2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

A 2-Dimensional Branch-and-Bound Algorithm for Hand-Eye Self-Calibration of SCARA Robots

Chengyu Tao[1], Na Lv[2] and Shanben Chen[1]

***Abstract*—Due to the high positioning accuracy and relatively low prices, SCARA robots are widely used in industrial fields. The objective of this paper is to propose a hand-eye self-calibration algorithm for SCARA robots which could consider both accuracy and computational cost. The previous global optimal hand-eye calibration algorithms based on branch-and-bound (BnB) optimization is limited by their expensive computational cost. The speed of these algorithms depends on the volume of the search space to a large extent, which is the main concern in this paper. Instead of searching over the 3-dimensional parameter space corresponding to the rotation component of hand-eye pose, a new 2-dimensional search space is defined by separating and coupling some calibration parameters by means of the special structure of SCARA robots, which have 4 degrees of freedom (DoFs) including three translation DoFs and only one rotation DoF. The simulation and real experiments show the similar accuracy but much faster speed of the proposed algorithm compared with previous optimal algorithms based on BnB.**

I. INTRODUCTION

In practical applications, in order to perceive the precise space position of an actual object through a camera mounted on the end-effector of a robot, it is necessary to identify the transformation of the camera coordinate system with respect to the end-effector coordinate system. The problem to identify this special transformation is called hand-eye calibration problem. We aim at the 4-DOFs (including 3 translational freedoms and 1 rotational freedom) SCARA robot, whose end-effector could only rotate around a fixed axis.

The general methods based on the knowledge of Lie groups [1, 2] and quaternions or dual quaternions [3-6] only optimize the algebraic errors without any geometric meanings, for instance re-projection errors. Moreover, the requirement for standard calibration objects restricts the applications of these algorithms in many outdoor robotics, in which self-calibration algorithms are needed. Karl and Hartley [7] introduced a framework to solve the relative pose problem by searching over the rotation space under re-projection angular errors, many global optimal methods for hand-eye self-calibration [8-10] are based on this framework. In addition, a similar strategy is used to solve the geometric registration problem [11]. However, as pointed out in [12], a main weakness of these global optimal estimation methods is their relatively high computational complexity, which restrains their actual applications greatly. For instance, in some medical robotic application [13], due to the frequent replacements of operating tools or vibration during long-time running, a computational-efficient algorithm is required.

There are two main factors which affect the speed of these algorithms, the volume of the search space and the tightness of the relaxed bounds. Some optimal methods based on BnB optimization have been reduced the computational cost by using tighter bounds [14, 15]. Unfortunately, these strategies could not be applied to hand-eye calibration problems directly. Here, we focus on reducing the dimension of the search space. The rotation component $\mathbf{R}_X$ of the hand-eye matrix is controlled by three parameters essentially. Thanks to the special configuration of the SCARA robots, this task has been realized by proposed algorithm.

In this paper, we propose a much faster algorithm for hand-eye calibration of SCARA robots based on BnB optimization. The unknown hand-eye calibration parameters are separated and coupled into two parts, the first part can be obtained global optimally by BnB optimization while the remaining one is calculated successively. The main contribution of this paper is that by separating and coupling parameters, the dimension of the search space is reduced from 3-dimension to 2-dimension, then the total number of feasibility tests in BnB optimization is greatly less than those of previous algorithms, which accelerate the running speed of proposed algorithm.

II. PROBLEM FORMULATION

The hand-eye calibration problem is to determine the transformation $\mathbf{X}$ of the end-effector coordinate system with respect to the camera coordinate system. Basically, the problem is based on the well-known equation

$$\mathbf{A}^k\mathbf{X} = \mathbf{X}\mathbf{B}^k \tag{1}$$

Note that, $\mathbf{A}^k$ and $\mathbf{B}^k$ represent the transformation of the kth relative camera motion and end-effector motion respectively. In the case of SCARA robots, the rotation part $\mathbf{R}_B^k$ of $\mathbf{B}^k$ has merely one freedom, which is the rotation angle around a fixed axis. Using the rotation and translation components of $\mathbf{A}^k$ and $\mathbf{B}^k$, (1) can be written as

$$\begin{aligned} \mathbf{R}_A^k\mathbf{R}_X &= \mathbf{R}_X\mathbf{R}_B^k \\ \mathbf{R}_A^k\mathbf{R}_X\vec{t}_X + \mathbf{R}_A^k\vec{t}_A^k &= \mathbf{R}_X\mathbf{R}_B^k\vec{t}_A^k + \mathbf{R}_X\vec{t}_X \end{aligned} \tag{2}$$

[1] Chengyu Tao and Shanben Chen are with School of Materials Science and Engineering, Shanghai Jiao Tong University, Shanghai, China taochengyu@sjtu.edu.cn

[2]Na Lv is with the School of Electronic Information and Electricity Engineering, Shanghai Jiao Tong University, Shanghai, China nana414526@sjtu.edu.cn

