The Visual Computer (2023) 39:381–391
https://doi.org/10.1007/s00371-021-02336-7

ORIGINAL ARTICLE

Active hand-eye calibration via online accuracy-driven next-best-view selection

Xinxin Zhang[1] · Yuefeng Xi[1] · Zhentao Huang[2] · Lintao Zheng[1] · Hui Huang[3] · Yueshan Xiong[1] · Kai Xu[1]

Accepted: 11 October 2021 / Published online: 4 January 2022

Abstract

We propose a novel high-accuracy active hand-eye calibration approach. In our method, the robot movement and camera view selection are both driven by and targeting the improvement of calibration accuracy. During the calibration process, the data acquisition is guided by an online estimated discrete viewing quality field (DVQF), representing the calibration quality of different views in various 3D locations. The view quality is measured by how much it reduces the uncertainty of calibration results and increases the diversity of robot poses, contributing to the calibration precision. Based on DVQF, we select the next-best-view as the target moving pose for each time step. A fully automatic system is presented to perform the overall hand-eye calibration process without any human intervention. Numerous experiments are conducted both in real-world and simulated scenarios. The proposed algorithm outperforms other approaches and shows much superiority in accuracy and robustness.

Keywords Hand-eye calibration · Automatic calibration · Next-best-view selection

1 Introduction

Hand-eye calibration is a process used in the field of robotics for determining the relative orientation of a camera (or sensor) in relation to a robot. It is essential for the vision-based robot operating system to guide robot movement and locate positional information in the surrounding workspace. There are various applications, including robotic grasping [1], medical procedures [2], and stereo vision [3] that require or, at least, could greatly benefit from the knowledge of hand-eye calibration. The accuracy of the calibration outcome could positively affect the decision-making procedure or the result of all the above applications.

As shown in Fig. 1, there are two different camera setups defined by the calibration problem, which are known as eye-in-hand, where the camera is rigidly attached to the robot end-effector, and eye-to-hand, where the camera is fixed in the workspace. Both setups could be formulated as the same mathematical form $\mathbf{AX = XB}$, where $\mathbf{X}$ is the matrix representing an unknown transformation of the robot an unknown transformation from the camera to the robot arm or the robot base, $\mathbf{A}$ and $\mathbf{B}$ are the transformation matrices representing the movement transformation of the robot arm and the camera, respectively. Most of the literature about hand-eye calibration focuses on how to solve the $\mathbf{AX = XB}$ Eq. [4]. Various closed-form or linear least-square solutions have been derived in the past to solve the mathematical representation [5–8].

However, the majority of those existing works perform offline, passive calibration analysis, in which the validation of their approach is conducted on the already acquired calibration data [5–8]. Unfortunately, this often significantly limits calibration accuracy since data acquisition is decoupled from the respective hand-eye calibration algorithms, i.e., in some situations, additional observation should be taken, or several previous captured views should be discarded in order to make more precise calculations. There are surprisingly limited works studying how to actively acquire calibration

Xinxin Zhang and Yuefeng Xi are joint first authors.

✉ Kai Xu
kevin.kai.xu@gmail.com

Lintao Zheng
lintaozheng1991@gmail.com

[1] National University of Defense Technology, Changsha 410000, China

[2] University of Nottingham Ningbo China, Ningbo 315100, China

[3] Shenzhen University, Shenzhen 518061, China

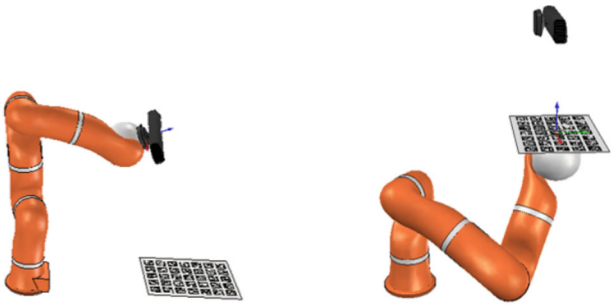


Fig. 1 Two types of hand-eye calibration: eye-in-hand (left), eye-to-hand (right)

data, which are most useful for improving the accuracy of calibration results. Besides that, few existing approaches can perform the whole calibration process automatically [9]. Antonello [9] solved the automation problem but did not improve the accuracy and robustness of hand-eye calibration. In most of the current calibration systems, the robot poses are predefined empirically and then conduct the robot arm shifting manually at each step. It is time-consuming and laborious to carry out the entire calibration process.

In this work, we propose a novel high-accuracy active hand-eye calibration method. In our system, the calibration process can be performed without any human intervention. The robot movement and camera view selection are guided by an online estimated discrete view quality field (DVQF) parameterized in the 3D view space. DVQF stores for each view a score measuring how much it reduces the uncertainty of calibration results and increases the diversity of robot poses. Based on DVQF, we select the next-best-view (NBV) as the target moving pose for each step. Our method shows high accuracy and robustness through extensive experimental evaluations and comparisons, which benefit from our online active hand-eye calibration process.

To sum up, the contributions of this work are listed as follows:

- A novel algorithm for active hand-eye calibration. It iteratively evaluates and chooses the next-best-view (NBV) according to a discrete view quality field (DVQF) in order to increase the accuracy and robustness of the calibration result.
- A fully automatic hand-eye calibration system which is easily deployed to most robotic arms and cameras, and is amenable to work with most calibration patterns.
- A simulation system based on Vrep [10] which encompasses settings of both eye-in-hand and eye-to-hand. The system introduces Gaussian noises over the positions of the end-effector to achieve a more realistic simulation.

2 Related work

Our goal is to enhance the standard hand-eye calibration process by improving the calibration accuracy and build an automatic hand-eye calibration system. Here, we focus on reviewing the most relevant work in hand-eye calibration.

