2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Markerless Hand-Eye Calibration by Flange Ellipse Detection

Ruoyu Jia[1], Ruomeng Fan[2], Qitong Guo[1], Xiaohang Shi[1], Masahiro Hirano[3] and Yuji Yamakawa[4]

***Abstract*— This paper proposes a simple yet effective markerless hand-eye calibration method that achieves low cost, high accuracy, and strong generalization across different types of robots. The method utilizes a circular flange, a standardized structure in industrial robots, for calibration via the perspective-n-point (PnP) algorithm, achieving superior performance with a simpler pipeline. The entire system is built using mature, off-the-shelf components, avoiding complex architectures. By combining a lightweight object detection network (e.g., Faster R-CNN) with classical geometric techniques, we construct a flange detector that is both accurate and robust. The training process requires no manual annotations, and the resulting model generalizes well across various robot platforms. Experiments demonstrate that our method achieves higher calibration accuracy than more complex existing approaches. Notably, the method maintains consistent precision even when applied to previously unseen robots. All developments are available at: `https://github.com/Ruoyu-Jia/Markerless_hand_eye`.**

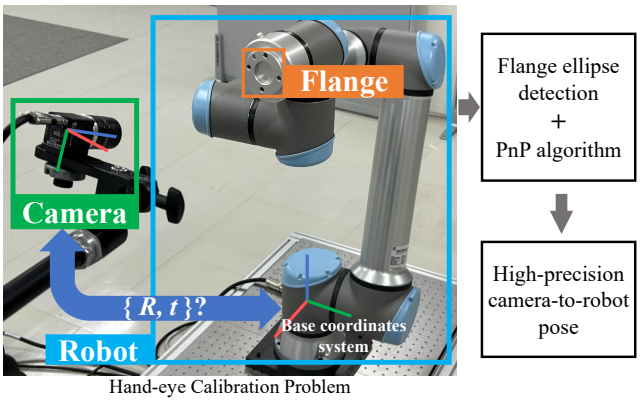

Fig. 1. Overview of the proposed markerless hand-eye calibration method. The framework estimates the camera-to-robot pose by detecting the flange ellipse and PnP problem solving, enabling accurate calibration without the need for fiducial markers.

I. Introduction

With advances in computer vision and artificial intelligence, robots equipped with external cameras are capable of performing a wide range of unstructured tasks, including object grasping [1], automatic assembly [2], and human-robot interaction [3]. In such scenarios, accurately estimating the pose between the robot and the camera, known as hand-eye calibration, is essential for transforming visual measurements into the robot's task space and enabling effective control.

Conventionally, hand-eye calibration is performed by using a fiducial marker [4] mounted on the robot's flange to capture a series of corresponding camera and robot poses. A non-linear optimization process is then used to estimate the unknown camera-to-robot pose [5]–[8]. Although accurate, these methods suffer from practical limitations: high-quality markers tend to be expensive and lack portability, and the calibration process demands careful manipulation by trained operators to maintain marker visibility and avoid robot collisions. Such requirements hinder usability and scalability, especially in non-laboratory settings.

To overcome these challenges and improve usability, markerless calibration methods that do not rely on fiducial markers have received increasing attention. A common approach is to incorporate deep learning to detect keypoints on the robot arm, followed by computing the camera pose using the Perspective-n-Point (PnP) algorithm [9]–[13]. While effective in constrained settings, the performance of these methods heavily depends on the quality of the training data. Moreover, these models are typically tailored to specific robot types, and adapting them to new robots requires retraining with newly collected data, resulting in high cost and poor scalability.

Recent works [14]–[16] employ differentiable rendering with pixel-wise mask loss to optimize the camera pose, which mitigates the impact of prediction errors and leads to improved calibration accuracy. By replacing keypoint detection with robot-arm segmentation, these methods enable the use of large models with stronger generalization. However, the optimization is computationally intensive and requires high-performance GPUs, limiting their applicability in resource-constrained settings.

In this paper, we propose a lightweight and robust hand-eye calibration framework with the following advantages:

- High-precision and fully automatic calibration without manual intervention.
- Strong generalization capability, seamlessly applicable to any robot arm with an ISO-compliant [17] flange.
- Extremely simple setup, with no need for GPU acceleration or CAD model priors.

