol{t_j}^{(k)} \right\|^2 \|\boldsymbol{q_x}\|^2 \\ &= \left\| \boldsymbol{q_x} * \boldsymbol{q_i}^{(k)} * \boldsymbol{t_b} * \overline{\boldsymbol{q}}_{\boldsymbol{i}}^{(k)} + \boldsymbol{q_x} * \boldsymbol{t_i}^{(k)} + \boldsymbol{t_x} * \boldsymbol{q_x} - \boldsymbol{t_j}^{(k)} * \boldsymbol{q_x} \right\|^2 \end{aligned} \tag{19}$$

The expression of (19) can be derived from the rules described in (13):

$$\begin{aligned} e^{(k)} &= \left\| \boldsymbol{Q}(\boldsymbol{q_x}) \boldsymbol{W}\left(\boldsymbol{q_i}^{(k)} \right)^T \boldsymbol{Q}\left(\boldsymbol{q_i}^{(k)} \right) \boldsymbol{t_b} + \boldsymbol{W}\left(\boldsymbol{t_i}^{(k)} \right) \boldsymbol{q_x} + \boldsymbol{W}(\boldsymbol{q_x}) \boldsymbol{t_x} - \boldsymbol{Q}\left(\boldsymbol{t_j}^{(k)} \right) \boldsymbol{q_x} \right\|^2 \\ &= \left\| \boldsymbol{D}^{(k)} \boldsymbol{q_x} + \boldsymbol{W}(\boldsymbol{q_x}) \boldsymbol{t_x} + \boldsymbol{Q}(\boldsymbol{q_x}) \boldsymbol{C}^{(k)} \boldsymbol{t_b} \right\|^2 \end{aligned} \tag{20}$$

where the $\boldsymbol{D}^{(k)}$ and $\boldsymbol{C}^{(k)}$ are coefficient matrices:

$$\boldsymbol{D}^{(k)} = \boldsymbol{W}\left(\boldsymbol{t_i}^{(k)} \right) - \boldsymbol{Q}\left(\boldsymbol{t_j}^{(k)} \right) \tag{21}$$

$$\boldsymbol{C}^{(k)} = \boldsymbol{W}\left(\boldsymbol{q_i}^{(k)} \right)^T \boldsymbol{Q}\left(\boldsymbol{q_i}^{(k)} \right) \tag{22}$$

Let:

$$\boldsymbol{f}^{(k)} = \boldsymbol{D}^{(k)} \boldsymbol{q_x} + \boldsymbol{W}(\boldsymbol{q_x}) \boldsymbol{t_x} + \boldsymbol{Q}(\boldsymbol{q_x}) \boldsymbol{C}^{(k)} \boldsymbol{t_b} \tag{23}$$

$\boldsymbol{f}^{(k)}$ is a 4 × 1 error vector:

$$\boldsymbol{f}^{(k)} = \left(f_1^{(k)}, f_2^{(k)}, f_3^{(k)}, f_4^{(k)} \right)^T \tag{24}$$

Each element in $\boldsymbol{f}^{(k)}$ is a function of ten parameters: q_{xw}, q_{xx}, q_{xy}, q_{xz}, t_{xx}, t_{xy}, t_{xz}, t_{bx}, t_{by} and t_{bz}. The error term is:

$$e^{(k)} = \boldsymbol{f}^{(k)T} \boldsymbol{f}^{(k)} \tag{25}$$

For n robot positions, n error vectors can be vectorized as a $4n \times 1$ vector $\boldsymbol{f}$:

$$\boldsymbol{f} = \left(\boldsymbol{f}^{(1)T}, \boldsymbol{f}^{(2)T}, ..., \boldsymbol{f}^{(n)T} \right)^T \tag{26}$$