Many theories had been proposed in history to solve the hand-eye calibration problem [4,6–8,11–15]. Shiu et al. firstly defined the problem as solving the function **AX = XB** and present the solution to it [8]. In this function, **A** and **B** are the relative transformations from respective previous pose to another pose, and **X** is the unknown transformation between the end-effector and camera. Besides this, solving the hand-eye calibration problem could also be formulated as solving the function **AX = YB**. In this function, **X** is the transformation between the end-effector and camera and **Y** is the transformation between the calibration board and robot agent. **A** and **B** are the poses of the end-effector and camera. The difference between these two functions is the number of unknown parameters. By solving the function **AX = YB**, two different transformations could be obtained, but it requires more data than solving the previous function.

There had been various solutions for the hand-eye calibration problem, which divided it into two stages: rotation and translation. Tsai and Lenz propose a method by solving the function and compute the rotation and translation [4]. Different from [8], Tsai and Lenz [4] use Euler angle rather than rotation matrix to solve the function results in the reduction of the numbers of unknown parameters. However, these methods do not show outstanding robustness through a small amount of data. It normally requires more than 50 sets of data for convergence. Horaud et al. [7,11,12] solve the problem by computing the rotation then calculate the translation using the least square method, which would increase the translation error by transferring from rotation error.

Rotation and translation could also be estimated simultaneously, which could avoid error transferring: [13] proposes an approach based on nonlinear constrained optimization. Li et al. [14] estimate the translation and rotation of X and Y by solving singular value decomposition in the form of dual quaternion and Kronecker product. Dekel [6] solves the problem by using the least square method based on dual quaternion.

Tabb and Ahmad Yousef select the form of **AX = YB** and utilize nonlinear optimizer to generate a new calibration pipeline [16]. It obtained satisfied results with different optimization targets, including rotation error, translation error and reprojection error. Based on [15], [16] presents rx, rz algorithm which targets **AX = XB** and **AX = YB**. It utilizes the Levenberg–Marquardt nonlinear optimization algorithm to reduce the reprojection error and results in hand-eye calibration accuracy increment.

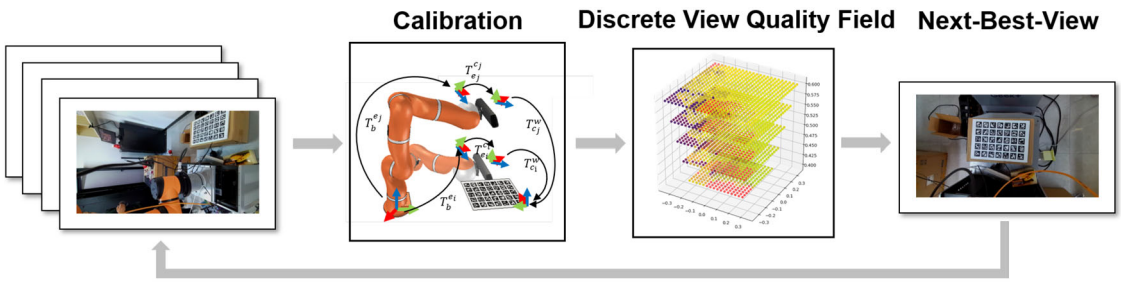


Fig. 2 A demonstration of the main process of the active hand-eye calibration

Except for previous approaches, in [17,18], two extended hand-eye calibration methods without the demand for a calibration pattern have been proposed. In [17], structure-from-motion and feature tracking were utilized. Though it removes the restriction of calibration patterns, its robustness extremely relies on the environmental condition. In [18], CAD model is used to replace the calibration pattern, which would raise the concern of the model's accuracy. Moreover, only certain types of the model could be used for calibration. In our scenarios, the AprilTags is a more practical choice than a known CAD model with high accuracy.

Most of the existing researches collect the data manually and then conduct an approach for estimating hand-eye calibration results. We focus on enhancing the data collection procedure, combining it with state-of-the-art algorithms to produce more satisfactory results.

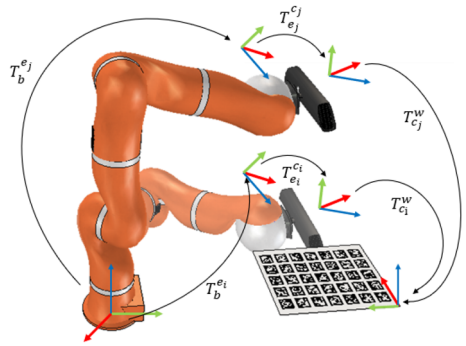


Fig. 3 An illustration of relative transformations in hand-eye calibration. Where i and j represent different data pairs, T_b^e is the transformation between the robot base and the end-effector. T_e^c is the transformation between the end-effector and the camera. T_c^w is the transformation between the camera and the calibration pattern

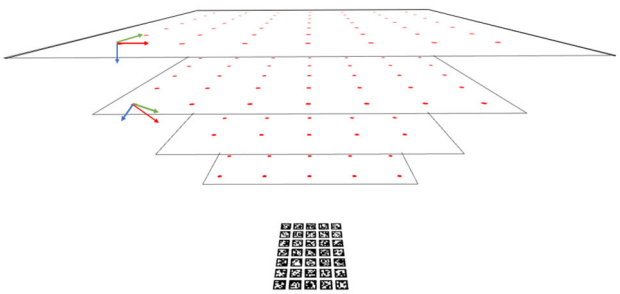


Fig. 4 An example of initialization with four sampled planes. Each plane contains various 3D positions(red dots) with different views to the calibration pattern

3 Methodology

The objective of this system is to perform high-accuracy calibration tasks without human intervention. The existing method conducts offline calibration by setting a fixed route for the robot arm [9]. We apply an online accuracy-driven camera view selection to complete the task. First, we collect five sets of data to perform initial calibration. Second, we construct the discrete view quality field (DVQF) based on the initial result. Finally, we perform an iterative online calibration parameter update based on the calculated result of selected next-best-view (NBV), illustrated in Fig. 2. The rest of this section would demonstrate the essential techniques in the system, including the discrete view quality field (DVQF), calibration method, view sampling, accuracy-based evaluation, and online hand-eye calibration parameters refinement.