The overview of the proposed method is shown in Fig. 1. The key motivation of our method is that the robot flange, with its standardized structure, provides a more stable and robust reference for pose estimation than robot-body-keypoints. Based on this, we replace the robot-body-keypoints with the flange center in the conventional PnP-based framework,

This work was supported by JST SPRING, Grant Number JPMJSP2108.
[1]Ruoyu Jia *(corresponding author)*, Qitong Guo, Xiaohang Shi are with Graduated School of Engineering, the University of Tokyo, Tokyo, Japan {jiaruoyu, guoqt, sxh}@iis.u-tokyo.ac.jp
[2]Ruomeng Fan is with Robot Learning Lab, Imperial College London, London, United Kingdom r.fan24@imperial.ac.uk
[3]Masahiro Hirano is with the Institute of Industrial Science, the University of Tokyo, Tokyo, Japan mhirano@iis.u-tokyo.ac.jp
[4]Yuji Yamakawa is with Interfaculty Initiative in Information Studies, the University of Tokyo, Tokyo, Japan y-ymkw@iis.u-tokyo.ac.jp

which effectively improves both accuracy and generalization. Leveraging the flanges distinct features and regular contour, we detect its center using both neural network inference and traditional ellipse fitting. By cross-validating the results, we obtain more accurate and stable 2-D points than using neural network alone. Owing to the ISO [17] standardization of flange geometry, the trained model can be directly applied across different robot platforms without any additional training or prior setup. In terms of pipeline design, we aim to reduce user burden and improve modularity by employing lightweight and reliable components. The entire system requires neither CAD-based rendering nor large neural networks, and all computations can be performed efficiently without GPU acceleration.

Experiments were carried out to evaluate the performance of our method. In comparative experiments with the existing classic and state-of-the-art methods, our framework achieves superior accuracy while maintaining efficient performance on low-cost hardware. Moreover, the calibration results exhibit consistent precision when applying the trained model on previous unseen robots, which proves the great generalization capability of our method.

II. Related Work

The implementation of low-cost, high-precision markerless hand-eye calibration without sacrificing generality has been challenging. Lu et al. [18] proposed a monocular hand-eye calibration method using Plücker coordinates and longitudinal-axis mapping, but its design tailored for surgical applications limits generality. Zhong et al. [19] introduced an iterative markerless approach applicable to both medical and industrial robots; however, it relies on manually selected initial points for optical flow, making it less accessible to non-experts. In some studies [20], [21], high-precision 3-D scanners have been employed to directly capture pose information of the flange, eliminating the need for fiducial markers. However, the high cost and limited applicability of these scanners contradict the original goal of cost reduction in markerless hand-eye calibration.

In recent years, learning-based methods have shown growing potential. Lambrecht et al. [10] proposed a method that utilized a dataset including real and synthetic data to train a network for identifying key points on robots, followed by PnP solving [9] for calibration. Lee et al. proposed DREAM [11], a framework that can effectively generate a large number of synthetic datasets for 2-D key point detection network through virtual environment. Building upon 2-D key point detection and PnP solving, a series of methods were proposed [12], [13] to enhance the precision and robustness of key point detectors. Despite the operational convenience offered by these methods, challenges remain in terms of accuracy and generalization capability.

To address these issues, differentiable rendering has recently emerged as a promising tool for optimizing camera poses by aligning rendered and observed masks. Lu et al [14] proposed a method which can recover camera's pose from multi-perspective, and is well-suited for perception tasks involving unknown or flexible robot structures. Aiming at high-precision markerless calibration, Chen, Hong et al. proposed EasyHeC [15] and its improved version EasyHeC++ [16]. EasyHeC employs PointRend [22] to automatically segment the robot mask, and combines differentiable rendering with consistency-driven joint space exploration to achieve high-precision hand-eye calibration by aligning rendered and observed masks at the pixel level. Building on this, EasyHeC++ introduces DKM [23] and AutoSAM [24], two large pre-trained modules that can carry out initial pose estimation and extract robot masks without additional training, significantly improving the methods generalization capability and zero-shot transferability. However, these improvements come at the cost of high computational requirements, making the methods less suitable for resource-constrained scenarios.

III. Methodology

A. Overview

We define a calibration scenario that involves a robotic arm with an external camera fixed in place, as shown in Fig. 1. A circular flange is at the end of the robotic arm, with no additional end-effector attached. Since robots with such flange designs are widely utilized owing to their compliance with ISO standards [17], the scenario we define is common.

Similar to existing methods [10]–[12], our framework estimates the camera-to-robot pose using the PnP algorithm, based on 3-D points obtained via robot proprioception and 2-D flange centers in the image. The 2-D targets are detected through both neural network inference and ellipse fitting, two fundamentally different approaches whose results can be cross-validated to improve reliability. Specifically, an **IoU** filter is employed to implement this cross-validation and ensure high precision. The entire process of our framework is shown in Fig. 2.

B. Training Pipeline

1) Data Collection and Annotation: To reduce manual annotation costs, we propose an automatic pipeline for building flange detection datasets. A thin marker, which can be fabricated by either 3-D printing or machining, is designed for labeling, as shown in Fig. 3. The pattern on the tool comprises a circle and an ArUco marker [25] that share the same center. The ArUco enables precise localization, while the circular contour is designed to match the flanges outer diameter. With a thickness of less than 0.5 mm, the pattern' outer edge circle can be roughly aligned with flange's outer edge when the marker is mounted on the flange. It should be noted that the marker is introduced solely for automatic annotation in dataset construction; the calibration stage itself is fully marker-free.