Then the total error function is:

$$E = \sum_{k=1}^{n} e^{(k)} = \sum_{k=1}^{n} \boldsymbol{f}^{(k)T} \boldsymbol{f}^{(k)} = \left(\boldsymbol{f}^{(1)T}, \boldsymbol{f}^{(2)T}, ..., \boldsymbol{f}^{(n)T} \right) \begin{pmatrix} \boldsymbol{f}^{(1)} \\ \boldsymbol{f}^{(2)} \\ \vdots \\ \boldsymbol{f}^{(n)} \end{pmatrix} = \boldsymbol{f}^T \boldsymbol{f} \tag{27}$$

Owing to E must be minimized under the constrain in (15). a penalty function is added at the end of error function:

$$E = \sum_{k=1}^{n} e^{(k)} + \lambda (1 - \boldsymbol{q_x}^T \boldsymbol{q_x})^2 \tag{28}$$

To ensure that $\boldsymbol{q_x}$ is in the form of unit quaternion, the high value of weight λ is required. In this paper, λ is set to be 10^6. The total error term can still take the form of $\boldsymbol{f}^T\boldsymbol{f}$ by rewriting $\boldsymbol{f}$ as a $(4n+1) \times 1$ vector:

$$\boldsymbol{f} = \left(\boldsymbol{f}^{(1)T}, ..., \boldsymbol{f}^{(n)T}, p \right)^T \tag{29}$$

where p is the square root of penalty function.

Then the problem of solving rotation and translation simultaneously based on n robot's positions can be stated as a non-linear minimization problem:

$$\min_{\boldsymbol{q_x}, \boldsymbol{t_x}, \boldsymbol{t_b}} E = \min_{\boldsymbol{q_x}, \boldsymbol{t_x}, \boldsymbol{t_b}} \left(\sum_{k=1}^{n} e^{(k)} + 10^6 (1 - \boldsymbol{q_x}^T \boldsymbol{q_x})^2 \right) = \min_{\boldsymbol{q_x}, \boldsymbol{t_x}, \boldsymbol{t_b}} \| \boldsymbol{f} \|^2 \tag{30}$$

Series of minimization methods were designed to tackle the non-linear minimization problems [22]. The core ideas among them are similar as simplifying the problem in a confidence interval by Tayler expansion (first and second order Newton method, Levenberg-Marquardt method). In this paper, we choose Levenberg-Marquardt method to minimize the problem.

To preform effective iterations, one has to specify the analytic form of Jacobian (first order derivative) matrix. The Jacobian Matrix can be expressed as:

$$\boldsymbol{J} = \begin{pmatrix} \frac{\partial \boldsymbol{f}^{(1)}}{\partial \boldsymbol{q_x}} & \frac{\partial \boldsymbol{f}^{(1)}}{\partial \boldsymbol{t_x}} & \frac{\partial \boldsymbol{f}^{(1)}}{\partial \boldsymbol{t_b}} \\ \vdots & \vdots & \vdots \\ \frac{\partial \boldsymbol{f}^{(n)}}{\partial \boldsymbol{q_x}} & \frac{\partial \boldsymbol{f}^{(n)}}{\partial \boldsymbol{t_x}} & \frac{\partial \boldsymbol{f}^{(n)}}{\partial \boldsymbol{t_b}} \\ \frac{\partial p}{\partial \boldsymbol{q_x}} & \frac{\partial p}{\partial \boldsymbol{t_x}} & \frac{\partial p}{\partial \boldsymbol{t_b}} \end{pmatrix}_{(4n+1) \times 10} \tag{31}$$

Each $\boldsymbol{f}^{(k)}$ in $\boldsymbol{f}$ has same form of expression yet different coefficients. For $\boldsymbol{f}^{(k)}$:

$$\frac{\partial \boldsymbol{f}^{(k)}}{\partial \boldsymbol{q_x}} = \begin{pmatrix} \frac{\partial f_1^{(k)}}{\partial q_{xw}} & \cdots & \frac{\partial f_1^{(k)}}{\partial q_{xz}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4^{(n)}}{\partial q_{xw}} & \cdots & \frac{\partial f_4^{(n)}}{\partial q_{xz}} \end{pmatrix}_{4 \times 4} \tag{32}$$