3.1 Discrete view quality field

To get a clear evaluation with an online update of each camera view, we constructed a discrete view quality field (DVQF). It is a 3D discrete field measures the calibration accuracy of the most representative camera view. We considered each view's calibration accuracy from two aspects: the consistency of its derived robot end-effector pose based on existing hand-eye calibration parameters and how much it contributes to the generality of the calculated calibration parameters to adapt various inputs. Further details of these terms would be illustrated in Accuracy-Based Evaluation subsection.

Measuring every spatial area in the scenario is inefficient and meaningless since the calibration board should be viewed. Therefore, as shown in Fig. 4, we sampled various heights with multiple scales: with the distance to calibration board increases, the scale would become larger. Finally, we

evaluate each view in the sampled field based on two defined terms: *Uncertainty* and *Pose Diversity*.

3.1.1 Calibration method

In this subsection, we would present our calibration method for calculating the final output hand-eye calibration parameters and the construction of the discrete view quality field (DVQF).

Hand-eye calibration has strict vision requirements for viewing the calibration pattern due to the occlusion caused by the robot arm. Therefore, we select AprilTags [19] as our calibration pattern for estimating the camera pose. It could tolerate partial occlusion and maintain a high level of accuracy. We first acquire the calibration board corners (p_{obj}) and the corresponding point in the image (p_{img}). Then, apply the epnp algorithm [20] and obtain the transformation between camera axis and calibration board axis $T_b^w = q_b^w, t_b^w$. After that, we use the Levenberg–Marquardt nonlinear optimization implemented in SciPy library [21] for refinement:

$$\begin{aligned}\left\{q_b^w, t_b^w\right\} &= argmin \sum_{i=1}^{n} ||proj(K, D, \sigma(q_b^w, t_b^w) \cdot p_{obj}(i)) \\ &\quad - p_{img}(i)||\end{aligned} \tag{1}$$

In this equation, K is the intrinsic parameter of the camera, D represents distortion coefficients, $\sigma(q_b^w, t_b^w)$ represents transforming quaternion (q_b^w) and translation (t_b^w) to a transformation matrix. $proj(K, D, \sigma(q_b^w, t_b^w) \cdot p_{obj(i)})$ represent projecting 3D points to 2D images. By this algorithm, the accuracy of camera pose $\left\{q_b^w, t_b^w\right\}$ could be further refined.

Then, we calculate the hand-eye calibration result based on the previous extrinsic parameter we obtained. In this procedure, a great amount of camera poses correspond to robot poses are required to compute the output. We use the form of solving the **AX = XB** function to estimate the transformation between camera and robot agent. The relative transformations in this context are illustrated in Fig. 3. We could derive two equations (Eqs. 2 and 3) based on the transformations:

$$T_b^{e_i} \cdot T_{e_i}^{c_i} \cdot T_{c_i}^w = T_b^{e_j} \cdot T_{e_j}^{c_j} \cdot T_{c_j}^w \tag{2}$$

$$T_{e_i}^{c_i} = T_{e_j}^{c_j} = T_e^c \tag{3}$$

In the above equations, *b,e,c,w* represent robot base axis, end-effector axis, camera axis and calibration board axis, respectively. T_m^n represents the transformation from the m axis to the n axis, it is a 4*4 homogeneous matrix including the rotation and translation.

Since the relative position of the robot base and the calibration pattern is fixed, Eq. (2) is valid. And because of the fixed relative position of the camera and the end-effector, Eq. (3) is valid. By combining equation (2,3), we could obtain the following equation:

$$T_b^{e_j{}^{-1}} \cdot T_b^{e_i} \cdot T_e^c = T_e^c \cdot T_{c_j}^w \cdot T_{c_i}^{w^{-1}} \tag{4}$$

In Equation 4, $T_b^{e_j}$ is the position of end-effector in robot base's axis. $T_{c_i}^w$ represent camera's extrinsic parameters. The function **AX = XB** could be formulated by considering $A = T_b^{e_j{}^{-1}} \cdot T_b^{e_i}, B = T_{c_j}^w \cdot T_{c_i}^{w_{-1}}, X = T_{e_j}^{c_j} = T_{e_i}^{c_i}$. The method proposed by Dekel [6] is applied for solving this **AX = XB** function. After this, we uses the Levenberg–Marquardt nonlinear optimization for refinement:

$$\begin{aligned}\left\{q_e^c, t_e^c\right\} = argmin \sum_{i=1}^{n-1} \sum_{j=i}^{n} ||(T_b^{e_j} \cdot \sigma(q_e^c, t_e^c) \cdot T_{c_j}^w)^{-1} \\ \cdot T_b^{e_i} \cdot \sigma(q_e^c, t_e^c) \cdot T_{c_i}^w \cdot p_{obj} - p_{obj}||\end{aligned} \tag{5}$$

In Eq. 5, q is the quaternion (qx, qy, qz, qw), t is the translation (tx, ty, tz), and $\sigma(q, t)$ represents the 4*4 matrix derived from quaternion q and translation t. The objective of this step is to optimize the quaternion and the translation. After the optimization, we could obtain the hand-eye calibration parameters. It is used for both the construct of the discrete view quality field (DVQF) and final output.

3.1.2 View sampling

As shown in Fig. 4, there are several sampling planes with different scales, and each point would contain more than one view. More specifically, the camera pose sampling is conducted in three steps:

- Starting from the original point at the calibration board, we defined a series of sampling planes. The distance between each plane is 0.1 meters. The range in x-axis and y-axis is from $-z * tan(45°)$ to $z * tan(45°)$.
- As shown in Fig. 4, the red points are camera positions in each plane. The distance between each point is 5cm on the x-axis and y-axis.
- The calibration board should be captured from more than one angle in each point to increase the variation of the whole dataset. In this case, we capture the board from different angles: 0°, ±30°, ±60° for x-axis, y-axis, or z-axis.

After the camera pose sampling, we then filter out the poses which could not view more than half of the AprilTags. Though AprilTags could be used without the full view, we still defined that more than half the board is required for accuracy. One thing that needs to be highlighted is that all

the sampling does not require any robot agent movement. The system could check each pose only by project the 3D point to the camera image and examine whether the corners are out of the view.