To collect the training data, the robot is first moved through a series of poses within the cameras field of view. At each pose, an image of the flange without any marker is captured, as shown in Fig. 2(a). After completing this first round of image acquisition, the marker is attached to the flange. The robot then revisits the same sequence of poses,

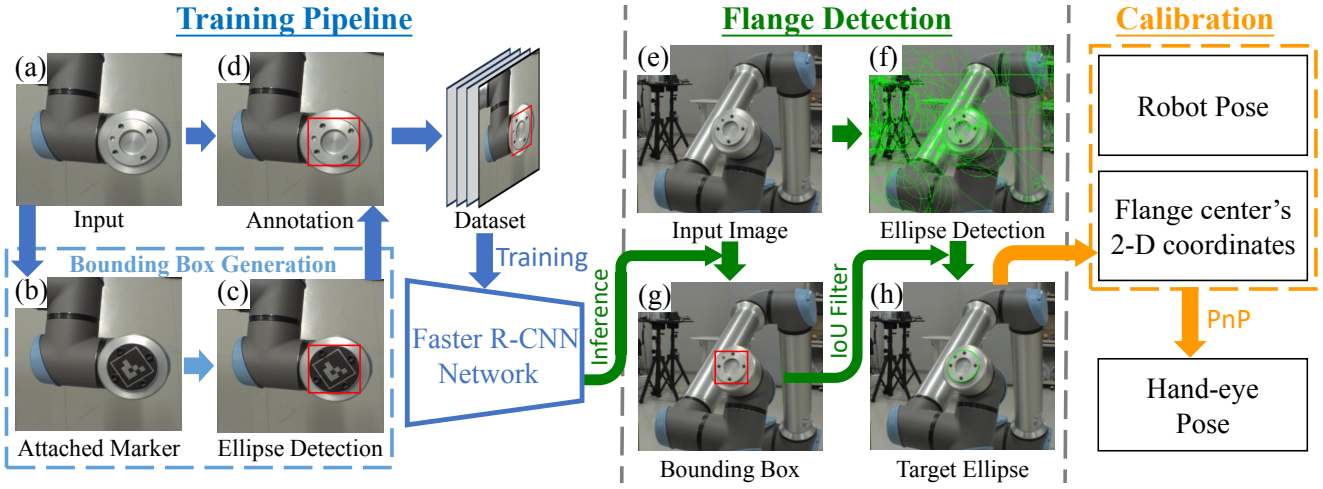

Fig. 2. Flow of the proposed method. (a) Flange image to be annotated. (b) Flange image with a marker attached when the robot is in the same pose. (c) Bounding box obtained by detecting the marker. (d) Image annotated by the bounding box. (e) Input image of the flange detection process. (f) Potential ellipses identified through ellipse detection. (g) Bounding box inferred through trained network. (e) Target ellipse obtained through **IoU** filter.

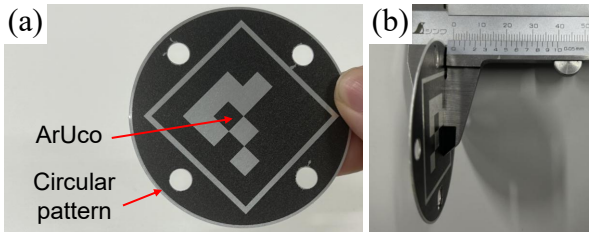

Fig. 3. The marker utilized in the proposed method: (a) The components of the pattern on the marker ; (b) The thickness (0.5mm) of the marker.

capturing a second set of images, as shown in Fig. 2(b). Owing to the robots high repeatability, each pair of images taken at the same pose is geometrically aligned. Therefore, the marker-containing image serves as a reference to localize the flange, enabling automatic annotation of its counterpart without the marker. Specifically, we apply ellipse detection to the image with the marker and use the ArUco center to create a bounding box $\mathcal{B}(E_l)$ corresponding to the flanges outer contour, as shown in Fig.2(c). This bounding box is then transferred to the corresponding image without the marker, allowing automatic annotation of the flange, as shown in Fig.2(d). The obtaining process of $\mathcal{B}(E_l)$ is as follows: we first apply ellipse detection and extract the centers p_E^i of all detected ellipses. The ArUco is also detected, yielding its center p_A. We then filter the ellipses by retaining those whose centers lie within a distance threshold thr_c of p_A:

$$|p_E^i - p_A| < thr_c. \tag{1}$$

In most cases, this step isolates the correct ellipse. However, when the flange faces the camera directly, other concentric contours, such as chamfers or nearby structures, may also satisfy this criterion. To handle such cases, we select the smallest ellipse among the candidates as the target. The bounding box $\mathcal{B}(E_l)$ is then constructed by sampling points along the ellipse and taking their extreme x and y coordinates. This box is transferred to the image without the marker to complete annotation.