$$\frac{\partial \boldsymbol{f}^{(k)}}{\partial \boldsymbol{t_x}} = \begin{pmatrix} \frac{\partial f_1^{(k)}}{\partial t_{xx}} & \cdots & \frac{\partial f_1^{(k)}}{\partial t_{xz}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4^{(k)}}{\partial t_{xx}} & \cdots & \frac{\partial f_4^{(k)}}{\partial t_{xz}} \end{pmatrix}_{4 \times 3} \tag{33}$$

and $\partial \boldsymbol{f}^{(k)} / \partial \boldsymbol{t}_b$ has the same form as $\partial \boldsymbol{f}^{(k)} / \partial \boldsymbol{t}_x$.

Each $f_m^{(k)}$ (m=1,2,3,4) in $\boldsymbol{f}^{(k)}$ is a scalar represented by accumulation of several polynomials containing the ten unknowns. Calculating the derivative of each $f_m^{(k)}$ in respect of each unknown is straightforward but trivial. The detailed Jacobian matrix will not be presented in this paper.

IV. NUMERICAL SIMULATION

To test the accuracy and robustness of our method, a numerical simulation of depth-based hand-eye calibration has been made through artificial datasets. We firstly assigned the predefined values to hand-eye parameters: $\boldsymbol{q_x}$, $\boldsymbol{t_x}$ and $\boldsymbol{t_b}$. These values are viewed as ground truth. Second, we generated multiple simulated robot's hand positions: $\boldsymbol{q_i}$ and $\boldsymbol{t_i}$. And then we used those predefined parameters and hand positions to calculate the corresponding translation part of camera extrinsic $\boldsymbol{t_j}$ based on (8). Third, the parameters $\boldsymbol{q_x}$, $\boldsymbol{t_x}$ and $\boldsymbol{t_b}$ were reconsidered as unknows and estimated by optimizing the error function (30) using the artificial datasets: $\boldsymbol{q_i}$, $\boldsymbol{t_i}$ and $\boldsymbol{t_j}$. Finally, the error between estimated value and ground truth were evaluated. In real hand-eye calibration experiments, noise is composed mainly by two parts: one is the noise from robots movements and another is from the captured points cloud. During the simulation procedure, gaussian noise were manually added to the artificial datasets to simulate the real experiment condition and to test the noise sensitivity.

Assuming the actual mounting position of the camera on robot's hand are represented as rotating 15 degrees along z axis and translating 0.15, 0.15, 0.1 meter on x, y and z directions, respectively, the quaternion representation of rotation and translation are as:

$$\boldsymbol{q_{xactual}} = \begin{bmatrix} 0.991445 & 0 & 0 & 0.130526 \end{bmatrix}^T \tag{34}$$

$$\boldsymbol{t_{xactual}} = \begin{bmatrix} 0 & 0.15 & 0.15 & 0.1 \end{bmatrix}^T \tag{35}$$

and the translation of calibration tools with respect to robot base is given by:

$$\boldsymbol{t_{bactual}} = \begin{bmatrix} 0 & 1.0 & 1.0 & 1.5 \end{bmatrix}^T \tag{36}$$

Next, each artificial robot's hand position is generated as follow:

1) Choosing a set number of equally distributed rotation axis.
2) Rotating each position along one of these axis by a randomly selected angle from 10 to 45 degrees.
3) Translating each position by randomly selected distance from -0.2 to 0.2 meter along *x, y* and *z* axis respectively.

Then the translation part of camera extrinsic can be calculated. The Gaussian noises are added separately on robot's hand rotation part, translation part, and on camera extrinsic's translation parts. For rotation parts, the rotation is firstly represented as Euler axis-angle. Then the gaussian noises are added on each one of the three components. For the remaining two translation parts, the Gaussian noises are added on all *x, y* and *z* components. For each gaussian standard deviation of each perturbed part, errors are calculated between the estimated parameter: $\boldsymbol{q}_{xestim}$, $\boldsymbol{t}_{xestim}$ and predefined parameters: $\boldsymbol{q}_{xactual}$, $\boldsymbol{t}_{xactual}$ respectively.

