Stable Tracking of Eye Gaze Direction During Ophthalmic Surgery

Tinghe Hong, Shenlin Cai, Boyang Li, Kai Huang*

Abstract— **Ophthalmic surgical robots offer superior stability and precision by reducing the natural hand tremors of human surgeons, enabling delicate operations in confined surgical spaces. Despite the advancements in developing vision- and force-based control methods for surgical robots, preoperative navigation remains heavily reliant on manual operation, limiting the consistency and increasing the uncertainty. Existing eye gaze estimation techniques in the surgery, whether traditional or deep learning-based, face challenges including dependence on additional sensors, occlusion issues in surgical environments, and the requirement for facial detection. To address these limitations, this study proposes an innovative eye localization and tracking method that combines machine learning with traditional algorithms, eliminating the requirements of landmarks and maintaining stable iris detection and gaze estimation under varying lighting and shadow conditions. Extensive real-world experiment results show that our proposed method has an average estimation error of 0.58 degrees for eye orientation estimation and 2.08-degree average control error for the robotic arm's movement based on the calculated orientation.**

I. Introduction

Ophthalmic surgical robots demonstrate significant stability and precision compared to human surgeons when performing surgical tasks. The design of robotic arms effectively eliminates or substantially reduces natural hand tremors during surgery, enabling more refined and controlled operations in extremely confined surgical spaces, such as those encountered in retinal surgery. This stability advantage not only increases the success rate of surgeries but also minimizes the risk of damage to surrounding healthy tissues.

Significant progress has been made in the development of ophthalmic surgical robots. Various studies have proposed vision- and force-based compliant control methods for ophthalmic surgical robots to achieve precise movement control of surgical instruments during contact with ocular tissues [1], [2]. However, despite the advancements in surgical robotics, the preoperative navigation in ophthalmic surgery largely remains dependent on manual operation. This reliance not only limits the consistency and repeatability of surgeries but also increases the uncertainty during the surgical process.

In the field of ophthalmic surgery, preoperative and intraoperative navigation serve distinct roles. Particularly before surgical operations commence, the lack of physical contact between the instruments and ocular tissues prevents surgical robots from utilizing force feedback mechanisms for accurate position determination [3]. Consequently, preoperative navigation primarily relies on eye-gaze estimation techniques guided by visual cues to achieve precise positioning of the robotic arm. Currently, eye-gaze estimation techniques are categorized into two main types: traditional edge detection-based algorithms [4] and the more recent deep learning-based approaches [5], [6].

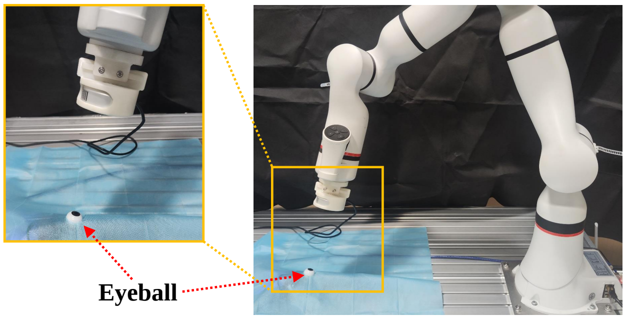


Fig. 1: A mock ophthalmic surgical environment with a robotic arm.

There are several key challenges in recognizing eye gaze direction in surgical environments (see Fig. 1). Firstly, the primary perception equipment used in the surgery is a variable focal length endoscope. [7] and [8] utilized Kinect depth sensors for real-time eye tracking. Although the methods are simple and low-cost [9], they requires the introduction of depth-sensing devices, thereby increasing the uncertainty in the surgical setting. Secondly, some methods choose to identify the eyes using fixed-point light reflections [10], [11]. Unfortunately, maintaining stable light reflection positions in the operating room is challenging, as they are often obstructed during surgery. Lastly, the patient's face are typically covered by a surgical pad sheet during the surgery, making the face not visible. Some learning-based methods [12], [13] are less sensitive to environmental interference. For instance, the authors in [14] employed an unsupervised approach to improve algorithm performance across different domains, while [15] used a weakly supervised method for unconstrained gaze estimation. These methods generally require initial face detection before identifying the eye's position and orientation, lacking stability during the tracking process.

To address the aforementioned challenges, this study proposes an eye localization and tracking method that integrates machine learning techniques with traditional algorithms. Firstly, a fine-tuned YOLO [16] model is employed to accurately identify the sclera and iris regions, effectively overcoming the limitations associated with face detection. Then, the RANSAC method is used to stably determine

The authors are with the school of computer science and engineering, Sun Yat-sen University,Guangzhou, China.