A. Object function

Though most works use image pixel re-projection distance as error metric, angular re-projection error is more consistent with the noise model for fish-eye or omnidirectional cameras and as valid as image pixel re-projection error even for projective cameras as pointed out in [16]. In the case of angular error metric, the image points are considered as unit directional vectors. Given a pair of image points $p_j^k \leftrightarrow q_j^k$, their unit directional vectors can be calculated by $\vec{u}_j^k \sim \mathbf{K}^{-1}p_j^k$ and $\vec{v}_j^k \sim \mathbf{K}^{-1}q_j^k$. Considering a target point x_j^k projected to $\vec{u}_j^k$ in the same coordinate frame and $\vec{v}_j^k$ in another, we have

$$\vec{u}_j^k \sim x_j^k, \qquad \vec{v}_j^k \sim \mathbf{R}_A^k(x_j^k - \vec{t}_A^k) \tag{3}$$

Then the angular errors ϵ_j^k, $\tilde{\epsilon}_j^k$ are given by the two angles $\epsilon_j^k = \angle\left(\vec{u}_j^k, x_j^k\right)$ and $\tilde{\epsilon}_j^k = \angle\left(\vec{v}_j^k, \mathbf{R}_A^k\left(x_j^k - \vec{t}_A^k\right)\right)$. We have the following objective function

$$\min_{\mathbf{R}_X, \vec{t}_X, x_j^k} \max_{k,j} \begin{cases} \angle\left(\vec{u}_j^k, x_j^k\right) \\ \angle\left(\vec{v}_j^k, \mathbf{R}_X\mathbf{R}_B^k\mathbf{R}_X^T\left(x_j^k - \vec{t}_A^k\left(\mathbf{R}_X, \vec{t}\right)\right)\right) \end{cases} \tag{4}$$

The existence of optimal solution of the above formulation has been proven in [7]. Notice that if $\mathbf{R}_X$ has been determined, (4) can be deduced to a kind of SOCP problem [17]. Traversing all possible solution $\mathbf{R}_X$ over the rotation space is impossible due to the extremely high computation complexity. An alternative algorithm is the BnB optimization.

III. ALGORITHM

The general idea of BnB optimization is to recursively subdivide the feasible domain by the feasibility tests [7, 8, 10, 18]. The computation complexity depends largely on the total number of feasibility tests, where the dimension of search space plays an important role. By means of the particularity of $\mathbf{R}_B^k$ of the SCARA robots, the dimension of search space could be reduced from 3-dimension in previous algorithms to 2-dimension here.

A. New search space

In this subsection, we will deduce the 2-dimension search space. For SCARA robots, the rotation axis of end-effector is fixed. Rewrite $\mathbf{R}_B^k$ as $\mathrm{Rot}\left(l_B, \theta_B^k\right)$, where $\mathrm{Rot}(\cdot)$ is the matrix exponential $\mathrm{Rot}(l,\theta) = e^{[l]_\times\theta}$ [1].

The rotation axis $\vec{l}_B$ could be transformed to $\vec{z}$ axis ($\vec{z} = [0,0,1]^T$) by introducing a rotation matrix $\tilde{\mathbf{R}}$. The following identity holds

$$\mathbf{R}_B^k = \tilde{\mathbf{R}}\,\mathrm{Rot}\left(\vec{z}, \theta_B^k\right)\tilde{\mathbf{R}}^T \tag{5}$$

where $\tilde{\mathbf{R}}$ meet the relationship $\vec{l}_B = \tilde{\mathbf{R}}\vec{z}$ [2]. Denote $\mathbf{R}_X\tilde{\mathbf{R}}$ as $\tilde{\mathbf{R}}_X$ and $\vec{r}_i, i = 1,2,3$ as columns of $\tilde{\mathbf{R}}_X$, we have

$$\mathbf{R}_X\mathbf{R}_B^k\mathbf{R}_X^T = e^{\left[\mathbf{R}_X\tilde{\mathbf{R}}\vec{z}\right]_\times\theta_B^k} = \mathrm{Rot}\left(\vec{r}_3, \theta_B^k\right) \tag{6}$$

Equation (6) means that $\mathbf{R}_X\mathbf{R}_B^k\mathbf{R}_X^T$ depends only on $\vec{r}_3$. Hence, $\vec{r}_3$ can be modeled by two parameters φ and ω as $\vec{r}_3 = [\sin\omega\cos\varphi \;\; \sin\omega\sin\varphi \;\; -\cos\omega]^T$. $\tilde{\mathbf{R}}_X$ can be modeled by three parameters φ, ω and γ.

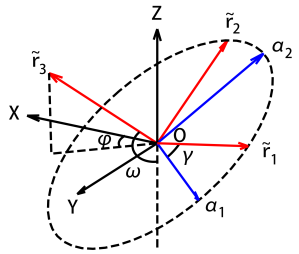

Fig. 1. The established coordinate system.

$$\tilde{\mathbf{R}}_X = \begin{bmatrix} -\sin\varphi & \cos\varphi\cos\omega & \sin\omega\cos\varphi \\ \cos\varphi & \sin\varphi\cos\omega & \sin\omega\sin\varphi \\ 0 & \sin\omega & -\cos\omega \end{bmatrix} \cdot \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Define $\vec{a}_1$ and $\vec{a}_2$ as the first two columns of the first matrix above, a special coordinate frame has been established, see Fig. 1. Rewrite the second equation of (2) as $\vec{t}_A^k = \tilde{\mathbf{R}}_X\vec{t}_B^k + \tilde{\mathbf{R}}_X\mathbf{M}_B^k\vec{t}_X$, where $\tilde{\mathbf{R}}^T\vec{t}_B^k$ and $\tilde{\mathbf{R}}^T\vec{t}_X$ are denoted as $\vec{t}_B^k$ and $\vec{t}_X$ respectively to simplify the notations, and $\mathbf{M}_B^k = \mathrm{Rot}\left(\vec{z}, -\theta_B^k\right) - \mathbf{I}$. Note that, the third column of $\mathbf{M}_B^k$ all equal zero, so $\vec{t}_{Xz}$ cannot be obtained directly [6, 19]. The method of obtaining $\vec{t}_{Xz}$ is shown in [6].