The error between $\boldsymbol{t}_{xestim}$ and $\boldsymbol{t}_{xactual}$ can be calculated by their Euclidean distance. And the error between $\boldsymbol{q}_{xestim}$ and $\boldsymbol{q}_{xactual}$ can be represented as the norm of the Euler axis-angle of $\boldsymbol{R}_{xestim}^{-1}\boldsymbol{R}_{xactual}$, where $\boldsymbol{R}_x$ is the rotation matrix of $\boldsymbol{q}_x$. Fig. 3(a) shows the noise sensitivity of $\boldsymbol{R}_x$ and $\boldsymbol{t}_x$ under translational perturbations of $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ respectively with different standard deviations (σ). The components of $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ are perturbed by adding gaussian noise with zero mean and standard deviations between 0 and 0.01 m. For both conditions of perturbing $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$, the rotation error increases 0.00025 rad per 0.001 m sigma increase, and the translation error increases 0.0005 m per 0.001 m sigma increase. Similarly, Fig. 3(b) shows the noise sensitivity of $\boldsymbol{R}_x$ and $\boldsymbol{t}_x$ under rotational perturbations of $\boldsymbol{R}_i$. Each component of Euler axis-angle representation of $\boldsymbol{R}_i$ is perturbed by zero mean and standard deviations between 0 and 0.01 rad. The rotation error increases 0.0004 rad per 0.001 rad sigma increase and the translation error increases 0.00025 m per 0.001 rad sigma increase.

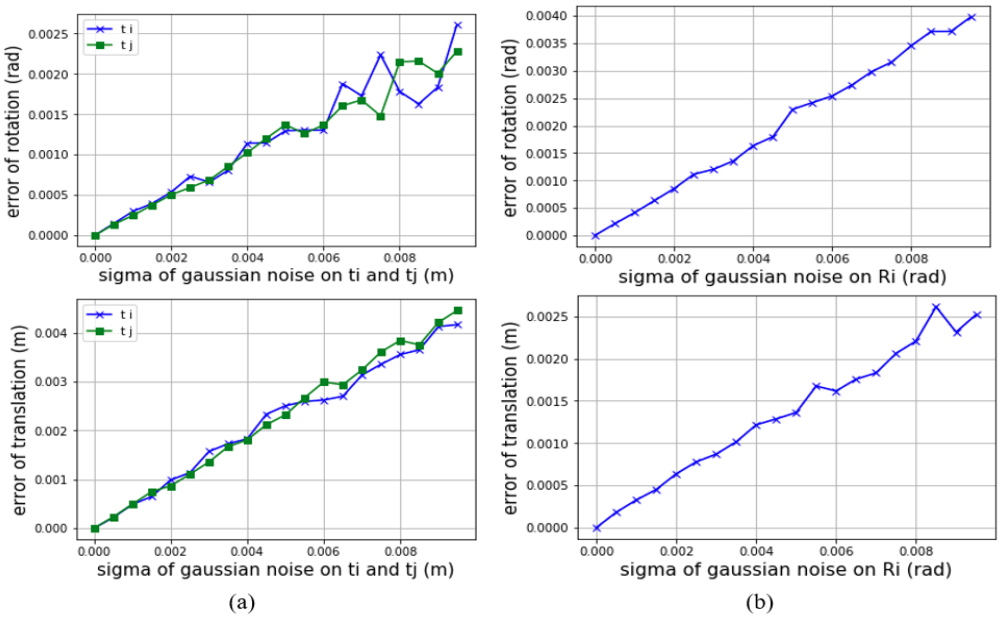

Figure 3. Noise sensitivity of $\boldsymbol{R}_x$ and $\boldsymbol{t}_x$: (a) under translational perturbations of $\boldsymbol{t}_i$ or $\boldsymbol{t}_j$. (b) under rotational perturbations of $\boldsymbol{R}_i$.

