2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Online Hand-Eye Calibration with Decoupling by 3D Textureless Object Tracking

Li Jin[1,4], Kang Xie[1,4], Wenxuan Chen[2], Xin Cao[1,4], Yuehua Li[2], Jiachen Li[3], Jiankai Qian[1,4], Xueying Qin[1,4,*]

***Abstract*— Hand-eye calibration estimates the pose of a camera relative to a robot, which is a fundamental problem for visually guided robots, especially for dynamic object grasping. Most methods use 2D fiducial markers with distinctive visual features and require pre-calibration for accurate calibration, which can not work online. In this paper, we propose a novel hand-eye calibration method based on the natural 3D object, which can work online and automatically even if the object is textureless or weakly textured. We first propose a Pose Refinement Network (PR-Net) to improve the accuracy of 3D object tracking. Then we build a 3D convergence point constraint based on the multi-view information with the accurate object pose to adjust the object position. Finally, we optimize the hand-eye pose by the closed-loop constraint with the optimized object position, solving the problem that is easy to fall into a local minimum. The experiments show that the average error of our hand-eye calibration method is 1.20 degrees and 23.18 mm. The results achieve state-of-the-art by using the working object to realize the online hand-eye calibration.**

I. INTRODUCTION

The camera is an essential sensor for the robot to perceive the environment and is crucial for visually guided robots. Hand-eye calibration solves the pose of the camera (eye) relative to the end of robot arm (hand). It then unifies the coordinate system of the hand and all perceived objects, enabling the robot arm to interact with objects precisely and naturally in tasks such as dynamic target grasping.

Most approaches for vision-based hand-eye calibration use single or multiple 2D fiducial markers for stable and robust calibration, and some are online methods [1]. Depth video can generally obtain more accurate poses, and spherical 3D markers are used to get better hand-eye calibration accuracy. However, all of these methods require the arrangement of markers or spheres in advance. Self-calibration methods [2] employ structure-from-motion theory by visual features in the site, but in practice, they still use markers to provide sufficient feature points to complete the calibration. Most of these methods are offline, and hand-eye calibration needs to be performed again when the relative pose of camera and robot change. Calibrating the hand-eye offline during outdoor exploration tasks can be challenging. It is necessary to quickly complete hand-eye calibration online.

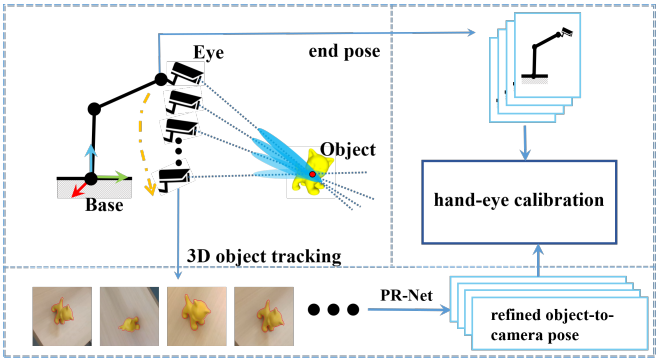


Fig. 1. Flowchart of our on-line hand-eye calibration with textureless or weakly textured objects pipeline.

The online approach requires the robot to perform hand-eye calibration at any time during the work without markers. If the poses of an object in the scene can be estimated in real-time, it is very convenient to use it to estimate the hand-eye pose automatically [3]. Unfortunately, this kind of method relies heavily on the rich features of the object and cannot work for textureless or weakly textured objects. It optimizes the hand-eye pose with a closed-loop constraint, in which both the hand-eye pose and the object position are optimized simultaneously. Consequently, the optimization is difficult to converge or quickly falls into the local minimum. Therefore, it requires manual assignment of initial value rather than automaticity, making it difficult to achieve accurate calibration in the working state.

In outdoor engineering, objects are often textureless or weakly textured, such as industrial parts, in-orbit satellite, etc. These objects for online automatic hand-eye calibration will raise higher requirements for object-to-camera poses tracking accuracy and optimization algorithms. To solve the above problems, we propose an online hand-eye calibration method based on 3D textureless or weakly textured object tracking trajectories. The specific contributions are as follows:

- An online automatic hand-eye calibration method is proposed to achieve accurate hand-eye calibration without initial value, which establishes constraints by 3D tracking of textureless or weakly textured objects.
- A 3D convergence point constraint is built, which disentangles the coupling of the object's pose and the hand-

This work is supported by the National Natural Science Foundation of China (No. U21B6001), Center-initiated Research Project of Zhejiang Lab (No.2021NB0AL01), and NSF of China (No. 62172260, 62202425).

[1] School of Software, Shandong University, P.R. China
[2] Intelligent Robot Research Center, Zhejiang Lab, P.R. China
[3] State Key Lab of CAD&CG, Zhejiang University, P.R. China
[4] Engineering Research Center of Digital Media Technology, Ministry of Education, P.R. China
[*] Corresponding Author.