It is possible to eliminate the influence of γ by assuming that $\vec{t}_{sub,B}^k$ equals zero vector, where $\vec{t}_{sub,B}^k$ is the first two elements of $\vec{t}_B^k$. In this case, the end-effector just move along the translation joint of SCARA robot and rotate arbitrarily. The reason for this assumption is not just to reduce the mathematical difficulty but more to improve efficiency of evaluating bounding functions, which will be analyzed in section III-D. Denote $\mathbf{M}_{sub,B}^k$ as the second order principal sub-matrix of $\mathbf{M}_B^k$, $\vec{t}'_{sub,X}$ as $\mathbf{R}_{sub,\gamma}\vec{t}_{sub,X}$ and $\vec{t}_{sub,X}$ as the first two elements of $\vec{t}_X$, we have

$$\vec{t}_A^k = \vec{t}_{Bz}^k \cdot \vec{r}_3 + [\vec{a}_1 \;\; \vec{a}_2]\,\mathbf{M}_{sub,B}^k\vec{t}'_{sub,X} \tag{8}$$

B. Branch-and-bound algorithm

After defining the 2-dimensional parameter space (φ, ω) , we want to know whether there exists a better solution given current minimal objective value $\epsilon_{\min}$ when $\mathbf{R}_X$ is restricted to a small square block D_i with its center (φ_i, ω_i) and half side length σ_i, described as the following **Problem 0**. If the answer is true, we could subdivide the square block and ask the same questions in its inherits, otherwise ignore it.

Problem 0. *Given D_i, σ_i, $\epsilon_{\min}$, do there exist $\varphi, \omega \in D_i$, $\vec{t}'_{sub,X}$ and x_j^k, for all j and k, such that $\angle\left(\vec{u}_j^k, x_j^k\right) \le \epsilon_{\min}$ and $\angle\left(\vec{v}_j^k, \mathrm{Rot}\left(\vec{r}_3(\varphi,\omega), \theta_B^k\right)\left(x_j^k - \vec{t}_A^k\left(\varphi, \omega, \vec{t}'_{sub,X}\right)\right)\right) \le \epsilon_{\min}$?*

Now, we will outline the main steps of BnB optimization. Given the current minimal objective value $\epsilon_{\min}$ of (4), the

main procedures are illustrated in the following **Algorithm 1**.

Algorithm 1 The Branch-and-Bound optimization

Input: a priority queue q with an initial division set $\{D_i\}$, and an appropriate initial cost estimate $\epsilon_{\min}$.

Output: optimal candidate set $\left\{\varphi^*, \omega^*, \epsilon^*, \vec{t}^*_{sub,X}\right\}$, where (φ^*, ω^*) are the center of block D^* with the estimate objective value ϵ^* and sampled $\vec{t}^*_{sub,X}$.

1: **repeat**
2: Obtain the first block D from q.
3: **if** the answer of the ***Feasibility test 2*** (see Section III-C) is positive **then**
4: re-evaluate new cost estimate ϵ_{new} by sampling $\vec{t}_{sub,X}$ in a feasible domain (See Section III-D).
5: **if** $\epsilon_{new} < \epsilon_{\min}$ **then**
6: $\epsilon_{\min} \leftarrow \epsilon_{new}$
7: **end if**
8: Divide D into four sub-blocks $(D_d)_{d=1}^4$ and insert them into q.
9: **end if**
10: Delete D from q.
11: **until** The sum of the areas of all feasible blocks in q is less than a threshold $s_{\min}$.

C. Feasibility test

The feasibility test is to determine whether there is a solution to **Problem 0**. The problem itself is highly complex and nonlinear, so we propose an alternative relaxed version **Feasibility test 1** by fixing φ and ω at the center of D_i.

Feasibility test 1. *Given D_i, σ_i, $\epsilon_{\min}$, $\vartheta_{\max} = \max_k \left|\theta_B^k\right|$, ${}^1\varphi$ and ${}^1\omega$ corresponding to the center of D_i, do there exist t_A^k and x_j^k, for all j and k, such that $\angle\left(u_j^k, x_j^k\right) \leq \epsilon_{\min}$ and $\angle\left(v_j^k, \mathrm{Rot}\left(\vec{r}_3({}^1\varphi, {}^1\omega), \theta_B^k\right)\left(x_j^k - t_A^k\right)\right) \leq \epsilon_{\min} + \sqrt{2}\vartheta_{\max}\sigma_i$?*

In the following proposition, we will prove that the **Feasibility test 1** is a relaxed version about the **Problem 0**, which means that the feasibility of original problem could be judged by the relaxed problem.