3.1.3 Accuracy-based evaluation.

For each view in the DVQF, we first calculate the corresponding end-effector pose ${t_p}_b^{e_i}$ through Eq. (6). Γ_b^e, Γ_c^w are the known robot pose and camera pose, T_e^c is the hand-eye calibration parameters, and t_c^w is the sampled camera pose. Then, we apply an evaluation to each pose based on two terms: *Uncertainty* and *Pose Diversity*.

$$ {t_p}_b^{e_i} = \Gamma_b^{e_i} \cdot T_e^c \cdot \Gamma_{c_i}^w \cdot (T_e^c \cdot t_c^w)^{-1} \tag{6} $$

Uncertainty. The uncertainty represents the consistency of its derivative calibration parameters. Equation 7 illustrates how to calculate the uncertainty of predicted end-effector pose ${t_p}_b^e$. We first calculate various sets of end-effector pose $({t_p}_b^e)$ based on previous robot pose (Γ_b^e) and camera pose (Γ_c^w). The standard deviations of the result poses measure the uncertainty of hand-eye calibration parameters. We tend to select poses with high standard deviations for the next step. In the equation, γ is set to 1 based on previous experiments.

$$ \begin{aligned} \phi({t_p}_b^e) = & \frac{1}{n}\sum_{i=1}^{n} || ({t_p}_b^{e_i})_q - \frac{1}{n}\sum_{j=1}^{n} ({t_p}_b^{e_j})_q ||^2 \\ & + \frac{\gamma}{n}\sum_{i=1}^{n} || ({t_p}_b^{e_i})_t - \frac{1}{n}\sum_{j=1}^{n} ({t_p}_b^{e_j})_t ||^2 \end{aligned} \tag{7} $$

Pose diversity. The calibration system requires various data from different views to maintain a high level of accuracy. As illustrated in Eq. 8 (ξ *and* μ are the thresholds of quaternion and translation), if the L1 distance between ${t_p}_b^e$ and Γ_b^e is less than a predefined threshold, a penalty would be added to the score. In this case, the predefined threshold $\xi and \mu$ is set to 0.2 and 0.22, respectively.

$$ \phi(\Gamma_b^e, {t_p}_b^e) = \begin{cases} -1, & if \quad \exists i \in (1,n), || (\Gamma_b^{e_i})_q - \frac{1}{n}\sum_{j=1}^{n} ({t_p}_b^{e_j})_q || < \xi \; and \\ & \qquad || (\Gamma_b^{e_i})_t - \frac{1}{n}\sum_{j=1}^{n} ({t_p}_b^{e_j})_t || < \mu \\ 0, & otherwise \end{cases} \tag{8} $$

Equation 9 is used to calculate each end-effector pose. It consists of two above evaluation terms: uncertainty and pose diversity. In this equation, α is a constant, and it is set to 0.09 based on our extensive experiments. Both terms are essential to the results. We had conducted an ablation study to justify their contribution, and it is presented in the next section.

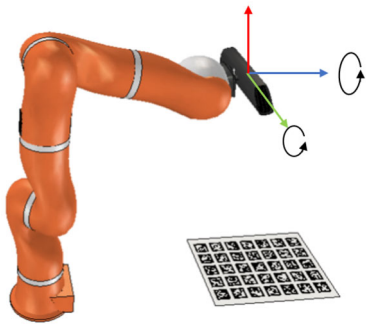


Fig. 5 An illustration of rotation in initial calibration

$$ S(\Gamma_b^e, {t_p}_b^e) = \alpha \cdot \phi({t_p}_b^e) + (1-\alpha) \cdot \phi(\Gamma_b^e, {t_p}_b^e) \tag{9} $$

3.2 Online hand-eye calibration parameters refinement

Given a sampled field of camera poses whose hand-eye calibration parameters are unknown, the objective is to evaluate each point in the field. It is impossible to select the next-best-view for calibration in advance since each view is evaluated based on the calibration parameters, which makes it a chicken-and-egg problem. We, therefore, perform an initial calibration with five sets of data to obtain the calibration parameters. We rotate the end-effector around the x-axis and y-axis in a stride of 10 degrees until it could barely see the AprilTags, as shown in Fig. 5. Then, collect the data each time and perform the computation.

The raw calibration parameters could be obtained through initial calibration but still require further refinement. More sets of data should be acquired from different poses. Therefore, the robot agent should move the end-effector to other positions. Antonello [9] propose an approach for the end-effector shifting: moving the end-effector toward the x-axis and y-axis at a fixed angle. By this strategy, a large amount of data could be collected rapidly. However, the accuracy of calibration could not reach a high level.

In contrast, our algorithm evaluates each camera pose and then selects the next-best-view (NBV) according to the DVQF for calibration. According to Eq. 9, we define the NBV as the view with the highest score. After we calculate the calibration parameters with NBV, we update the DVQF based on the current result. The whole process is conducted iteratively until it meets specific end criteria. The pseudocode is presented in Algorithm 1.

3.3 Implementation details

In order to gain high usability of the hand-eye calibration system, our system had been designed with the capability of easily extended to any type of robot, camera, and calibration

Algorithm 1 Active Handeye Calibration

Input: $iterationNumber$
Output: t_e^c \\ **Hand-eye calibration parameters**
1: $images, T_b^e \leftarrow$ INITDATASELECT \\ **Initial data select.**
2: $t_e^c \leftarrow$ CALIBRATION$(images, T_b^e)$ \\ **Initial hand-eye calibration.**
3: $(T_p)_c^w \leftarrow$ DVQFSAMPLE \\ **View Sampling.**
4: **for** $i = 0 \rightarrow iterationNumber - 1$ **do**
5: $score \leftarrow$ DVQFSCORE$((T_p)_c^w, t_e^c)$ \\ **Update DVQF.**
6: $(t_p)_c^w, (t_p)_b^e \leftarrow$ NBV$(score, (T_p)_c^w)$ \\ **Get NBV and end-effect pose**
7: $t_b^e \leftarrow$ MOVEROBOT$((t_p)_b^e)$ \\ **Robot move to specified pose**
8: $image \leftarrow$ CAPTURE
9: $T_b^e.append(t_b^e)$
10: $images.append(image)$
11: $t_e^c \leftarrow$ CALIBRATION$(images, T_b^e)$ \\ **Update calibration parameters**
12: **end for**

pattern. In extension, please note that the extended robot class must inherit the base robot class, it should be able to obtain the working space of the robot arm and control its movement. Similarly, the extended camera must inherit the base camera class, it should be able to capture the image. The extended calibration pattern must inherit the base calibration pattern class to obtain the corner on the board and perform calibration. One thing to highlight is that the calibration pattern which could tolerate partial occlusion is suggested for hand-eye calibration. Since the pattern could be easily occluded during the whole process, specific calibration patterns such as chessboard would cause an unnecessary burden.