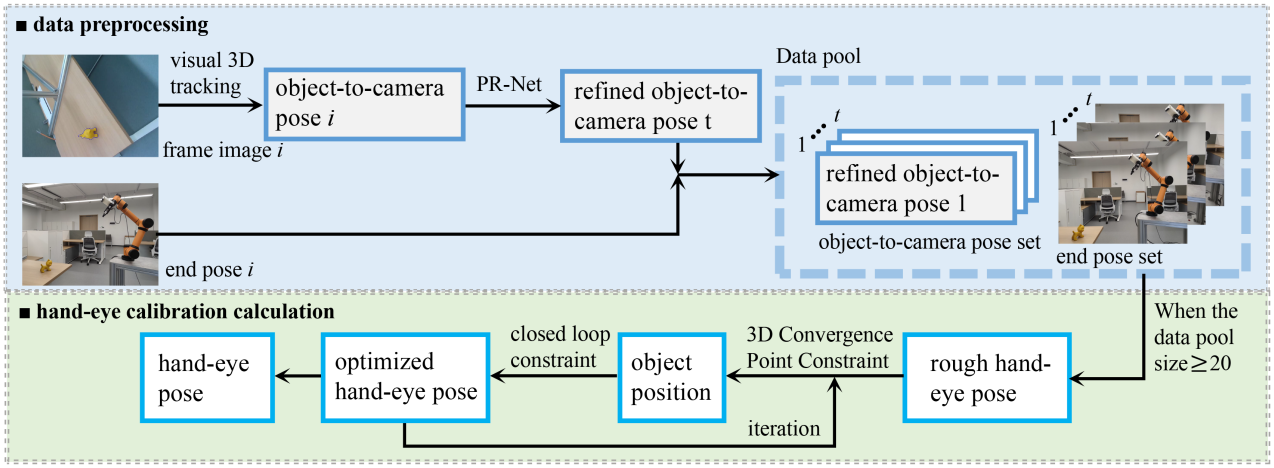


Fig. 2. The framework of the proposed method. Data preprocessing (blue) collects the object-to-camera poses and the robotic end poses. The object-to-camera poses are estimated by 3D object tracking and then refined with RP-Net. After getting enough valid datasets (more than 20 frames), hand-eye calibration calculation (green) is triggered. The initial value of the hand-eye pose is obtained with method [4], and then we optimize the hand-eye pose iteratively using the 3D convergence point constraint and closed-loop constraint.

eye pose by the strategy of alternating iterations of the closed-loop constraint and the convergence constraint.

- A transformer-based 3D tracking refined network PR-Net is proposed, which significantly improves the accuracy of object poses.

II. RELATED WORK

High-accuracy hand-eye calibration mainly involves two aspects: the estimation of the camera pose and the solution of the hand-eye pose. The former depends on accurate correspondences of visual features and the latter focuses on how to accurately resolve the hand-eye pose in the presence of data noise.

The hand-eye pose can be solved accurately with precise camera pose by 2D markers, where the pose representation can use the Lie algebra [4], Quaternion [5], [6], Procrustes analysis [7], etc. Optimization approaches can improve the robustness with some constraints, such as minimizing re-projection error [8], [9], or the pose through closed-loop transformation [8], [10].

ChArUco Board with a group of markers is usually applied for the online calibration method [1]. Some targets with distinctive features can also replace the 2D planar markers, such as 3D standard balls [11]. However, these methods require to pre-set markers in the environment.

Using scenes or objects in the working place eliminates the need for pre-set markers. The self-calibration methods [2], [12]–[14] solve the hand-eye pose by tracking the camera with feature matching in a static environment. Thus, it is affected by the richness of features and image quality. It is very convenient to calibrate the hand-eye pose by tracking an object [3], in which 3D tracking obtains the camera poses. However, the calibration accuracy of this method is heavily dependent on the richness of visual features of the object. Therefore it will fail if the object is weakly textured or even textureless. Moreover, it employs only the closed-loop constraint, which suffers from the coupling of hand-eye pose and object positions, and makes the optimization challenging to perform.

Textureless or weakly textured objects widely exist in the working environment of robots, which is difficult to extract accurate feature points in the picture and hardly match between different frames. The 3D tracking methods based on object CAD models are relatively mature [15]–[17], which can obtain the 6D poses of the objects in real-time. However, the 3D tracking methods suffer from severe noise. Recently, deep learning shows the ability to get 3D information from images [18], and the transformer has also proven the ability to extract fused multi-modal features [19]. It makes it possible to use the network to improve the accuracy of 3D tracking.

III. METHOD

Our hand-eye calibration method is performed automatically online by using an object.

A. Overview

Our hand-eye calibration consists of two main procedures: 3D object pose tracking and hand-eye calibration. As shown in Fig. 2, the object-to-camera poses are estimated with CAD model of the object by a 3D tracking method [16] and refined by our PR-Net, while the corresponding kinematic parameters of the robotic arm are collected to calculate the end poses. After there are at least 20 pairs of object-to-camera poses and the end poses, the hand-eye calibration can be processed, and updated with new information continuously collected online.