2) Network Architecture: Utilizing the data obtained through the aforementioned method, a convolutional neural network is trained to identify and locate the robot flange in images. In this study, Faster R-CNN [26] was chosen for object detection owing to its widespread use and availability of pre-trained models. Faster R-CNN assumes an RGB image of size $w \times h \times 3$ as input, and the output of the network is a single bounding box, denoted as $\mathcal{B}(F_D)$.

C. Flange Detection and Pose estimation

Although the flange center can be roughly estimated from the predicted bounding box $\mathcal{B}(F_D)$, achieving high-precision PnP calibration based on such predictions is challenging due to their limited robustness and potential bias. To address this, we propose an **IoU**-based filtering strategy to refine the 2-D position of the flange center with high accuracy.

Specifically, we apply an ellipse detection method to identify all potential ellipses in the image using a combination of Hough transform and fitting techniques. An example result is shown in Fig. 2(f), where the flange's outer contour is detected along with other candidates. For each detected ellipse E^i, similar to the annotation, we generate the bounding boxes $\mathcal{B}(E^i)$. Hence, the **IoU** between each $\mathcal{B}(E^i)$ and $\mathcal{B}(F_D)$ can be calculated as follows:

$$\mathbf{IoU}^i = \frac{\mathcal{B}(E^i) \cap \mathcal{B}(F_D)}{\mathcal{B}(E^i) \cup \mathcal{B}(F_D)}. \tag{2}$$

Since $\mathcal{B}(F_D)$ represents the inference result of the flange, the ellipse with the highest $\mathbf{IoU}^i$ is identified as our target ellipse, as shown in Fig. 2(h). The center of it is denoted as p_C, with its corresponding **IoU** labeled as $\mathbf{IoU}^{max}$.

During data collection, the robot was moved to multiple poses while ensuring sufficient flange visibility in the camera view, and a flange image was captured at each pose. For each image, the maximum $\mathbf{IoU}^{max}$ and the corresponding

p_C were computed. Samples were retained only if:

$$\mathbf{IoU}^{max} > thr_{\mathbf{IoU}}, \tag{3}$$

where $thr_{\mathbf{IoU}}$ is a manually defined threshold. Since (3) is met only when both the Faster R-CNN and ellipse detection are accurate, it effectively filters out erroneous samples from either source, ensuring that only reliable data contribute to the PnP solving.

Hence, utilizing the 2-D coordinates p_C and their corresponding 3-D coordinates $\mathbf{P}_i$, the camera-to-robot pose can be estimated by solving the PnP problem:

$$p_{Ci} = K(\mathbf{R}_b^c \mathbf{P}_i + \mathbf{t}_b^c), \tag{4}$$

where K denotes the camera intrinsic, $\mathbf{R}_b^c$ and $\mathbf{t}_b^c$ represent the rotation and translation part of camera-to-robot pose, respectively. In this study, the RANSAC-PnP method [27] was employed to eliminate outliers in the data, resulting in a more robust pose estimation.

IV. Evaluation

This section is organized as follows. In Section IV-A, we provide a detailed explanation of the experimental setup. Sections IV-B and IV-C evaluate the accuracy of flange detection and analyze key influencing factors. In Section IV-D, we evaluate the proposed methods precision in hand-eye calibration and compare its performance to that of existing approach. Finally, in Section IV-E, we investigates the generalization capability of the trained models.

A. Experimental Setup

1) Facilities: As shown in Fig.1 and Fig.4, we conducted experiments using five robots: UR10e, UR5e, UR3e, Franka Emika Panda, and Fairino FR5, all of which are equipped with 63mm ISO-standard flanges.

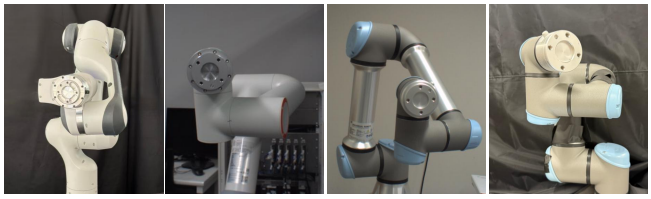

Fig. 4. Other robots used in the experiments: (a) Franka Emika Panda; (b) Fairino FR5; (c) UR5e; (d) UR3e.