Lemma 1. *If there exists a solution to Problem 0, there is a solution to Feasibility test 1. On the other hand, if there is no solution to the problem 0 on a domain D, then D could be split into subdomains D_i with sufficiently small half side length σ_i such that there is no solution to Feasibility test 1 on each subdomain D_i*

Proof. Firstly, we prove the first proposition. Assuming that there is a solution $\left({}^0\varphi, {}^0\omega, \vec{t}_{sub,X}, x_j^k\right)$ to **Problem 0** and denoting $\vec{t}_A^k\left({}^0\varphi, {}^0\omega, \vec{t}_{sub,X}\right)$ as ${}^0\vec{t}_A^k$, $x_j^k - {}^0\vec{t}_A^k$ as $\vec{s}_j^k$, $\vec{r}_3({}^0\varphi, {}^0\omega)$ as ${}^0\vec{r}_3$ and $\vec{r}_3({}^1\varphi, {}^1\omega)$ as ${}^1\vec{r}_3$, then $\left({}^0\vec{t}_A^k, x_j^k\right)$ is a solution to **Feasibility test 1** as proven below.

$$\begin{aligned}
&\max_{k,j} \angle\left(\vec{v}_j^k, \mathrm{Rot}\left({}^1\vec{r}_3, \theta_B^k\right)\vec{s}_j^k\right) \\
&\leq \max_{k,j} \angle\left(\vec{v}_j^k, \mathrm{Rot}\left({}^0\vec{r}_3, \theta_B^k\right)\vec{s}_j^k\right) + \\
&\qquad \max_{k,j} \angle\left(\mathrm{Rot}\left({}^1\vec{r}_3, \theta_B^k\right)\vec{s}_j^k, \mathrm{Rot}\left({}^0\vec{r}_3, \theta_B^k\right)\vec{s}_j^k\right) \quad (9) \\
&\leq \epsilon_{\min} + \max_k \angle\left(\mathrm{Rot}\left({}^1\vec{r}_3, \theta_B^k\right), \mathrm{Rot}\left({}^0\vec{r}_3, \theta_B^k\right)\right) \quad (10) \\
&\leq \epsilon_{\min} + \max_k \left|\theta_B^k\right| \left\|{}^1\vec{r}_3 - {}^0\vec{r}_3\right\| \quad (11) \\
&\leq \epsilon_{\min} + \max_k \left|\theta_B^k\right| \left\|\left[{}^1\varphi, {}^1\omega\right]^T - \left[{}^0\varphi, {}^0\omega\right]^T\right\| \quad (12) \\
&\leq \epsilon_{\min} + \sqrt{2}\vartheta_{\max}\sigma_i \quad (13)
\end{aligned}$$

The triangle inequality is used in (9) and so does Lemma 1 of [7] in (10), Lemma 2 of [7] in (11) and $({}^1\varphi, {}^1\omega)$ and $({}^0\varphi, {}^0\omega)$ belong to the same block D_i with half side length σ_i. In order to prove (12), we need another Lemma 2 in this paper as shown as follows.

As for the second proposition in this Lemma, the proof is similar to that of Lemma 3 in [7], so we do not reiterate it. □

Lemma 2. *For any parameters $\left({}^1\varphi, {}^1\omega\right)$ and $\left({}^2\varphi, {}^2\omega\right)$, denote $\vec{r}_3\left({}^1\varphi, {}^1\omega\right)$ as ${}^1\vec{r}_3$, $\vec{r}_3\left({}^2\varphi, {}^2\omega\right)$ as ${}^2\vec{r}_3$, the following inequality holds.*

$$\left\|{}^1\vec{r}_3 - {}^2\vec{r}_3\right\| \leq \left\|\left[{}^1\varphi, {}^1\omega\right]^T - \left[{}^2\varphi, {}^2\omega\right]^T\right\| \quad (14)$$

Furthermore, for any constant 2×1 vector v, denote $\vec{a}_i\left({}^j\varphi, {}^j\omega\right)$ as ${}^j\vec{a}_i$ and $\left[{}^j\vec{a}_1\ {}^j\vec{a}_2\right]$ as ${}^j\mathbf{S}$, we have another inequality

$$\left\|\left({}^1\mathbf{S} - {}^2\mathbf{S}\right)\vec{v}\right\| \leq \|\vec{v}\| \left\|\left[{}^1\varphi, {}^1\omega\right]^T - \left[{}^2\varphi, {}^2\omega\right]^T\right\| \quad (15)$$

Proof. Let J_1 and J_2 be the Jacobians of the mapping $f : y = [\varphi, \omega]^T \rightarrow \vec{r}_3$ and the mapping $f : y = [\varphi, \omega]^T \rightarrow \mathbf{S}\vec{v}$, then the maximal eigenvalue of $J_1^T J_1$ is 1 and that of $J_2^T J_2$ equals $\|v\|$. Therefore, the inequalities (14) and (15) are established. □

The work [7] deduces **Feasibility test 1** into equivalent linear problem

$$\vec{n}_j^{k\,T}\vec{t}_A^k \geq 0 \qquad \vec{m}_j^{k\,T}\vec{t}_A^k \geq 0 \quad (16)$$

Equation (16) are nonlinear inequalities about ${}^0\varphi$, ${}^0\omega$ and $\vec{t}_{sub,X}$ which still can not be solved easily. As done in [10], (16) can be further relaxed by fixing ${}^0\varphi$ and ${}^0\omega$ at $({}^1\varphi, {}^1\omega)$, which is the center of block D_i .