In the process of hand-eye calculation, a rough hand-eye pose can be computed by using one of hand-eye calibration methods, such as [4]. Further, we optimize the object position using the 3D convergence point constraint, then update the hand-eye pose using the closed-loop constraint, and repeat these two processes until convergence. These alternating iterations can get a more reliable hand-eye pose and object position in the base coordinate system.

The hand-eye calibration task has two typical cases: eye in hand (camera is fixed on the robotic arm) and eye out of hand (camera is fixed outside the robotic arm). These two cases are solved optimally using similar strategies.

B. Transformer-based PR-Net

Textureless or weakly textured objects hardly extract reliable and distinguishable features. The 6D pose obtained by 3D tracking algorithms has a significant error, and improving its accuracy can improve the accuracy of the hand-eye calibration. To solve this problem, we propose the PR-Net based on the transformer to refine the results of 3D tracking.

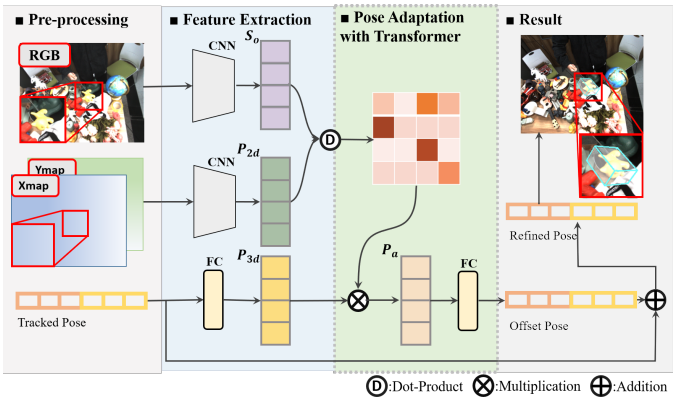


Fig. 3. Architecture overview of PR-Net. In the pre-processing stage (left), we crop RGB images, Xmap, and Ymap to obtain the object from the scene, where the Xmap and Ymap represent the x-axis and y-axis positions of the 2D image, respectively. The cropped outputs and initial tracking poses are sent to the network for pose refinement. Our PR-net (middle) uses a transformer network to adapt the features and predicts the offset pose to compensate for the initial pose to get a more accurate pose(right).

As shown in Fig. 3, PR-Net uses the image data, the 2D position map, and the object pose obtained by the 3D tracking algorithm as input. To eliminate much background information irrelevant to the object being tracked, we first crop the image using the mask generated by the tracking pose. However, cropping the image causes the loss of the 2D position information. We introduce Xmap and Ymap as the 2D information of the object, which makes up for the loss of 2D information and reduces the redundant information. After that, we use the feature extraction module to extract the appearance feature S_o, the 2D position feature P_{2d}, and the 3D pose feature P_{3d} obtained by tracking, respectively.

These extracted features are sent to a pose adaptation module using a transformer network [20]. It first computes the similarity matrix of the appearance features S_o and 2D position features P_{2d}, then projects the 3D features into the dimensional space P_a composed of the appearance features and position, thus measuring the offset between the tracking pose and the actual pose. We train the network using L_1loss as follows:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{O}_i - \tilde{\boldsymbol{O}}_i\right\|_1 \tag{1}$$

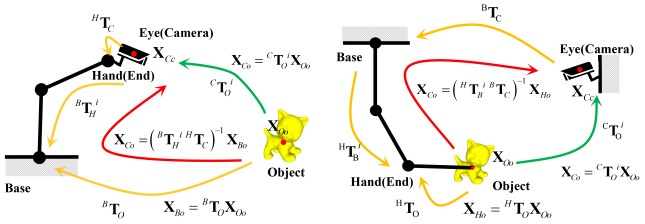
N is the number of predicted positional offset parameters, $\boldsymbol{O}_i$ is the ground truth value of the pose offset, and $\tilde{\boldsymbol{O}}_i$ is the pose offset predicted by the network. The offset is compensated to the tracking pose to obtain the object-to-camera pose ${}^{C}\mathbf{T}_{O}$.

C. The case of the eye in hand

In the case of the eye in hand, the camera is fixed to the robotic arm's end and follows its movement. The object is static under the base coordinate system, and the robot arm is unrestricted to move around the object to complete the calibration. The process is divided into three parts: closed-loop constraint, 3D convergence point constraint, and alternate iterations to solve the hand-eye position pose.