2) Datasets collection and network training: The experimental datasets were collected using the UR10e robot for training the flange detection network. The XIMEA xiQ camera, with a resolution of 1290×1024 pixels, was used to capture images for experiments. A total of 1,119 image pairs, as illustrated in Fig. 2(a) and Fig. 2(b), were collected to construct the datasets. To increase data diversity and mitigate potential overfitting, the images were captured from three distinct backgrounds and viewpoints. For training, we employed the *MMDetection* [28] and utilized its pre-trained model to construct and train the Faster R-CNN network. The inference process was carried out on an Intel i7-12700H CPU Laptop without GPU acceleration.

B. Precision of Flange Detection

The precision of flange detection significantly impacts the outcome of the PnP solving. We utilized the marker in Fig. 3 to measure the error. Experimental data was collected by controlling the UR10e robot in the following manner: Initially, the robot was manually positioned to ensure that the flange was centered in the image, with the flange and optical axis of the cameras approximately perpendicular. Gaussian noise was then introduced to the flange's pose, where the translation standard deviation σ_t was set to 0.3 m, and for rotation, σ_r was set to 30 degrees. At each pose, two images were captured: one with the marker mounted on the flange and another with the marker unmounted. The 2-D coordinates obtained from the ArUco marker were utilized as the ground truth, and the error in flange detection was calculated accordingly. A total of 531 pairs of 2-D coordinate data were analyzed for errors, and the results are shown in Fig. 5. Each point in the figure represents a flange detection result for a data instance, with the horizontal axis indicating the $\mathbf{IoU}^{max}$ achieved during detection, and the vertical axis representing the error of the extracted ellipse center.

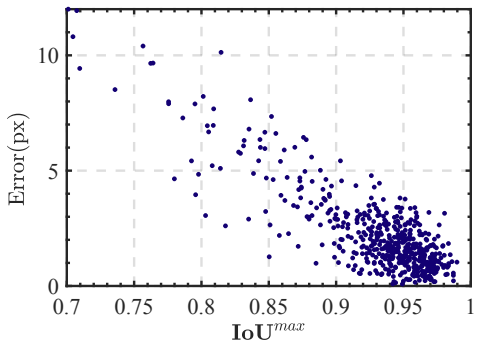

Fig. 5. Relationship between flange detection error and $\mathbf{IoU}^{max}$

The figure shows that as $\mathbf{IoU}^{max}$ increases, the error in the flange detection remains within a narrow range, indicating that $\mathbf{IoU}^{max}$ can effectively filter out low-precision 2-D point coordinates. Meanwhile, although the proportion of high-precision data is significantly high, a non-negligible amount of low-precision data with large errors still exists. This suggests that a considerable time is spent on collecting low-precision data. Therefore, to improve the efficiency of data collection, it is important to examine the factors that influence $\mathbf{IoU}^{max}$.

C. Factors influencing IoU

Two factors were considered as potential influences on the percentage of data with higher $\mathbf{IoU}^{max}$: the angle between the flange and camera axes, as well as the size of the flange in each image. Experiments were conducted to quantitatively validate these influences.

1) Angle Between Flange and Camera Axes: To investigate the impact of the angle between the flange and camera axes, the experiment was structured as follows: First, the robot was controlled to position its flange at the center of the camera's field of view, with the flange and camera's optical

axis nearly perpendicular. Gaussian-based disturbances were then introduced to the robots fourth and fifth joints, causing the robot to assume a new pose. At each pose, two images were captured: one with the marker (Fig. 3) attached, and one without it. Through flange detection, the datas $\mathbf{IoU}^{max}$ was obtained, whereas the corresponding angle between the flange and camera axes was determined by analyzing the ArUco marker. A total of 1800 samples were collected, with the outcomes shown in Fig. 6.

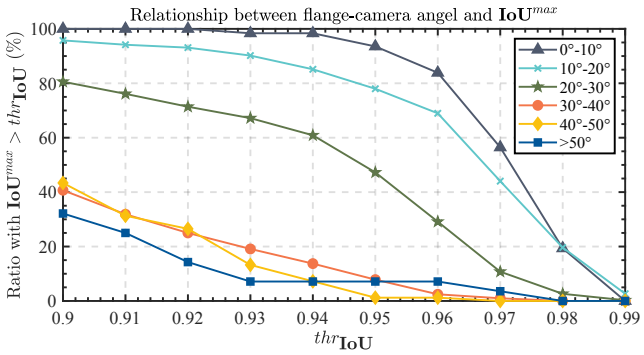

Fig. 6. Relationship between flange-camera and $\mathbf{IoU}^{max}$

The horizontal axis represents the threshold $thr_{\mathbf{IoU}}$ set for filtering data with high $\mathbf{IoU}^{max}$, whereas the vertical axis represents the percentage of data that satisfy this threshold relative to the total data. The various lines in the chart correspond to the ranges of angles associated with the data. As shown in the chart, the probability of obtaining high $\mathbf{IoU}^{max}$ data demonstrates a noticeable downward trend as the angle increases. Therefore, we recommend keeping the angle between the camera and flange below 20 degrees during data collection to ensure optimal efficiency.