V. Experiment Result

A. Experiment Setups

The real experiments were conducted with Intel RealSense SR300 camera mounted on the wrist of the Motoman SDA10F robot's left arm as shown in Fig. 4. SR300 camera can align depth and color image stream in same camera optical frame. The calibration tools are fixed at approximate distance of 1.0, 1.0 and 0.5 meter away from robot's base frame. The experiments are divided to two sets. First sets used chessboard and color image to estimate hand-eye relationship by color-based method in. Second sets used sphere model and points cloud extract from depth density image to estimate the relationship by our method. In depth-based experiment set, two independent experiments are conducted. The data acquired from the first experiment were used to solve the non-liner optimization problems, while those from the second experiment are used to quantitatively evaluate the optimization result. To uniquely locate the center of sphere in the camera's optical frame, one can allocate a sphere model with unknown parameters to fit the points cloud by RANSAC. RANSAC returns the parameterized sphere model with most points as inliers and less accumulated error. The optimization problem requires an initial value of the hand-eye parameters. The translation parts from camera to hand and from robot base to sphere center can be coarsely measured, and the quaternion-based rotation part from camera to hand can be set as unit pure real quaternion. During each calibration experiment, the robotic

hand was moved to 45 locations along with 45 different camera positions. With the limitation of the robotic working volume and camera's capturing range, the 45 locations are manually planned to satisfy.

B. *Quantitative Evaluation and Comparation*

Since the ground truth hand-eye transformation is unaccessiable, we evaluate the performance of hand-eye calibraiton on the data from second experiment by firstly projecting the sphere model from camera optical frame into the robot base frame, and then calculating the Euclidian distance between the center of the projected sphere and of the estimated one (t_b). The projection can be mathmatically represented as:

$$p_{proj} = T_i T_x p_{cam} \tag{37}$$

where T_i, T_x stand for the transformation from robot hand to base and from camera to hand respectively, and p_{cam}, p_{proj} are the homogeneous representation of 3D points in camera and in robot base frame. Then the distance (error) between the projected sphere center p_{proj} and the estimated center, both in robot base frame, were calculated. In comparsion with the color-based calibraito n result, the transformation between robot hand and camera (T_x) was subtituted by those obtained from color-based method. Then the same evaluation of the performacne was conducted.

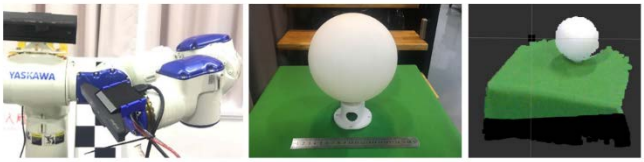

Figure 4. The depth-based eye-in-hand calibration experiments setups. (a) Camera mouning positin. (b) fixed sphere model as calibration tool. (c) the points cloud captured by RGBD camera.

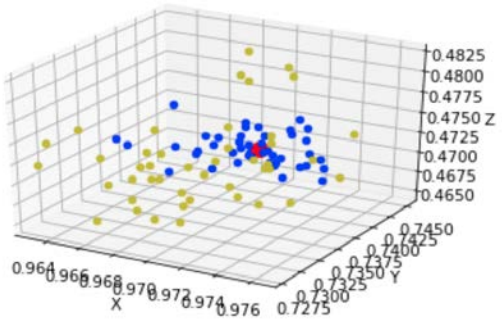

Figure 5. The projected sphere centers of both depth and color calibration.

Fig. 5 shows the projected sphere centers of both depth and color calibration results for all 45 points clouds in secend experiment. The red point is the estimated center of the sphere center. the yellow points are projected centers from color-based method and the blue points are from depth-based method. It can be seen that the points from color-based calibration are more scattered and disordered than points from depth-based calibration. To quantitatively evaluate the performance of both methods, The average distance from the estimated center to the projected center are calculated: for color-based method is 8.423 mm while for our depth-based mothod is 2.864 mm

The accuracy of RANSAC output are highly depended on the accuracy of points cloud. Therefore the performance of depth-based hand-eye calibraiton are largely affacted by points cloud. The depth density image captured from RealSense SR300 camera can only reach millimeter scale accuracy. Thus if one wish to highly improve the performance, using depth sensor with higher accuracy is a appealing choice.