Variable representation: The object, base, end, and camera are denoted by the letters O, B, H, and C, respectively, with upper case letters representing the coordinate system and lower case letters denoting the position in coordinates. The Euclidean transformation matrix between coordinate systems is denoted by $\mathbf{T}$. The right subscript indicates the original coordinate system, and the left superscript indicates the transformed coordinate system. For example, ${}^{H}\mathbf{T}_{C}$ denotes the camera-to-end pose. Points in 3D space are represented by $\mathbf{X}$. We use the first subscript to indicate the coordinate system in which they are located and the second subscript to indicate the object to which the point belongs, e.g., $\mathbf{X}_{Bo}$ suggests the representation of the object center in the base coordinate system. Since the data are time-ordered, we use the superscript i to show the corresponding data at the moment i.

closed-loop constraint: The closed-loop constraint can optimize the camera-to-end pose ${}^{H}\mathbf{T}_{C}$ [3]. As shown in Figure 4(a), the transformation among the coordinate systems in the hand-eye system forms a closed-loop. Using the constraint of the object position through different path transformations is the same in the camera coordinate system, then the solution equation for the hand-eye pose ${}^{H}\mathbf{T}_{C}$ is established.



(a) the eye in hand (b) the eye out of hand

Fig. 4. Closed-loop constraint.

The object center in the base coordinate system $\mathbf{X}_{Bo}$, transformed to the camera coordinate system ${}^{H}\mathbf{T}_{C}^{-1}{}^{B}\mathbf{T}_{H}^{-1}\mathbf{X}_{Bo}$; the object center in object coordinate system $\mathbf{X}_{Oo}$ transformed to the camera coordinate system ${}^{C}\mathbf{T}_{O}^{i}\mathbf{X}_{Oo}$. In principle, these two positions should be the same. Therefore, the error function is obtained as follows:

$$\mathbf{e}^{i} = {}^{C}\mathbf{T}_{O}^{i}\mathbf{X}_{Oo} - {}^{H}\mathbf{T}_{C}^{-1}{}^{B}\mathbf{T}_{H}^{-1}\mathbf{X}_{Bo} \tag{2}$$

The method [3] minimizes the norm at all moments, and also optimizes ${}^{H}\mathbf{T}_C, \mathbf{X}_{Bo}$ by minimizing this function:

$$({}^{H}\mathbf{T}_C, \mathbf{X}_{Bo})^* = \arg\min_{{}^{H}\mathbf{T}_C, \mathbf{X}_{Bo}} \sum_{i=1}^{n} (\mathbf{e}^i)^\top \mathbf{e}^i \tag{3}$$

However, the hand-eye pose ${}^{H}\mathbf{T}_C$ and the object position in the base $\mathbf{X}_{Bo}$ are interdependent. There is a coupling between the parameters to be optimized, which leads to difficulties in optimizing the solution. And the simultaneous optimization of the hand-eye pose ${}^{H}\mathbf{T}_C$ and the object position $\mathbf{X}_{Bo}$ is nonlinear and easy to converge to local minima, causing optimization failure.

3D convergence point constraint: The monocular 3D tracking method for textureless or weakly textured object is estimated by minimizing the re-projection error of the object on the image plane, therefore the error distribution of the object position is quite narrow on the image plane, while it is relatively wide in the ray of viewing direction, as shown in Fig. 5(left). Fortunately, the object has been actually observed from multiple view points, thus it should locate at a stationary position no matter where the camera is. Therefore, all those rays from the center of the camera to the object in different moments will form a 3D convergence point, as shown in Fig. 5(right).

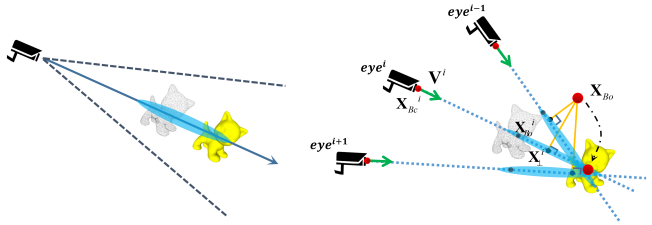


Fig. 5. Tracking characteristics and 3D convergence point constraint.

First, the coordinate system of the robot base is stationary , and we represent centers of camera and object in this coordinate system as follows:

$$\mathbf{X}^i_{Bc} = {}^{B}\mathbf{T}^i_H \, {}^{H}\mathbf{T}_C \mathbf{X}_{Cc} \tag{4}$$

$$\mathbf{X}^i_{Bo} = {}^{B}\mathbf{T}^i_H \, {}^{H}\mathbf{T}_C \, {}^{C}\mathbf{T}^i_O \mathbf{X}_{Oo} \tag{5}$$

Thus, the convergence point should be on all viewing rays starting from $\mathbf{X}^i_{Bc}$ to $\mathbf{X}^i_{Bo}$. Unfortunately, these 3D rays are impossible to meet at a point, and we should find a 3D point $\mathbf{X}_{Bo}$ that has the smallest sum distance with all these rays. We call $\mathbf{X}_{Bo}$ as a convergence point. The cost function is:

$$E_{\mathrm{g}} = \sum_{i=1}^{n} \left\| \mathbf{X}_{Bo} - \mathbf{X}^i_{\perp} \right\|_2^2 \tag{6}$$

where $\mathbf{X}^i_{\perp}$ denotes the vertical foot of $\mathbf{X}_{Bo}$ corresponded to the ray of sight at i-th moment.