Denoting $\cos^{-1}\left(1 - \sigma_i^2\right)$ as ε and $\angle\left(\vec{n}_j^k, \vec{t}_{Bz}{}^1\vec{r}_3\right)$ as ξ_n, $\angle\left(\vec{m}_j^k, \vec{t}_{Bz}{}^1\vec{r}_3\right)$ as ξ_m, the relaxed test is shown as below.

Feasibility test 2. *Given D_i, half side length σ_i, normals $\vec{n}_j^k$, $\vec{m}_j^k$, ${}^1\varphi$ and ${}^1\omega$ corresponding to the center of D_i, do there exist $\vec{t}_{sub,X}$ such that for both $o \in \{n, m\}$*

$$\vec{o}_j^{k\,T}\left(\vec{t}_{Bz}{}^1\vec{r}_3 + \left[{}^1\vec{a}_1, {}^1\vec{a}_2\right]\mathbf{M}_{sub,B}^k\vec{t}_{sub,X}\right) + \delta_o \geq 0? \quad (17)$$

where

$$\delta_o = 2L_X \cos((\pi-\varepsilon)/2) + \begin{cases} \vec{t}^k_{Bz}\,(1-\cos\xi_o) & \textit{if } \xi_o < \varepsilon \\ \vec{t}^k_{Bz}\,(\cos(\xi_o-\varepsilon)-\cos\xi_o) & \textit{otherwise} \end{cases} \tag{18}$$

Lemma 3. *if there is a solution to Feasibility test 1, then this solution will also be a solution to Feasibility test 2.*

Proof. Let ${}^1\vec{t}_{sub,X}$ with its L_2 norm having an upper bound L_0 be a solution to the inequalities constraints (16) (**Feasibility test 2**), we have $\mathbf{M}^k_{sub,B}{}^1\vec{t}_{sub,X}$ will also have an upper bound L_X (this could derived by the computation of the largest eigenvalue of $\mathbf{M}^k_{sub,B}$, which is $\sqrt{2-2\cos\theta^k_B}$. Using the Lemma 2 of [10] and the cosine rule on (15) , the following inequality can be obtained.

$$\vec{t}^k_{Bz}\cdot\vec{n}_j^{k^T}{}^0\vec{r}_3 \le \vec{t}^k_{Bz}\cdot\vec{n}_j^{k^T}{}^1\vec{r}_3 + \begin{cases} \vec{t}^k_{Bz}\,(1-\cos\xi_n) & \text{if } \xi_n < \varepsilon \\ \vec{t}^k_{Bz}\,(\cos(\xi_n-\varepsilon)-\cos\xi_n) & \text{otherwise} \end{cases} \tag{19}$$

With the use of Lemma 3 of [10], another inequality can be obtained.

$$\vec{n}_j^{k^T}\left[{}^0\vec{a}_1\ {}^0\vec{a}_2\right]\mathbf{M}^k_{sub,B}{}^1\vec{t}_{sub,X} \le 2L_X\cos((\pi-\varepsilon)/2) + \vec{n}_j^{k^T}\left[{}^1\vec{a}_1\ {}^1\vec{a}_2\right]\mathbf{M}^k_{sub,B}{}^1\vec{t}_{sub,X} \tag{20}$$

□

Finally, using the Lemma 1 and Lemma 3, we have the following relationship between all feasibility tests.

$$\neg Feasibility\ test\ 2 \rightarrow \neg Feasibility\ test\ 1 \rightarrow \neg Problem\ 0 \tag{21}$$

D. Object function evaluation

In order to update $\epsilon_{\min}$, we need to sample a possible ${}^1\vec{t}_{sub,X}$ from the feasible domain of (17), triangulate $\left\{x^k_j\right\}$ and then re-evaluate the objective value. The objective function evaluation highly influence the overall convergence rate. It is possible that even a sampled solution still cannot update $\epsilon_{\min}$. The work [8] uses nonlinear optimization to obtain a better sample point at the expense of higher computational cost. However, in our case, we only randomly sample a solution but get relatively satisfactory performance. The reason is that there are only two variables in the feasible domain. A sample with higher dimension (for instance, 3 variables including ${}^1\vec{t}_{sub,X}$ and γ) means higher uncertainty which may make the whole evaluation less effective. This is another reason why we eliminate γ in (8). As for triangulation, we use the algorithm in [16], which is optimal under L_∞ angular error metric and consistent with our objective function (4).