2) Flange's Size in Image: Another experiment was conducted to validate the effects of flange size in the image. The same methodology outlined in Section IV-B was utilized to gather experimental data, with the rotation standard deviation σ_r set at 0 degrees to eliminate the impact of the flange-camera angel. We leveraged perspective effects by controlling the distance between the flange and the camera to vary the flanges size in the image. Two initial poses were set at different distances from the camera, with $\sigma_t = 0.1$m for the closer pose and $\sigma_t = 0.4$m for the farther one. For each set of data, the proposed flange detection method was applied, and the $\mathbf{IoU}^{max}$ and length of the long side of the bounding box were recorded. A total of 1700 samples were collected, with the experimental findings shown in Fig. 7.

The horizontal and vertical axes carry the same significance as in Fig. 6, whereas the different lines represent data categorized by the size of the bounding box's long side. Notably, as the bounding box area decreased, the $\mathbf{IoU}^{max}$ demonstrated a significant downward trend, particularly when the bounding box size is below 100 pixels. Therefore, the flange should not be positioned too far from the camera during data collection to prevent its image size from becoming too small.

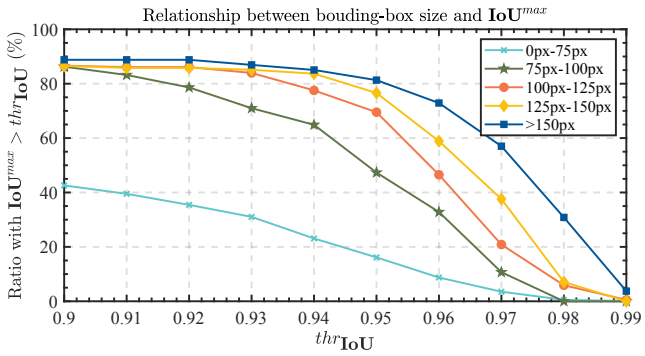

Fig. 7. Relationship between the flange size in the image and $\mathbf{IoU}^{max}$

Given that the two aforementioned factors can be directly estimated through neural network inference (with the angle determined from the ratio of the bounding box sides and flange size inferred from the bounding box size), the proposed method can provide real-time feedback to the operator during the data collection process. This functionality enables the system to indicate whether the robot's flange is positioned correctly, allowing untrained operators to efficiently complete the calibration process.

D. Precision of Hand-eye Calibration

In comparison experiments, we assessed the performance using root mean square error of the reprojection (RRMSE) [29], expressed as follows:

$$e_{\text{rrmse}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\|p_i - \Pi\left(K, \mathbf{T}_b^c, \mathbf{T}_e^b, \mathbf{T}_w^e, \mathbf{P}_i^w\right)\right\|_2^2}, \quad (5)$$

where p_i denote the 2-D points regarded as ground truth, Π represents the operation that projects the 3-D points from camera coordinates to image space, $\mathbf{T}_b^c$ represents the camera-to-robot pose, $\mathbf{T}_e^b$ represents the robot pose, $\mathbf{T}_w^e$ represents the transformation between the calibration board and flange coordinate systems. Furthermore, $\mathbf{P}_i^w$ represent the 3-D points in the coordinate system of the calibration board. In this section, to calculate e_{rrmse}, we utilized the marker shown in Fig. 3 to serve as a calibration board. This marker can be seen as a simplified calibration board, with the center of the ArUco pattern serving as the sole reference point. By defining the origin of the calibration board's coordinate system at the center of the ArUco pattern, the coordinates of point $\mathbf{P}_i^w$ are consistently expressed as follows:

$$\mathbf{P}_i^w = \mathbf{0}_{3\times 1}. \quad (6)$$

When the marker was mounted on the flange, the alignment of the center of the flange with that of the ArUco pattern allowed for the definition of the calibration board's coordinate system to align with the flange coordinate system, resulting in the following equation

$$\mathbf{T}_w^e = \mathbf{I}_{4\times 4}, \quad (7)$$

in which $\mathbf{I}_{4\times4}$ denotes the identity transformation. Then (5) becomes

$$e_{\text{rrmse}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left\|p_i - \Pi\left(K, \mathbf{T}_b^c, \mathbf{T}_e^b, \mathbf{I}_{4\times4}, \mathbf{0}_{3\times1}\right)\right\|_2^2}. \quad (8)$$