In order to express $\mathbf{X}^i_{\perp}$, the unit vector $\mathbf{V}^i$ of the ray direction from camera center $\mathbf{X}^i_{Bc}$ to object center $\mathbf{X}^i_{Bo}$ is defined:

$$\mathbf{V}^i = (\mathbf{X}^i_{Bo} - \mathbf{X}^i_{Bc}) / \|\mathbf{X}^i_{Bo} - \mathbf{X}^i_{Bc}\|_2 \tag{7}$$

The $\mathbf{X}^i_{\perp}$ is the projection point of convergence point $\mathbf{X}_{Bo}$ on the ray, thus it can be formulated as the following:

$$\mathbf{X}^i_{\perp} = \mathbf{X}^i_{Bc} + (\mathbf{V}^i)^\top (\mathbf{X}_{Bo} - \mathbf{X}^i_{Bc}) \mathbf{V}^i \tag{8}$$

It should be noticed that it is a linear formula about the convergence point $\mathbf{X}_{Bo}$, when all others are known or calculated.

Therefore, minimizing E_{g} in Eq. 6 is a linear least squares problem in case of fixed hand-eye pose. Based on a simple derivation, we can get the Jacobi matrix:

$$\frac{\partial E_g}{\partial \mathbf{X}_{Bo}} = \sum_{i=1}^{n} diag(1-(v^i_x)^2, 1-(v^i_y)^2, 1-(v^i_z)^2) \\ (\mathbf{X}_{Bo} - \mathbf{X}^i_{Bc} + (\mathbf{V}^i)^\top \mathbf{X}_{Bo} \mathbf{V}^i + (\mathbf{V}^i)^\top \mathbf{X}^i_{Bc} \mathbf{V}^i) \tag{9}$$

For reasons of space limitation, the detailed derivation process is not given here. Finally, the convergence point, which is the optimized object position $\mathbf{X}_{Bo}$, can be obtained by matrix calculation when the hand-eye pose is known.

Alternating iterative solving hand-eye pose: Since the 3D convergence point constraint can optimize the object position independently, the object position optimization is separated from the closed-loop constraint. For this reason, the closed-loop constraint is to only optimize the hand-eye pose. Therefore, an alternately iteration strategy of the convergence constraint and closed-loop constraint is used to disentangle the coupling of object poses and hand-eye pose.

First, we fix the hand-eye pose ${}^{H}\mathbf{T}_C$, the object position $\mathbf{X}_{Bo}$ is obtained by minimizing the objective function of the 3D convergence point constraint:

$$\mathbf{X}_{Bo}{}^* = \arg\min_{\mathbf{X}_{Bo}} E_{\mathrm{g}} \tag{10}$$

Then, fixed the object position $\mathbf{X}_{Bo}$, we only optimize hand-eye pose ${}^{H}\mathbf{T}_C$ using the closed-loop constrain:

$${}^{H}\mathbf{T}_C{}^* = \arg\min_{{}^{H}\mathbf{T}_C} \sum_{i=1}^{n} (\mathbf{e}^i)^\top \mathbf{e}^i \tag{11}$$

These two optimization procedures alternately iterate. In this way, given the hand-eye pose ${}^{H}\mathbf{T}_C$ obtained by the closed-loop constraint, the 3D convergence point constraint can optimize the object position. Next, given the object position $\mathbf{X}_{Bo}$ obtained by the 3D convergence point constraint, the closed-loop constraint can calculate the hand-eye pose more precisely. These alternating iterations can achieve accurate and robust hand-eye pose ${}^{H}\mathbf{T}_C$.

The object position and hand-eye pose are optimized by different functions separately. Due to the fixed hand-eye pose, the optimal object position is not affected by the change of the hand-eye pose and is unique. Similarly, the object position is fixed to make the optimal hand-eye pose unique. Therefore, we achieve the decoupling of object position and hand-eye pose. Then we solve the problem that the nonlinear optimization is prone to fall into the minimal value.

D. The case of the eye out of hand

In the case of the eye out of hand, the camera is fixed outside of the robotic arm, and the arm grasps the object in motion to calibrate the camera-to-base pose ${}^{B}\mathbf{T}_{C}$, as shown in fig. 4(b).

When viewed under the end coordinate system, the object is static; otherwise, the base and the camera are in motion. Corresponding to the coordinate systems of the two cases, we can calibrate the hand-eye pose in the coordinate systems of the eye out of hand similarly. The end coordinate system is equated to the base coordinate system in the case of the eye in hand; the base coordinate system is equated to the end coordinate system. The camera and object coordinate systems remain unchanged. Further, by transforming the camera and object to the end coordinate system, we use the 3D convergence point to solve the object position $\mathbf{X}_{Ho}$. Then combining the closed-loop constraint, we alternate iterations to solve the camera-to-base pose transformation ${}^{B}\mathbf{T}_{C}$.