E. Convergence analysis

The half side length σ_i of subdomain D_i tends to 0 as the algorithm runs, the terms $\sqrt{2}\theta_{max}\sigma_i$, δ_n and δ_m introduced by the feasibility tests tend to vanish. In the limit case, the Feasibility test 2 is equivalent to the **Feasibility test 0**. Therefore, if the **Feasibility tests 2** is infeasible in all subdomains D_i when $\sigma_i \to 0$, then all **Feasibility tests 1** will also be infeasible, so will the **Problem 0**. In fact, when the **Problem 0** is infeasible and σ_i is sufficiently small, there is no solution to **feasibility test 2** in each D_i. From the second part of Lemma 1, we could find a sufficient small σ_p such that **Feasibility test 1** is infeasible in all subdomains. For each domain D_i, we could find a maximal σ_{im} satisfying $0 < \sigma_{im} \le \sigma_p$ which could guarantee that the **Feasibility test 2** is also infeasible. This could be derived from observations that for two unequal objective values $\epsilon_1 < \epsilon_2$, the boundary of their feasible domains are separated and that the relaxed terms in **Feasibility test 2** could tend to 0. Since the number of subdomains is finite, so the minimum of $\{\sigma_{im}\} > 0$. Therefore, once the **Feasibility test 2** is infeasible in all subdomains with the half side length smaller than $min\{\sigma_{im}\}$, we could conclude that the **Problem 0** is infeasible.

F. Remaining parameters calculation

In fact, we could get a possible optimal candidate set $\left\{\vec{t}^*_{sub,X}, \varphi^*, \omega^*\right\}$ with corresponding objective value set $\{\epsilon^*\}$ by the BnB optimization (algorithm 1). In order to obtain the unknown γ, $\vec{t}^k_{sub,B}$ cannot be zero vector as assumed in (8). This requirement could be conducted easily by exchanging and recombining the former robot motions. Denote $\vec{t}^k_A(\gamma) = \mathbf{P}^k[\cos\gamma, sin\gamma]^T + \vec{f}^k$ where $\mathbf{P}^k$ and $\vec{f}^k$ are combinations of $\vec{t}^*_{sub,X}$, φ^* and ω^* according to (9). Now, we formulate the **Problem 1**.

Problem 1. *Given $\vec{t}^*_{sub,X}$, φ^* and ω^*, the task is to determine x^k_j and γ which minimizes the cost estimate ϵ subjecting to*

$$\begin{aligned} &\angle\left(u^k_j, x^k_j\right) \le \epsilon \\ &\angle\left(v^k_j, \mathrm{Rot}\left(\vec{r}^*_3, \theta^k_B\right)\left(x^k_j - t^k_A(\gamma)\right)\right) \le \epsilon \end{aligned} \tag{22}$$

The above problem will be a feasibility problem if ϵ is determined. Using the coplanarity condition [7], the **Problem 1** can be transformed into finding a feasible solution $\left(x^k_j, \gamma\right)$ which meets the following inequalities that for $o \in \{n, m\}$

$$\vec{o}^k_j(\epsilon)^T\mathbf{P}^k[\cos\gamma, \sin\gamma]^T + \vec{o}^k_j(\epsilon)^T\vec{f}^k \ge 0 \tag{23}$$

The above inequalities have the form of $\sin(\gamma+\psi_i) + d_i \ge 0$, thus the range of γ is the intersection of several sets of intervals. We can search over a set of possible ϵ for the minimal $\epsilon_{\min}$ when the range of γ is not empty. One possible method is the binary search. If the set $\{\gamma^*\}$ has obtained, then the final hand-pose $\left\{\mathbf{R}^*_X, \vec{t}^*_{sub,X}\right\}$ can be calculated by the former definition. We could estimate the re-projection angular errors for all candidate, the candidate with minimal error (no matter L_2 or L_∞ metric) will be regarded as optimal solution.

IV. Experiments

In this section, synthetic and real world data were used to evaluate the proposed algorithm. The **Feasibility test 2** was solved by CVXPY [20]. The program was implemented by PYTHON on personal laptop with 2.3 GHz Intel Core i5 CPU.

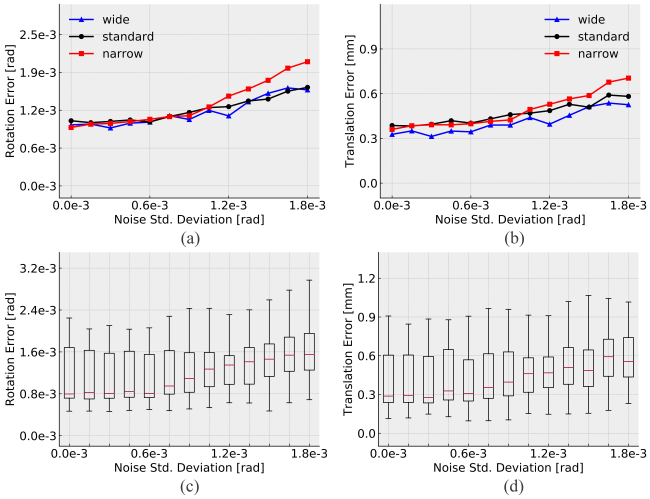

Fig. 2. (a) and (b) Rotation and translation mean deviation over different FOV and noise levels. (c) and (d) rotation and translation box plots over noise levels in the "standard" FOV level.

A. *Synthetic data*

we generated several sets of synthetic scenes for the hand-eye calibration of SCARA robots. Each experiment contains ten pairs of views. There are 64 uniformly distributed 3D point located within a ball with radius 200 mm in each scene. In order to analysis the effect of field of view (FOV), other 5 balls with smaller radius were generated. FOV decreases as the radius of the balls. Different scenes are grouped into three categories "wide", "standard" and "narrow" with FOV equals $117.4°, 67.9°, 38.8°$ respectively. Further, we added noises to the direction of u_j^k and v_j^k. The step size of the noise standard deviation is 1.5×10^{-4} rad. we use 13 different noise levels and noised data were generated by 10 times at each level.