In addition to the UR10e, the Panda robot shown in Fig. 4(a) was also employed, as it is commonly used as a benchmark platform for AI-based markerless calibration methods. To ensure a fair comparison, the image data of the Panda robot were captured using a RealSense camera, consistent with the setups in [11], [16]. During the experiment, the robot was first controlled to move around with the marker attached to collect test data for calculating e_{rrmse}. The movement method was consistent with the approach used in section IV-B. For each robot, 500 pairs of evaluation samples consists of a captured image and the corresponding robot pose were collected, then invalid data such as those where the ArUco marker was not detected were filtered out. On the UR10e, we compared our method with Tsai's [6] method, a widely used classic marker-based method. On the Panda robot, we additionally included comparisons with DREAM [11] and EasyHeC++ [16]. All methods, except EasyHeC++, were executed on an Intel i7-12700H CPU. Due to its high GPU memory requirements, EasyHeC++ was evaluated on an NVIDIA A100 GPU. The results were then applied into (8) for error comparison. For each method, 20 imagepose pairs were used in the calibration. For the markerless methods (ours, DREAM, and EasyHeC++), the robot followed the same predefined pose trajectory and the same set of imagepose pairs was used for calibration. For Tsais method, we used a separately designed trajectory to satisfy chessboard visibility and viewpoint requirements, and a high-precision chessboard calibration board was used during the experiment. Since the precision of EasyHeC++ depends heavily on the quality of the initial pose used for optimization, which is originally provided by DKM [23], we use the result from Tsais method alternatively as a theoretically more accurate initialization to reproduce the methods ideal performance. In our method, the **IoU** threshold ($thr_{\mathbf{IoU}}$) was set to 0.93. To evaluate the methods best performance on the Panda robot, we additionally trained a model on a combined dataset consisting of UR10e and 389 supplementary Panda images. The mean errors were calculated, with the results listed in Table I.

TABLE I
COMPARISON OF CALIBRATION ACCURACY BETWEEN THE PROPOSED AND BASELINE METHODS

Method	Train/Val Datasets	Test Robot	e_{rrmse}(px)
Tsai [6]	/	UR10e	4.93
Tsai	/	Panda	9.83
DREAM [11]	Panda	Panda	72.66
EasyHeC++ [16](*)	/	Panda	7.53
Ours	UR10e	UR10e	1.20
Ours	UR10e	Panda	3.23
Ours	UR10e+Panda	Panda	1.93

*Using Tsai method's result as initial pose.

The results demonstrate that the proposed method achieves outperformed calibration precision than all baseline methods. Compared with EasyHeC++, it not only achieves higher precision without requiring a robot CAD model as prior information, but also runs efficiently on significantly less powerful hardware. Notably, in experiments on the Panda robot, even the model trained solely on UR10e outperformed all baseline methods, despite being less accurate than the version trained with UR10e+Panda datasets. This highlights the strong generalization capability of our method across different robot platforms.

E. Generalization Capability Test

A key advantage of the proposed method is the good generalization capability of the trained model. As demonstrated in Section IV-D, the model trained on the UR10e robot achieves competitive accuracy when applied to the calibration of the Panda robot, despite differences in robot configuration. To further assess this generalization capability, we conducted the same experiments in Section IV-D on Fairino FR5, UR5e and UR3e robots shown in Fig. 4, and compared the results with those obtained from UR10e and Panda. Among them, UR5e and UR3e shares the same flange design as UR10e but has a different robot body, allowing us to evaluate performance under conditions where the flange remains consistent but the robot differs. In contrast, the Panda robot is equipped with a flange featuring additional groove structures, while the Fairino FR5 robot employs a matte-finished flange with low surface reflectivity. Therefore, these robots were used to evaluate the generalization performance in cases where both the flange and body structures differ from those seen during training. The results are as shown in Table II.

TABLE II
COMPARISON OF CALIBRATION ACCURACY ACROSS DIFFERENT ROBOTS

Test Robot	Train/Val Dataset	e_{rrmse} (px)
UR3e	UR10e	2.17
UR5e	UR10e	0.91
UR10e	UR10e	1.20
Panda	UR10e	5.12
Panda	UR10e+Panda	1.93
Fairino	UR10e	4.59
Fairino	UR10e+Panda	0.95

It can be seen that the result on both the UR3e and the UR5e robots shows similar precision with the result on the UR10e robot which used on training the model, and the result on the UR5e robot is even better. For Panda and Fairino, the UR10e-trained model could be directly applied with only a slight drop in precision. Furthermore, incorporating a small portion of Panda data into the training set effectively mitigated this precision loss. Notably, when the mixed UR10e+Panda model was applied to the Fairino robot, which was never shown in datasets before, the calibration exhibited excellent precision, demonstrating the zero-shot transferability of our approach. These findings suggest that the proposed method demonstrates strong generalization ca-

pability among robots with similar flange structures, thereby avoiding additional training.