Algorithm 1: Online Hand-Eye Calibration with Decoupling by 3D Textureless Object Tracking

Input: N: configuration set size, L: sequence length, $\{\mathbf{T}_r^t, {}^{B}\mathbf{T}_H^t\}$:measurements at time t

Output: final estimate:$\{{}^{H}\mathbf{T}_C^L, \mathbf{X}_{Bo}^L\}$

initialization;
for $t = 1$ *to* L **do**
  // $c()$ Computes the object-to-camera poses
  ${}^{C}\mathbf{T}_O^t = c(\mathbf{T}_r^t)$;
  $\mathbf{s}_t = \{{}^{B}\mathbf{T}_H^t, {}^{C}\mathbf{T}_O^t\}$;
  $\mathcal{S}_t^* = \mathcal{S}_{t-1}^* \cup \mathbf{x}_t$;
  if $t > N$ **then**
    // $g()$ Computes the rough hand-eye pose
    $\mathbf{y}_r^t = g(\mathcal{S}_t^*)$;
    // $a()$ updates the calibration estimate
    $\{{}^{H}\mathbf{T}_C^t, \mathbf{X}_{Bo}^t\} = a(\mathbf{y}_r^t, \mathcal{S}_t^*)$;
  end
end

IV. EXPERIMENTS

We implement our experiments on a desktop with i7 3.7GHz CPU and an RTX2080s. We use AUBO-i5 robotic arm for real environment tests, which repeats positioning accuracy can reach ±0.02mm. The MER-31-210UM/C camera with 1280×1024 resolution is used for the case of the eye out of hand. The RealSense-SR300 camera with 640×480 resolution is used for the case of the eye in hand, where we only use the RGB data. We used two 3D printed weakly textured objects, a cat and a rabbit, from the RBOT [16] dataset. More experiments are shown in the video.

TABLE I

TRACKING COMPARISONS ON BCOT DATASET [23]. (↑): HIGHER BETTER, (↓): LOWER BETTER. (R) AND (S) DENOTE THE REFINE FOR RBOT AND SLOT RESULTS, RESPECTIVELY.

	Error($\circ/mm$)(↓)				Accuracy(%)(↑)	
	$erot$	etr	ADD	$ADDs$	ADD	$ADDs$
RBOT [16]	4.25	23.52	23.78	13.18	53.63	78.97
SLOT [17]	3.31	18.90	19.13	10.84	62.77	84.50
PR-Net(R)	4.23	17.15	17.51	10.00	69.34	88.01
PR-Net(S)	**3.30**	**14.11**	**14.41**	**8.52**	**75.28**	**92.60**

A. Evaluation metrics

We use rotation error $erot$ and translation error etr [12] to measure hand-eye calibration accuracy and ADD [21] and ADDs [22] metrics for 3D object tracking accuracy. We report accuracy within 10% of object's diameter [22].

B. PR-Net

We first evaluate the improvement of PR-Net on 3D tracking accuracy. The 3D tracking dataset BCOT [23] with ground truth poses is used to train and test the PR-Net network. We use 80% of the dataset as the training set and the other 20% as the testing set. We train each object separately.

We use the 3D tracking methods [16] and [17] for comparison with the proposed PR-Net. Table I compares the error and accuracy of different methods, highlighting PR-Net's superior performance in precise object position location. However, the network faces challenges in rotational estimation due to insufficient rotation representation in images.

C. Hand-eye calibration

We took six sets of data in real scenes to evaluate the method's effectiveness, and the trajectory of the robot arm is shown in Figure 6. The speed of the our entire system is 35 FPS, which meets the requirements for real-time calibration.

We mainly compare two categories of algorithms. One is the hand-eye calibration method which uses 2D markers to obtain the camera poses. For the fairness of the comparison, we replace the camera poses obtained by markers with those by 3D tracking. The algorithms include $T17_1$ [8], $T17_2$ [8], $K19$ [9], and $W21$ [7]. The other is the method [3] using 3D object tracking.

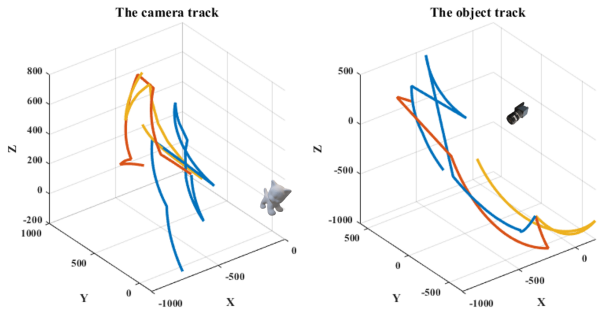


Fig. 6. Eye in hand and eye out of hand trajectory.