This research is supported by the Shenzhen Medical Research Foundation (Grant No. SZMRF-D2404009) and the Science and Technology Development Fund of Macau Special Administrative Region (Grant No. 0010/2023/AKP).

the eye position within the identified region, enhancing the robustness of the tracking process. Finally, the positional relationship between the eye and pupil is used to resolve ambiguities in gaze estimation. Our approach provides a stable solution for eye localization and tracking suitable for use with robotic arms. Extensive real-world experiments demonstrate that the proposed method achieves an average orientation estimation error of 0.58°, an average robotic arm control error of 2.08°, and a relative distance error of 6.4mm between the eye and camera.

II. Related Works

In research on eye gaze tracking, accurate eye localization is a fundamental and critical step, involving the recognition of the iris or sclera. In [17], researchers estimate the center position of the eye by calculating the intersections of multiple lines passing through the edges of the iris. However, this method does not address the problem of precise localization of the iris center in the visual frame. Another study [18] proposed an approach based on facial feature point detection, where key facial landmarks are firstly identified. The eye-related features are further recognized. While this method can assist in eye localization to some extent, it remains limitations by the recognizability of facial features.

In estimating eye gaze direction, the method proposed in [19] requires obtaining facial orientation information before accurately estimating the eye's gaze, which is challenging in cases of facial occlusion or unclear vision. Similarly, the approaches in [20] and [21] rely on accurate recognition of facial features for eye gaze estimation, leading to the reduced accuracy or failure to estimate when facial features are indistinct or obscured. Additionally, the method in [22] requires facial features such as eyelids to estimate eye gaze direction, thus also being limited by the detectability of the features. Although the study in [17] provides gaze information under certain conditions, its effectiveness depends on the relative stability of the camera and eye positions, which is difficult to maintain in a dynamically changing surgical environment. In [23], the authors achieve surgical robot navigation by identifying trocars, providing an innovative solution for eye gaze estimation. However, this method requires pre-placement of trocars on the sclera.

III. Method

The proposed method in this study consists of several steps: (1) accurately and stably identifying the position of the iris in the camera frame under varying lighting conditions; (2) calculating the eye gaze direction using the camera's intrinsic parameters and the position of the iris; (3) resolving the ambiguity in gaze estimation by analyzing the relative positions of the iris and sclera centers.

A. Iris Position Acquisition

To determine the position of the iris in an image, a fine-tuned YOLO model is first employed to identify the region containing the iris, avoiding the target loss issues caused by interference from other similar objects in the camera's field of view, which can occur when directly using the pupiltrack method mentioned in [24]. The YOLO model is trained on a dataset of approximately one hundred annotated images of eyes. The fine-tuned model demonstrates robustness against interference and can accurately detect the general region of the iris, although its determination of the precise iris position remains somewhat unstable.

To achieve stable localization of the iris in the image, the region identified is firstly subjected to a binarization process to distinguish the darker and the lighter area of the sclera.

To enhance the stability of detection and ensure consistent differentiation of the iris's dark region under varying shadow or lighting conditions, an adaptive thresholding method for binarization is employed.

$$thd = avg * k. \tag{1}$$

Here, thd represents the threshold, avg is the mean color value within the region, k is a scaling factor. When the overall brightness within the region decreases, the threshold will also decrease accordingly to ensure that the iris region can still be distinguished after binarization (see Fig. 2b and Fig. 2c). Additionally, performing the assessment within the defined region helps to avoid interference from variations in ambient lighting and shadows.

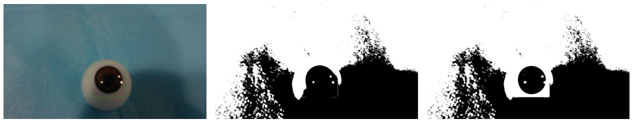

(a) Original. (b) Fixed. (c) Adaptive.

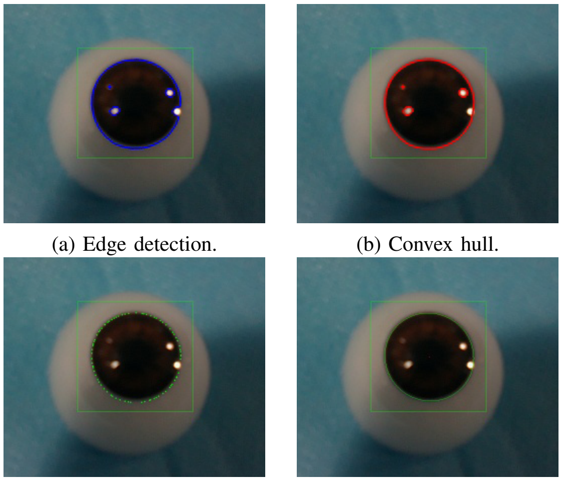
Fig. 2: Binarization results with fixed and adaptive threshold under shadow conditions.