V. Conclusion

This study introduced a simple yet effective markerless hand-eye calibration framework with high modualarity and generalization capability. Instead of relying on robot-body-keypoints detection, the proposed approach leveraged flange ellipse detection to provide input data for the PnP method. This is complemented by an IoU filter for discarding high-error data, ensuring the accuracy of both the input data and calibration results. Experimental results demonstrated that our method offered superior precision compared with the conventional Tsai's method and latest state-of-the-art Easy-HeC++ method. Additionally, our approach demonstrated strong generalization capability, showed strong performance even when applied to previously unseen robots. In the future, we aim to develop a universal model capable of recognizing a wider range of flanges. This will further enhance the method's generalization capability, allowing it to be directly applied to various robot types.

References

[1] A. Efendi, Y.-H. Shao, and C.-Y. Huang, "Technological development and optimization of pushing and grasping functions in robot arms: A review," *Measurement*, vol. 242, p. 115729, 2025.

[2] S. Zobov, F. Chervinskii, A. Rybnikov, D. Petrov, and K. Vendidandi, "Auto-assembly: a framework for automated robotic assembly directly from cad," *arXiv preprint arXiv:2301.02643*, 2023.

[3] N. Robinson, B. Tidd, D. Campbell, D. Kulić, and P. Corke, "Robotic vision for human-robot interaction and collaboration: A survey and systematic review," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 1, pp. 1–66, 2023.

[4] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 4, p. 71, 2021.

[5] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, "A comparative review of hand-eye calibration techniques for vision guided robots," *IEEE Access*, vol. 9, pp. 113 143–113 155, 2021.

[6] R. Y. Tsai, R. K. Lenz *et al.*, "A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.

[7] N. Andreff, R. Horaud, and B. Espiau, "Robot hand-eye calibration using structure-from-motion," *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 228–248, 2001.

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

[9] X. X. Lu, "A review of solutions for perspective-n-point problem in camera pose estimation," in *Journal of Physics: Conference Series*, vol. 1087, no. 5. IOP Publishing, 2018, p. 052009.

[10] J. Lambrecht and L. Kästner, "Towards the usage of synthetic data for marker-less pose estimation of articulated robots in rgb images," in *2019 19th International Conference on Advanced Robotics (ICAR)*, 2019, pp. 240–247.

[11] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9426–9432.

[12] A. Simoni, S. Pini, G. Borghi, and R. Vezzani, "Semi-perspective decoupled heatmaps for 3d robot pose estimation from depth maps," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 569–11 576, 2022.

[13] J. Lu, F. Richter, and M. C. Yip, "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4622–4629, 2022.

[14] J. Lu, F. Liu, C. Girerd, and M. C. Yip, "Image-based pose estimation and shape reconstruction for robot manipulators and soft, continuum robots via differentiable rendering," *arXiv preprint arXiv:2302.14039*, 2023.

[15] L. Chen, Y. Qin, X. Zhou, and H. Su, "Easyhec: Accurate and automatic hand-eye calibration via differentiable rendering and space exploration," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7234–7241, 2023.

[16] Z. Hong, K. Zheng, and L. Chen, "Easyhec++: Fully automatic hand-eye calibration with pretrained image models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 816–823.

[17] "Manipulating industrial robots Mechanical interfaces Part 1: Plates," International Organization for Standardization, London, UK, Standard, 2004, https://www.iso.org/standard/36578.html.

[18] B. Lu, B. Li, Q. Dou, and Y. Liu, "A unified monocular camera-based and pattern-free hand-to-eye calibration algorithm for surgical robots with rcm constraints," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5124–5135, 2022.

[19] F. Zhong, B. Li, W. Chen, and Y.-H. Liu, "Robotcamera calibration in tightly constrained environment using interactive perception," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4952–4970, 2023.

[20] F. Wan and C. Song, "Flange-based hand-eye calibration using a 3d camera with high resolution, accuracy, and frame rate," *Frontiers in Robotics and AI*, vol. 7, p. 65, 2020.

[21] V. Đalić, V. Jovanović, and P. Marić, "Submillimeter-accurate markerless hand–eye calibration based on a robots flange features," *Sensors*, vol. 24, no. 4, p. 1071, 2024.

[22] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9799–9808.

[23] J. Edstedt, I. Athanasiadis, M. Wadenbäck, and M. Felsberg, "Dkm: Dense kernelized feature matching for geometry estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 765–17 775.

[24] T. Shaharabany, A. Dahan, R. Giryes, and L. Wolf, "Autosam: Adapting sam to medical images by overloading the prompt encoder," *arXiv preprint arXiv:2306.06370*, 2023.

[25] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[27] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[28] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[29] I. Ali, O. Suominen, A. Gotchev, and E. R. Morales, "Methods for simultaneous robot-world-hand–eye calibration: A comparative study," *Sensors*, vol. 19, no. 12, p. 2837, 2019.