TABLE II

HAND-EYE CALIBRATION RESULTS IN REAL SCENES: COMPARISON WITH SOTA METHODS. BEST RESULTS ARE HIGHLIGHTED IN **BOLD**. SECOND-BEST RESULTS ARE HIGHLIGHTED IN *underline*. "*nan*" DENOTES THAT THE ROTATION ERROR OF THE METHOD IS GREATER THAN 30 DEGREES OR THE TRANSLATION ERROR IS GREATER THAN 1 METER.

	eye in hand						eye out of hand					
	seq1		seq2		seq3		seq1		seq2		seq3	
error	$erot^{(\circ)}$	$etr(mm)$	$erot^{(\circ)}$	$etr(mm)$	$erot^{(\circ)}$	$etr(mm)$	$erot^{(\circ)}$	$etr(mm)$	$erot^{(\circ)}$	$etr(mm)$	$erot^{(\circ)}$	$etr(mm)$
$T17_1$	3.26	110.28	2.83	72.83	1.29	106.35	4.03	109.24	3.17	78.95	2.16	57.82
$T17_2$	5.41	82.51	4.27	68.41	*nan*	997.28	6.32	113.64	*nan*	*nan*	*nan*	*nan*
$K19$	*nan*	*nan*	*nan*	154.12	*nan*	539.22	4.73	97.88	2.99	57.79	13.43	277.85
$W21$	2.50	63.16	4.74	115.66	1.87	**31.88**	3.28	77.27	2.81	74.67	3.31	45.13
$^{\star}K16_1$	19.17	*nan*	*nan*	*nan*	*nan*	*nan*	3.96	58.88	0.83	22.03	1.99	59.42
$^{\star}K16_2$	4.85	43.02	1.44	43.04	17.15	279.04	2.85	41.18	0.82	21.67	1.72	54.78
ours	**2.44**	**16.01**	**0.77**	**30.87**	**0.89**	32.47	**0.84**	**22.17**	**0.65**	**16.48**	**1.60**	**21.11**

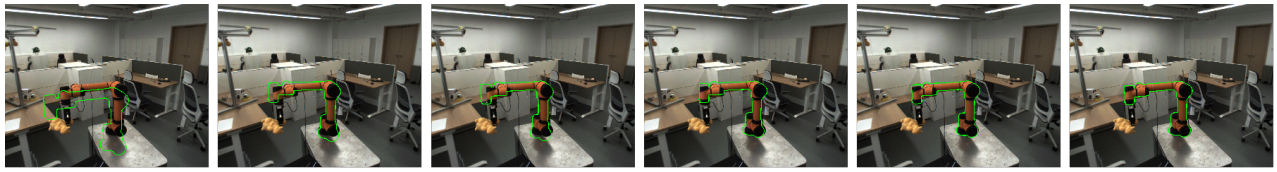


Fig. 7. Snapshots from the online calibration procedure applied to real-world sequences involving a static camera out of hand.

Method [3] lacks initial value calculation, requiring manual setting for hand-eye calibration, preventing automation. Calibration of large samples depends on small sample calibration, but [3] diverges in small samples due to significant 3D tracking error of textureless or weak texture. To address this, we use the same initial value for hand-eye calibration in any sample size and numerical solution method [4] to obtain meaningful results. Thus, $^{\star}K16_1$ and $^{\star}K16_2$ use different initial values to utilize small and large samples' results.

The results are in Table II. $T17_1$, $T17_2$, $K19$, and $W21$ methods failed due to large errors in object-to-camera pose obtained from 3D object tracking. Algorithm [3] optimizing both the hand-eye pose and object position parameters failed for $^{\star}K16_1$, and $^{\star}K16_2$ had difficulty obtaining accurate results. Our proposed method improved the hand-eye calibration's rotation and translation accuracy, achieving the best positional results and proving its effectiveness.

Figure 7 shows the case of the eye out of hand as our method gradually converges in the iterations. The robot arm in space is reprojected to the image using the hand-eye poses obtained from the calibration. As the amount of data increases, the outline of the reprojected robotic arm increasingly fits the robotic arm on the image.

D. Ablation study

We evaluate the effectiveness of PR-Net, the closed-loop constraint, and iterative optimization with convergence constraint, as shown in the table III. We use the RBOT tracking algorithm [16] for 3D object tracking and use a numerical solution method [4] for the initial hand-eye pose.

The numerical hand-eye calibration results are not significantly improved by PR-Net refinement of 3D pose tracking. This is because the translation parameter is always estimated based on the rotation parameter in the numerical method, and PR-Net has limited improvement in rotation accuracy. The use of closed-loop constraints, however, improves accuracy even with simultaneous optimization of hand-eye pose and object position. The accuracy is further improved when closed-loop and 3D convergence point constraints are combined with alternate iterative strategies. The highest accuracy is achieved with PR-Net optimization.

Our closed-loop constraint improved accuracy compared to numerical methods, and iterative optimization strategies further improved it. PR-Net also improved accuracy. Compared to Table I, our hand-eye pose accuracy is close to object tracking and positioning, showing the effectiveness of our method.

V. CONCLUSION

We proposed an online hand-eye calibration method for textureless objects, achieving robust results despite poor tracking accuracy. Decoupling hand-eye pose and object position using iterative optimization with convergence and closed-loop constraints led to improved accuracy. Our PR-net enhances 3D tracking pose accuracy and calibration preciseness. Point cloud information will be added in future work to improve accuracy in industrial scenarios.