4 Results

This section presents our experiment results based on the real-world scenarios and simulated scenarios, which have added the Gaussian noise for authenticity. In simulated scenarios, we had compared the rotation error and translation error for different methods. Moreover, we added an ablation study to justify the value of each evaluation term. In real-world scenarios, we had compared the standard deviations and density to evaluate the consistency of each method. We select [6] as the calibration method for all the experiments to justify the next-best-view selection algorithm.

4.1 Simulated scenarios

The ground truth of the hand-eye calibration result is unable to be retrieved in real-world scenarios. Therefore, we constructed a simulated scenario for comparison study. We select Vrep [10] for constructing the simulated scenarios because it is widely used and user-friendly. As shown in Fig. 6, we select Kinect-DK camera and simulate both eye-in-hand and eye-to-hand contexts.

By adding the Gaussian noise to the robot poses and the camera intrinsic parameters, we had construct scenarios with verisimilitude. The end-effector rotation is represented in Euler angles, which had added the Gaussian noise with $\delta = 0.002$. The translation part had been conducted in the same way. Since the camera's intrinsic parameter could be slightly different from the ground truth, we also add the Gaussian noise to the parameters in the simulated scenario. We had add the Gaussian noise with $\delta = 10$ to the fx and fy, and $\delta = 1$ to the cx and cy.

We had implemented six hand-eye calibration methods and compared them with ours in the simulated scenario. By calculating the difference with ground truth, our method could reduce the rotation error by 7%, and translation error by 5% (shown in Table 1). Our approach shows better performance with reasonable noise.

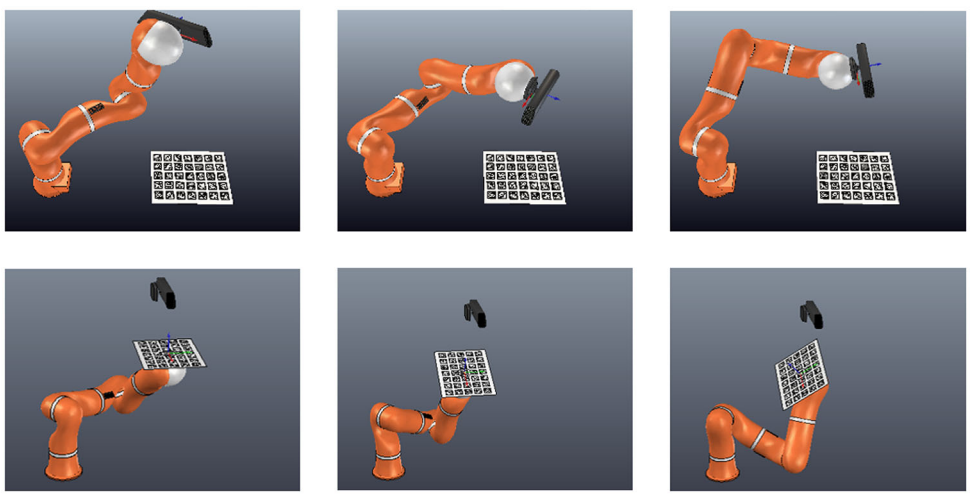
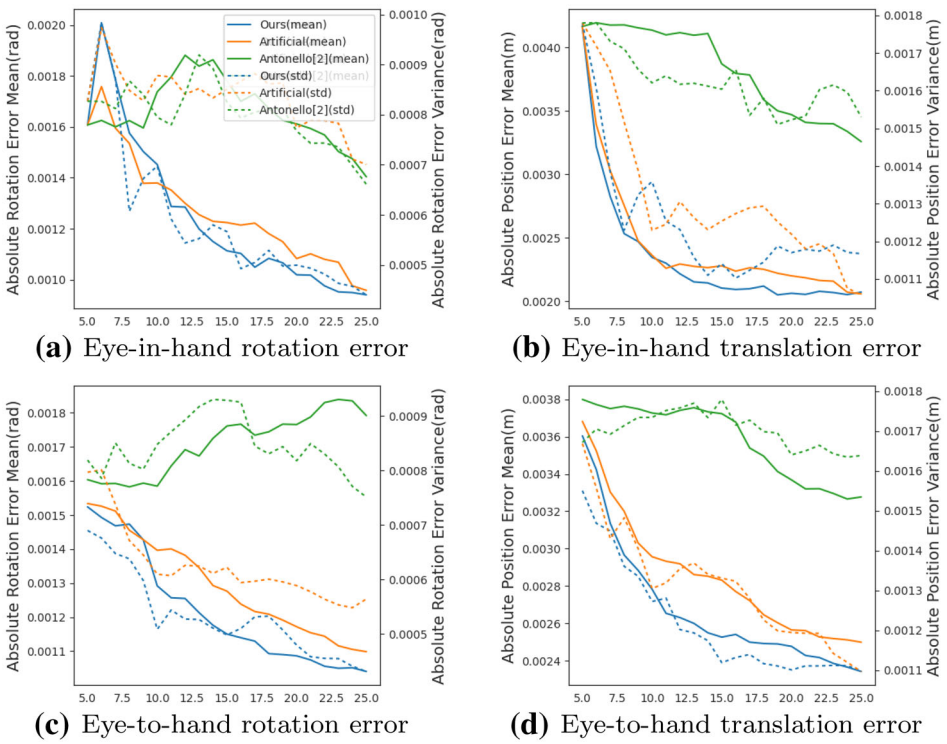