Ten different hand-eye transformation matrices X were generated. The rotation axis of SCARA robot was $\vec{z}$ axis simply. Relative robot motions were generated randomly but meet our previous assumption that the translations are along the rotation axis. The camera model was "Pinhole" model. The initial cost estimate $\epsilon_{\min}$ and the final termination condition $s_{\min}$ were set to 0.01 and 0.0004 respectively. We used an 8×16 initial subdivisions.

Simulation results are shown in Fig. 2. The rotation deviation is measured by the rotation angle of the delta rotation $R_X^T R_X^*$ between estimated rotation R_X^* and ground truth rotation R_X. The translation deviation is measured as the L_2 norm error between estimated and ground truth translation. Fig. 2 illustrates the rotation and translation deviation over different FOV and noise levels.

As shown above, both rotation and translation error increase with noise level at all FOV levels. Fig 2c and 2d mean that the robustness of proposed algorithm is not perfect, since we only randomly sample from the feasible domain in order to decrease the computation cost and do not actually obtain the point which reaches the minimum. The algorithm performs best at the "wide" FOV level, which is consistent with the experiments of [8]. The trend of computation time at

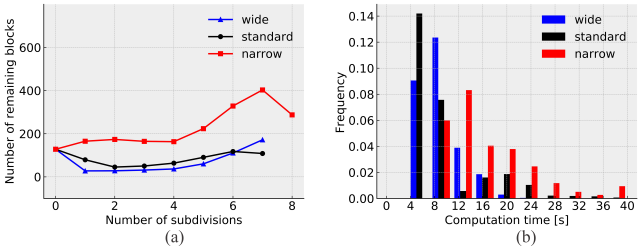

Fig. 3. (a) The average number of remaining blocks plotted against the number of subdivision phases. Note that only the most common subdivision modes in each FOV level are shown here. (b) histogram of computation time.

different FOV levels also meets their conclusion. The average computation time of "wide", "standard" and "narrow" FOV levels are 13.1s, 13.8s and 24.5s respectively. What's more, the median computation time of these three levels are 11.6s, 11.2s and 19.0s respectively. The results of computation time and the number of remaining blocks are shown in Fig. 3

Next, we realized the algorithm [10] (denoted as 'R12') to compare with our algorithm (denoted as 'T21'). Due to the high time cost, we only tested under the "standard" FOV level (radius equal 110 mm) with the noise standard deviation 0.6×10^{-3} rad. We randomly shuffled the views to eliminate the interference of the assumption of relative motions. The initial subdivision was set to be $16 \times 16 \times 16$ and the algorithm stopped when the remaining volumes reached below 3×10^{-5} (it's smaller than the termination condition in our algorithm since it is computed by the remain volumes not the areas in our algorithm). The final mean rotation and translation errors of the algorithm [10] are 1.11×10^{-3} rad and 0.392 mm respectively while the errors of our algorithm are 1.12×10^{-3} rad and 0.427 mm respectively. The errors are similar but the mean computation time of the comparison algorithm is 2326.3s and that of our algorithm is 14.3s. The average number of remaining blocks of the two algorithms are shown in Fig. 4.

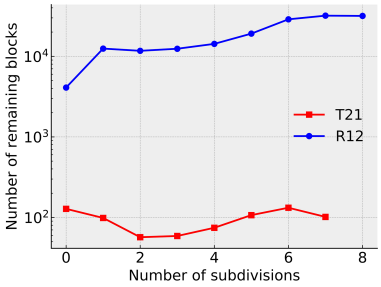

Fig. 4. The average number of remaining blocks of our algorithm and the algorithm [10] plotted against the number of subdivision phases

Finally, we compared the proposed algorithm with three previous ones under the "standard" FOV level. Since previous methods are not self-calibration method and need standard calibration pattern to compute camera external parameters, we added manufacturing and structural errors (no more than 5‰, similar to [21]) to the 3D positions of points. The

labels 'Z11', 'H98' and 'A99' stand for [22], [19] and [23] respectively. The proposed algorithm 'T21' achieve the best performance. Though the foundation of 'Z11' is built on 'A99', 'Z11' is under L_∞ metric and more sensitive to noise than that of L_2 metric, which lead to the worst performance of 'Z11'. Fig. 5 show the comparison results.

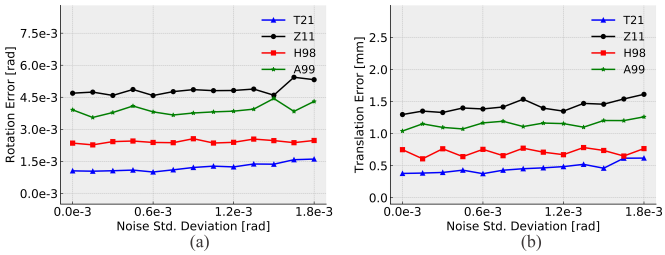

Fig. 5. (a) and (b) Rotation and translation errors of proposed algorithm and previous methods.