TABLE III

ABLATION STUDY. CL DENOTES CLOSED-LOOP CONSTRAINT, AND 3CPCI DENOTES ITERATIVE OPTIMIZATION WITH 3D CONVERGENCE POINT CONSTRAINT.

			eye in hand		eye out of hand	
PR-Net	CL	3CPCI	$erot^{(\circ)}$	$etr(mm)$	$erot^{(\circ)}$	$etr(mm)$
			15.30	112.49	4.19	106.78
✓			15.30	105.65	4.19	107.14
	✓		4.85	43.02	2.65	38.37
✓	✓		3.22	29.58	2.09	26.62
	✓	✓	3.41	22.29	1.26	23.12
✓	✓	✓	**2.44**	**16.01**	**0.84**	**22.17**

REFERENCES

[1] W. Lin, P. Liang, G. Luo, Z. Zhao, and C. Zhang, "Research of online hand-eye calibration method based on charuco board," *Sensors*, vol. 22, no. 10, p. 3805, 2022.

[2] H. Zhuang, "Hand/eye calibration for electronic assembly robots," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, USA, April 20-25, 1997*, pp. 1341–1346, IEEE, 1997.

[3] K. Pauwels and D. Kragic, "Integrated on-line robot-camera calibration and object pose estimation," in *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016* (D. Kragic, A. Bicchi, and A. D. Luca, eds.), pp. 2332–2339, IEEE, 2016.

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

[5] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robotics Res.*, vol. 10, no. 3, pp. 240–254, 1991.

[6] Z. Fu, J. Pan, E. Spyrakos-Papastavridis, X. Chen, and M. Li, "A dual quaternion-based approach for coordinate calibration of dual robots in collaborative motion," *IEEE Robotics Autom. Lett.*, vol. 5, no. 3, pp. 4086–4093, 2020.

[7] J. Wu, Y. Sun, M. Wang, and M. Liu, "Hand-eye calibration: 4-d procrustes analysis approach," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 2966–2981, 2020.

[8] A. Tabb and K. M. A. Yousef, "Solving the robot-world hand-eye(s) calibration problem with iterative methods," *Mach. Vis. Appl.*, vol. 28, no. 5-6, pp. 569–590, 2017.

[9] K. Koide and E. Menegatti, "General hand-eye calibration based on reprojection error minimization," *IEEE Robotics Autom. Lett.*, vol. 4, no. 2, pp. 1021–1028, 2019.

[10] G. Wang, W. Li, C. Jiang, D. Zhu, H. Xie, X. Liu, and H. Ding, "Simultaneous calibration of multicoordinates for a dual-robot system by solving the AXB = YCZ problem," *IEEE Trans. Robotics*, vol. 37, no. 4, pp. 1172–1185, 2021.

[11] L. Yang, Q. Cao, M. Lin, H. Zhang, and Z. Ma, "Robotic hand-eye calibration with depth camera: A sphere model approach," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 2018.

[12] C. Tao, N. Lv, and S. Chen, "A 2-dimensional branch-and-bound algorithm for hand-eye self-calibration of SCARA robots," in *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pp. 11408–11414, IEEE, 2021.

[13] Z. Zhao, "Hand-eye calibration using convex optimization," in *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pp. 2947–2952, IEEE, 2011.

[14] S. Qiu, M. Wang, and M. R. Kermani, "A modern solution for an old calibration problem," *IEEE Instrum. Meas. Mag.*, vol. 24, no. 3, pp. 28–35, 2021.

[15] G. Wang, B. Wang, F. Zhong, X. Qin, and B. Chen, "Global optimal searching for textureless 3d object tracking," *Vis. Comput.*, vol. 31, no. 6-8, pp. 979–988, 2015.

[16] H. Tjaden, U. Schwanecke, E. Schömer, and D. Cremers, "A region-based gauss-newton approach to real-time monocular multiple object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1797–1812, 2019.

[17] H. Huang, F. Zhong, and X. Qin, "Pixel-wise weighted region-based 3d object tracking using contour constraints," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2021.

[18] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 6568–6577, IEEE, 2019.

[19] J. Wang, X. Hu, Z. Gan, Z. Yang, X. Dai, Z. Liu, Y. Lu, and L. Wang, "UFO: A unified transformer for vision-language representation learning," *CoRR*, vol. abs/2111.10023, 2021.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (I. Guyon, U. von Luxburg,

S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds.), pp. 5998–6008, 2017.

[21] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018* (H. Kress-Gazit, S. S. Srinivasa, T. Howard, and N. Atanasov, eds.), 2018.

[22] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6d pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 3003–3013, Computer Vision Foundation / IEEE, 2021.

[23] J. Li, B. Wang, S. Zhu, X. Cao, F. Zhong, W. Chen, T. Li, J. Gu, and X. Qin, "BCOT: A markerless high-precision 3d object tracking benchmark," *CoRR*, vol. abs/2203.13437, 2022.