Fig. 6 Samples of simulated scenarios in the context of eye-in-hand (top row) and eye-to-hand (bottom row)

Table 1 Comparison of methods in simulated scenarios

	Mean rotation error(rad)	Mean translation error(m)
Tsai [4]	0.001202639	0.002285711
Dekel Amit [6]	0.001223916	0.002280357
Horaud and Dornaika [13]	0.001308097	0.002384463
Park and Martin [22]	0.001229095	0.002294825
Li et al. [14]	0.001223547	0.002212508
Shah [7]	0.001233155	0.002279889
Our	**0.001113143**	**0.002104458**



(a) Eye-in-hand rotation error **(b)** Eye-in-hand translation error

(c) Eye-to-hand rotation error **(d)** Eye-to-hand translation error

Fig. 7 Comparison of rotation error and translation error between three approaches in the context of eye-in-hand (top row) and eye-to-hand (bottom row)

Table 2 The error of the calibration results in different evaluation terms

	rx(rad)	ry(rad)	rz(rad)	tx(m)	ty(m)	tz(m)
Uncertainty	0.001985	**0.001504**	0.002495	0.001564	0.001056	0.001906
Pose diversity	0.001984	0.002569	0.001875	0.001166	0.001665	0.001845
Uncertainty & Pose diversity	**0.001593**	0.001567	**0.001847**	**0.00099**	**0.0008598**	**0.001582**

We had compared the result of two methods in different types of robot movement with our algorithm and utilized the same calibration algorithm. The first method is choosing the camera poses randomly, aiming to imitate manual hand-eye calibration. The second one is a regular automatic method based on [9]. We had to test a hundred sets of data for each approach. As illustrated in Fig. 7, the solid line shows the mean rotation error of the calibration parameters, and the dotted line shows the standard deviation of the calibration parameters. The general trend of the three method's result is to become more accurate and consistent with the increase of the iteration numbers. For application, numerous iteration is inefficient. Our method shows much higher accuracy at the end of reasonable times of iteration.

We had listed the result of the ablation study in Table 2. Each column contains the component of the calibration results. It could be seen from the table that when combining the uncertainty and pose diversity, the algorithm shows

Fig. 8 A capture of the real-world scenario. A Kinect-DK camera is mounted on the end-effector (orange box) to view the AprilTag (red box)

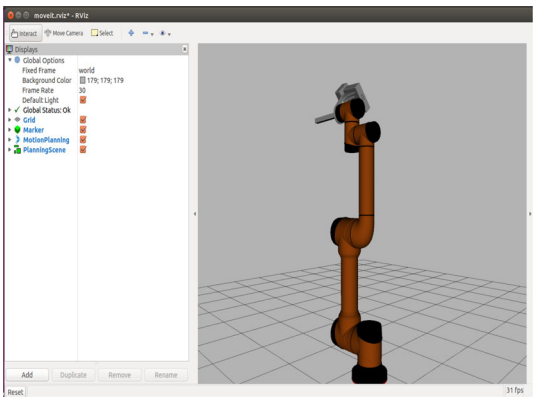


Fig. 9 A screenshot of graphical user interface of ROS platform

the highest accuracy. Therefore, we had concluded that both terms are essential for evaluating each view in the DVQF.

4.2 Real-world scenarios

We select Aubo-i5 for experiments in real-world scenarios (Fig. 8). It has a high level of locating accuracy (0.5 mm) and an 886.5-mm maximum working radius. It could reach all the points in the sampled field. Moreover, it supports ROS [23], a widely used platform for robot control. Figure 9 shows our interface supported by ROS. Collision avoidance and target reachability are guaranteed by the MoveIt! library, which could be easily integrated into the platform.

As illustrated in Fig. 8, we had set up an Aubo-i5 robot arm with a Kinect-DK camera mounted on the end-effector. The distance from AprilTags and the robot arm is set to 40 cm. The tag size is 3.033 cm, and the distance between every two tags is 0.776 cm.

We select standard deviations to reflect the accuracy level of the hand-eye calibration method since the lack of ground truth. We had collected 20 sets of data, each contains 30 pairs of camera poses and end-effector poses. We had compared our method with six previous methods [4,6,7,13,14,22]. As shown in Table 3, our method performs the best in the comparison.

We had conducted a series of comparisons of robot movement between our method, Antonello's [9] and manual calibration. We selected four planes with a distance of 0.4 m, 0.5 m, 0.6 m, and 0.7 m to the calibration board. The camera poses for both our algorithm and Antonello's are sampled from these four planes to reduce the variable. We had collected 20 sets of data, each contains 30 pairs of camera poses and end-effector poses. Figure 10 shows the standard deviations of each hand-eye calibration parameter with the increased amount of input data. It indicates that based on our algorithm, the calibration parameter would convergence most after a certain amount (30 pairs) of pose sampling. Figure 11 shows the kernel density estimation of six calibration parameters in three methods when input 25 pairs of data. The mean value is close, but our method shows a more concentrated distribution, which indicates higher robustness. Figure 12 shows the pose selection and online update of the DVQF in the hand-in-eye context. Each point is evaluated due to its accuracy and validness.

Table 3 Comparison of methods in real-world scenarios

	rx(rad)	ry(rad)	rz(rad)	tx(m)	ty(m)	tz(m)
Tsai [4]	0.002042	0.002807	0.007009	0.003621	0.003514	0.005210
Dekel Amit [6]	0.001891	0.003045	0.004925	0.002786	0.003931	0.005162
Horaud and Dornaika [13]	0.002402	0.001529	0.003769	0.003665	0.006822	0.007176
Park and Martin [22]	0.001195	0.001819	0.002424	0.002092	0.003207	0.005061
Li et al. [14]	**0.001149**	0.001780	0.002274	0.001242	0.001668	0.003019
Shah [7]	0.001170	0.001823	0.002479	0.001952	0.003654	0.005334
Our	0.001652	**0.001123**	**0.001061**	**0.000669**	**0.000918**	**0.000849**