B. Real data

A STH030S/E serial SCARA arm with a MI5100 CMOS and a ST-65105-A5 lens (focal length 4.2mm and field of view 80°) were used to acquire real calibration data sets. The rotation axis of relative motions is the $\vec{z}$ axis of the end-effector coordinate frame. 7 pairs of relative views were used to compute the hand-eye matrix per dataset. One pair of the relative views are shown in Fig. 6 for example.

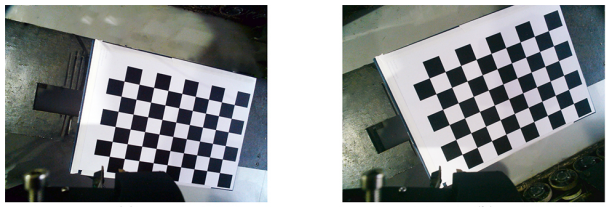

Fig. 6. A pair of views used for the hand-eye calibration.

The internal parameters of the camera were calibrated using the algorithm [24]. The corner points were detected by OPENCV software and then transformed into directional vectors. There are 54 correspondences used for hand-eye calibration per pair of views.

The final computation time on the real data sets was 122.9s. There are two main reasons for the relative longer computation time on the real data set, one is that the 2D chessboard used for calibration provides weaker constraints due to its relative small size and few corner points, and another is that the position and direction of the camera are constrained by the motion space of the actual manipulator, which limits the ability to collect more comprehensive calibration data. The distribution of final re-projection angular errors using obtained hand-eye parameters were computed to evaluate the performance of proposed algorithm on real data sets. The results are shown in Fig. 7.

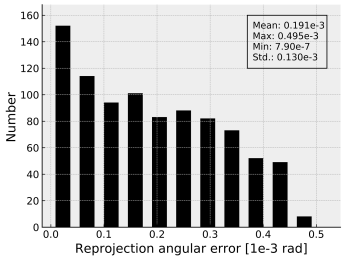

Fig. 7. Histogram of re-projection angular errors on real data.

V. Conclusion

We propose a faster optimal algorithm for hand-eye self-calibration of SCARA robots with one rotational DoF under re-projection angular errors in this paper. The previous global optimal algorithms search for global solution over a 3-dimensional search space, whose computational complexity is much higher, resulting some limitations on their applications in some situations which require the near real-time performance of a hand-eye calibration algorithm. Similarly, our proposed algorithm contains the branch-and-bound optimization. However, utilizing the special properties of this kind of robots, we reduce the search space of branch-and-bound optimization to a 2-dimensional parameter space by coupling some calibration parameters, which could improve the computation efficiency a lot. The experiments on synthetic data and real data show that our algorithm can achieve good balance between accuracy and computational cost.

VI. Acknowledgements

This work is partly supported by the National Natural Science Foundation of China under the Grant No.61873164, 51975367, and Guangxi Natural Science Foundation of China, No.GKAD18281007.

References

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

[2] 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.

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

[4] K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," in *proceedings of 13th International Conference on Pattern Recognition*, vol. 1. IEEE, 1996, pp. 318–322.

[5] J. Schmidt, F. Vogt, and H. Niemann, "Robust hand–eye calibration of an endoscopic surgery robot using dual quaternions," in *Joint Pattern Recognition Symposium*. Springer, 2003, pp. 548–556.

[6] M. Ulrich and C. Steger, "Hand-eye calibration of scara robots using dual quaternions," *Pattern Recognition and Image Analysis*, vol. 26, no. 1, pp. 231–239, 2016.

[7] R. I. Hartley and F. Kahl, "Global optimization through rotation space search," *International Journal of Computer Vision*, vol. 82, no. 1, pp. 64–79, 2009.

[8] J. Heller, M. Havlena, and T. Pajdla, "A branch-and-bound algorithm for globally optimal hand-eye calibration," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1608–1615.

[9] T. Ruland and K. Dietmayer, "Globally optimal hand-eye calibration under free choice of cost-function," in *2012 IEEE Second International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*. IEEE, 2012, pp. 1–5.

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

[11] T. M. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," *Computer Vision and Image Understanding*, vol. 90, no. 3, pp. 258–294, 2003.

[12] A. Parra Bustos and T.-J. Chin, "Guaranteed outlier removal for rotation search," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2165–2173.

[13] Z. Zhang, L. Zhang, and G.-Z. Yang, "A computationally efficient method for hand–eye calibration," *International journal of computer assisted radiology and surgery*, vol. 12, no. 10, pp. 1775–1787, 2017.

[14] A. Parra Bustos, T.-J. Chin, and D. Suter, "Fast rotation search with stereographic projections for 3d registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3930–3937.

[15] K. Joo, T.-H. Oh, J. Kim, and I. S. Kweon, "Robust and globally optimal manhattan frame estimation in near real time," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 682–696, 2018.

[16] S. H. Lee and J. Civera, "Closed-form optimal two-view triangulation based on angular errors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[17] F. Kahl and R. Hartley, "Multiple-view geometry under the L_∞-norm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1603–1617, 2008.

[18] Y. Seo, Y.-J. Choi, and S. W. Lee, "A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1173–1178.

[19] H. Zhuang, "Hand/eye calibration for electronic assembly robots," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 612–616, 1998.

[20] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[21] J. Heller, M. Havlena, and T. Pajdla, "Globally optimal hand-eye calibration using branch-and-bound," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 1027–1033, 2015.

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

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

[24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.