VI. CONCLUTION

This paper introduced a high-accuracy, simple and elegant approach for robot hand-eye calibration with depth camera. In this approach, a sphere model are used to substitude chessboard, sophisticated 3D model or other calibraiton tools. The superiority of sphere model is that it has a simple and elegant paramiterized representation and also has geometrical invariability within different view. In our experiment, The quatitatively evaluation shows that our method are more accurate than the color-based method on RealSense SR300 camera. Certainly, our method can be used by all sensors which can capture point clouds. Future accuracy improvement can be achieved by improving the accuray of the depth sensor and by selecting well-suited hand-eye movements.

ACKNOWLEDGMENT

This work has been supported by National Natural Science Foundation of China (Grant No. 61673261) and Shanghai Kangqiao Robot Industry Development Joint Research Centre.

REFERENCES

[1] Tsai R Y, Lenz R K. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration[J]. IEEE Transactions on robotics and automation, 1989, 5(3): 345-358.

[2] Daniilidis K. Hand-eye calibration using dual quaternions[J]. The International Journal of Robotics Research, 1999, 18(3): 286-298.

[3] Strobl, K. H. and Hirzinger, G. (2006). Optimal hand-eye calibration. In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 4647–4653

[4] Reinbacher C, Rüther M, Bischof H. RoμNect: Hand mounted depth sensing using a commodity gaming sensor[C]//Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012: 461-464.

[5] Kahn S, Kuijper A. Fusing Real-Time Depth Imaging with High Precision Pose Estimation by a Measurement Arm[C]//CW. 2012: 256-260.

[6] Fuchs D I D B B A S. Calibration and multipath mitigation for increased accuracy of time-of-flight camera measurements in robotic applications[J]. 2012.

[7] Kahn S, Haumann D, Willert V. Hand-eye calibration with a depth camera: 2D or 3D?[C]//Computer Vision Theory and Applications (VISAPP), 2014 International Conference on. IEEE, 2014, 3: 481-489.

[8] Makadia A, Patterson A, Daniilidis K. Fully automatic registration of 3D point clouds[C]//Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. IEEE, 2006, 1: 1297-1304.

[9] Piatti D. Time-of-Flight cameras: tests, calibration and multi-frame registration for automatic 3D object reconstruction[J]. PhD diss. Turin, Italy: Polytechnic University of Turin, 2011.

[10] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[M]//Readings in computer vision. 1987: 726-740.

[11] Shiu Y C, Ahmad S. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX= XB[J]. IEEE Transactions on robotics and automation, 1989, 5(1): 16-29.

[12] Chou J C K, Kamel M. Finding the position and orientation of a sensor on a robot manipulator using quaternions[J]. The international journal of robotics research, 1991, 10(3): 240-254.

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

[14] Ikits M. Coregistration of pose measurement devices using nonlinear least squares parameter estimation[M]//Technical Report UUCS-00-018. University of Utah, 2000.

[15] Horaud R, Dornaika F. Hand-eye calibration[J]. The international journal of robotics research, 1995, 14(3): 195-210.

[16] Chen H H. A screw motion approach to uniqueness analysis of head-eye geometry[C]//Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. IEEE, 1991: 145-151.

[17] Lu Y C, Chou J C K. Eight-space quaternion approach for robotic hand-eye calibration[C]//Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on. IEEE, 1995, 4: 3316-3321.

[18] Daniilidis K. Hand-eye calibration using dual quaternions[J]. The International Journal of Robotics Research, 1999, 18(3): 286-298.

[19] Strobl K H, Hirzinger G. Optimal hand-eye calibration[C]//Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, 2006: 4647-4653.

[20] Schmidt J, Niemann H. Data selection for hand-eye calibration: a vector quantization approach[J]. The International Journal of Robotics Research, 2008, 27(9): 1027-1053.

[21] Hamilton W R. Lectures on quaternions[M]. Hodges and Smith, 1853.

[22] Gill P E, Murray W, Wright M H. Practical optimization[J]. 1981.