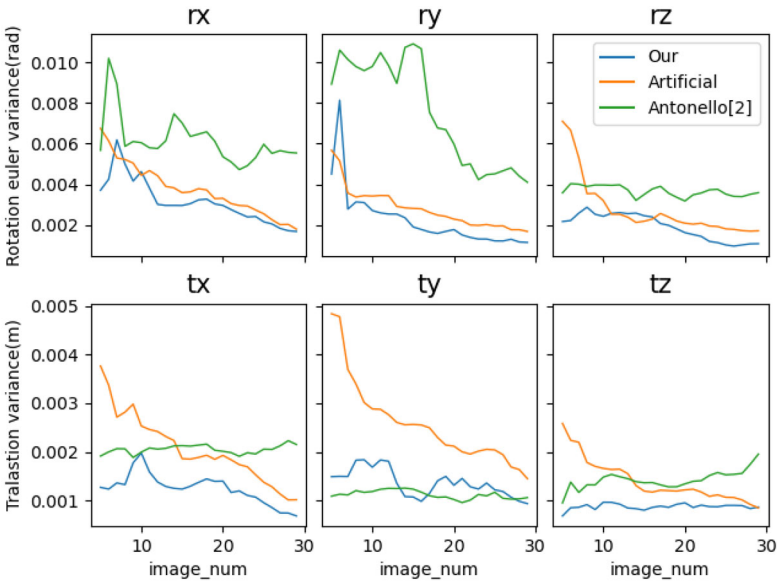
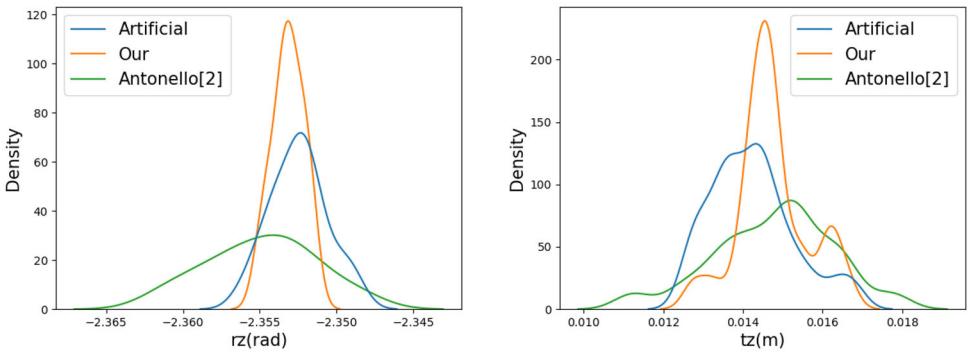


Fig. 10 A comparison of six calibration parameters' rotation and transition standard deviations for three approaches. With the input image increases, our algorithm shows the greatest consistency: the rotation term error is lower than 0.002 rad, the translation term is lower than 0.001 m



(a) Simulate hand-in-eye error **(b)** Simulate hand-to-eye error

Fig. 11 A comparison of two most representative calibration parameters rz (left) and tz (right) between three approaches. Our algorithm shows concentrated distribution

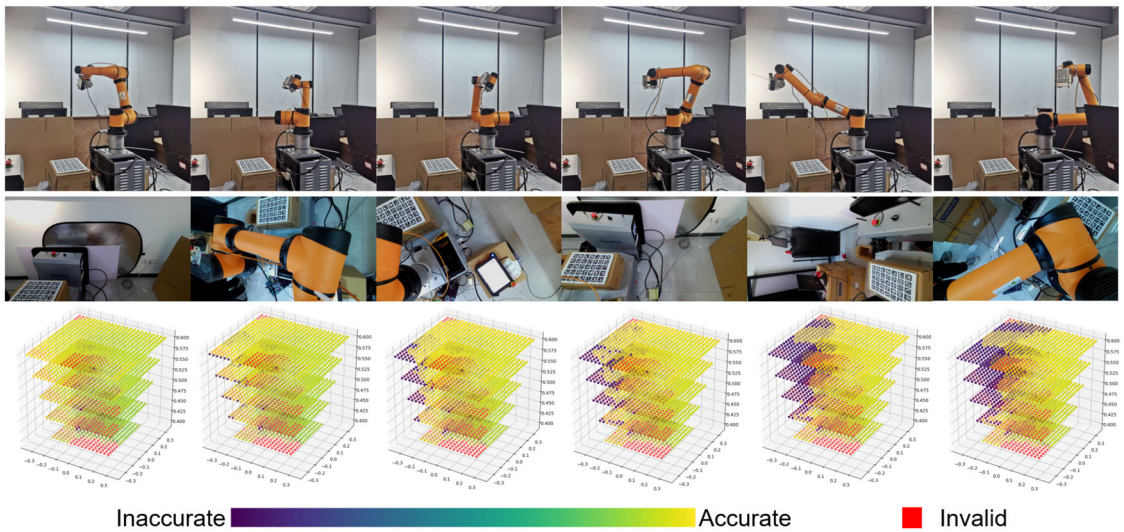


Fig. 12 An illustration of the online accuracy-driven next-best-view (NBV) selection in the hand-in-eye context. Each column represents a single update during the whole process, including the movement of the robot arm (top row), the view of the mounted camera (middle row), visualization of the DVQF (bottom row). Note that each point represents the most accurate view in that position. Some positions are invalid because they could not capture the calibration pattern or are unreachable for the robot arm

5 Conclusion

In this paper, we present an active hand-eye calibration system with high accuracy. We propose a discrete view quality field (DVQF) for calibration. Each point in the field represents a camera pose evaluated from two aspects: uncertainty and pose diversity. Then, select the next-best-view (NBV) and compute the corresponding robot pose for calibration. We provide a comparison of camera pose sampling between our algorithm, existing algorithm, and manual sampling in simulated scenarios and real-world scenarios. The future work would aim to represent the uncertainty of the camera pose in a better form and reduce it.

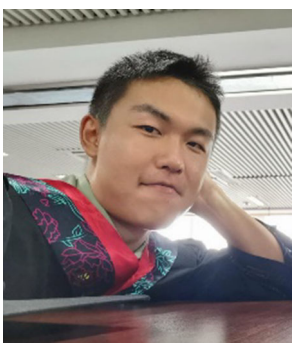
Funding This work was supported by the National Key Research and Development Program of China (No. 2018AAA0102200).

Declaration

Conflict of interest Xinxin Zhang declares that he has no conflict of interest. Zhentao Huang declares that he has no conflict of interest. Lintao Zheng declares that he has no conflict of interest. Kai Xu declares that he has no conflict of interest.

References

1. Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int. J. Robot. Res. **37**(4–5), 421 (2018)
2. Pachtrachai, K., Allan, M., Pawar, V., Hailes, S., Stoyanov, D.: In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE, 2016), pp. 2485–2491
3. Chai, X., Wen, F., Cao, X., Yuan, K.: In: 2013 IEEE International Conference on Mechatronics and Automation (IEEE, 2013), pp. 57–62
4. Tsai, R.Y., Lenz, R.K., et al.: A new technique for fully autonomous and efficient 3 D robotics hand/eye calibration. IEEE Trans. Robot. Autom. **5**(3), 345 (1989)
5. Flandin, G., Chaumette, F., Marchand, E.: In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), vol. 3 (IEEE, 2000), vol. 3, pp. 2741–2746
6. Dekel, A., Harenstam-Nielsen, L., Caccamo, S.: Optimal least-squares solution to the hand-eye calibration problem. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020), pp. 13598–13606
7. Shah, M.: Solving the robot-world/hand-eye calibration problem using the Kronecker product. J. Mech. Robot. **5**(3) (2013)
8. Shiu, Y.C., Ahmad, S.: Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX= XB. IEEE Trans. Robot. Autom. **5**(1), 16 (1989)
9. Antonello, M., Gobbi, A., Michieletto, S., Ghidoni, S., Menegatti, E.: In: 2017 European Conference on Mobile Robots (ECMR) (IEEE, 2017), pp. 1–6
10. Freese, M.: V-rep. https://www.coppeliarobotics.com/ (2019)
11. Horaud, R., Dornaika, F.: Hand-eye calibration. Int. J. Robot. Res. **14**(3), 195 (1995)
12. Zhuang, H., Roth, Z.S., Sudhakar, R.: Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX= YB. IEEE Trans. Robot. Autom. **10**(4), 549 (1994)
13. Dornaika, F., Horaud, R.: Simultaneous robot-world and hand-eye calibration. IEEE Trans. Robot. Autom. **14**(4), 617 (1998)
14. Li, A., Wang, L., Wu, D.: Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product. Int. J. Phys. Sci. **5**(10), 1530 (2010)
15. Ali, I., Suominen, O., Gotchev, A., Morales, E.R.: Methods for simultaneous robot-world-hand-eye calibration: a comparative study. Sensors **19**(12), 2837 (2019)
16. Tabb, A., Yousef, K.M.A.: Solving the robot-world hand-eye (s) calibration problem with iterative methods. Mach. Vis. Appl. **28**(5), 569 (2017)
17. Schmidt, J., Vogt, F., Niemann, H.: In: Joint Pattern Recognition Symposium (2005)
18. Pachtrachai, K., Allan, M., Pawar, V., Hailes, S., Stoyanov, D.: In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
19. Wang, J., Olson, E.: In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE, 2016), pp. 4193–4198
20. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the PNP problem. Int. J. Comput. Vis. **81**(2), 155 (2009)
21. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ, Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P.: SciPy 1.0 contributors fundamental algorithms for scientific computing in Python. Nat. Methods **17**, 261 (2020). https://doi.org/10.1038/s41592-019-0686-2
22. Park, F.C., Martin, B.J.: Robot sensor calibration: solving AX=XB on the Euclidean group. IEEE Trans. Robot. Autom. **10**(5), 717 (2002)
23. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: In: ICRA workshop on open source software, vol. 3 (Kobe, Japan, 2009), vol. 3, p. 5



Xinxin Zhang is a masters student at the School of Computer Science, National University of Defense Technology (NUDT), supervised by Yueshan Xiong and Kai Xu. He received his bachelors degree in computer science from NUDT. His research interests include computer vision and computer graphics.

Yuefeng Xi is a masters student at the School of Computer Science, National University of Defense Technology (NUDT), supervised by Kai Xu. His research interests include 3D computer vision and robot navigation.

Zhentao Huang is an undergraduate student at the School of Computer Science, University of Nottingham Ningbo China (UNNC), supervised by Kai Xu. His re search interests include computer vision and human -computer interaction.

Lintao Zheng is an Assistant Professor at College of Meteorology and Oceanography, National University of Defense Technology (NUDT). He received his PhD in computer science in 2019 from National University of Defense Technology.

Hui Huang is a Distinguished TFA Professor at Shenzhen University, where she directs the Visual Computing Research Center. Her resear ch interests span computer graphics, 3D vision, and visualization. She is currently a Senior Member of IEEE/ACM/CSIG and a Distinguished Member of CCF.

Yueshan Xiong is a Professor at the School of Computer, National University of Defense Technology (NUDT). The current directions of interest include virtual surgery system, image and graphics processing and intelligent computing.

Kai Xu is a Professor at the School of Computer, National University of Defense Technology, where he received his PhD in 2011. He is currently an Adjunct Professor of Simon Fraser University. He was a Visiting Scholar at Princeton University during 2017 -2018. His research interests include geometric modeling and shape analysis, especially on data-driven approaches to the problems in those directions, as well as 3D vision and its robotic applications. He has published 100+ papers, including 20+ SIGGRAPH/TOG papers. He co-organized several SIGGRAPH Asia courses, CVPR tutorials and Eurographics STARs. He serves on the editorial board of ACM Transactions on Graphics, Computer Graphics Forum, Computers & Graphics, and The Visual Computer. He also served as program co-chair of CAD/Graphics 2017, ICVRV 2017 and ISVC 2018, as well as PC member for several prestigious conferences including SIGGRAPH, SIGGRAPH Asia, Eurographics, SGP, PG, etc. His research work can be found in his personal website: www.kevinkaixu.net.