Within the defined region, edge detection is firstly performed to extract points located at the boundaries (Fig. 3a). It can be observed that some reflective spots on the iris are also identified as edge points, which introduces a certain degree of interference in the recognition process. Subsequently, a convex hull is constructed based on the detected edge points (Fig. 3b). This step helps to filter out the interfering points located at the edges. However, some noise points within the iris region are also detected as a separate convex hull. To accurately identify the true convex hull corresponding to the iris, we utilize the region defined by YOLO and select the largest convex hull by area, which represents the iris, to effectively filter out smaller convex hulls formed by noise points. The vertices of this convex hull are then extracted (Fig. 3c). Finally, an ellipse is fitted to the selected vertices, providing the precise position of the iris (Fig. 3d).

B. Calculating Eye Position and Gaze Direction

The ellipse fitted in the previous section represents the projection of the iris in 3D space onto the camera frame (see Fig. 4), The iris in 3D space can be considered as a circle with a radius of Ir_R, measured in millimeters.

Let the major and minor axes of the ellipse defined as ax_{maj} and ax_{min} (in pixel units), respectively. The center of



(a) Edge detection.

(b) Convex hull.

(c) RANSAC iteration.

(d) Fit ellipse.

Fig. 3: Binarization results with different thresholds under shadow conditions.

the ellipse in the image is located at p_x and p_y, with the ellipse's rotation angle relative to the image's x-axis being ψ. f_x and f_y represent the x-axis and y-axis pixel focal lengths of the camera, while pr_x and pr_y denote the pixel coordinates of the screen's center. To compute the coordinates of the eyeball in three-dimensional space, let $f_z = (f_x + f_y)/2$. Based on the similar triangles, the relative coordinates $[Ir_x, Ir_y, Ir_z]$ of the iris to the camera frame center are given by:

$$\begin{cases} Ir_z = f_z * Ir_R / ax_{maj} \\ Ir_x = -Ir_z * (p_x - pr_x)/f_x \\ Ir_y = Ir_z * (p_y - pr_y)/f_y. \end{cases} \tag{2}$$

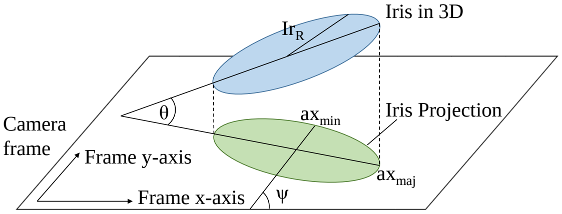


Fig. 4: The projection of the iris from 3D space onto the 2D camera frame.

The rotation angle θ of the eye relative to the camera plane is defined as the follows:

$$\theta = \arccos(ax_{min}/ax_{maj}). \tag{3}$$

Thus, the normal vector $\vec{n}$ of the eye gaze direction is defined as follows:

$$\vec{n} = [-\sin(\theta) * \cos(\psi), -\sin(\theta) * \sin(\psi), \cos(\theta)]^\top. \tag{4}$$

When the eye is not centered in the camera's field of view, the obtained eye gaze direction needs to be adjusted. Firstly, the distance d between the center of the ellipse and the center of the field of view is calculated by:

$$d = (Ir_x^2 + Ir_y^2)^{1/2}. \tag{5}$$

The angle of rotation γ required is given by,

$$\gamma = \arctan(d/Ir_z). \tag{6}$$

The rotation axis $\vec{l}$ is given by,

$$\vec{l} = [Ir_y/d, Ir_x/d, 0]^\top. \tag{7}$$

The normal vector $\vec{n}$ of the eye gaze direction is corrected using Rodrigues' rotation formula, resulting in the corrected normal vector $\vec{n'}$,

$$\vec{n'} = \vec{n} \cdot \cos(\gamma) + (\vec{l} \times \vec{n}) \cdot \sin(\gamma) + \vec{l}(\vec{l} \cdot \vec{n})(1 - \cos(\gamma)). \tag{8}$$

C. Ambiguities in Eye Gaze Direction

In the process of recovering the 3D eye orientation from 2D images, ambiguity arises because a single projection may correspond to two different eye orientations. Therefore, resolving this ambiguity is essential to accurately determine the eye's orientation.

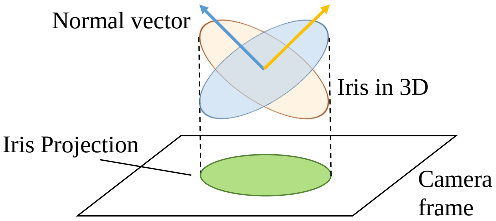


Fig. 5: For an elliptical projection, two possible circular orientations can be reconstructed.

As shown in Fig. 5, when the 2D projection in the camera is reconstructed into the 3D space of the iris, two distinct solutions can arise, corresponding to the two different normal vectors. To resolve this, a method similar to that described in Section 3.A is used to obtain the sclera's pixel coordinates, which are then compared with the iris's pixel coordinates to determine the true orientation of the iris (see Fig. 6).

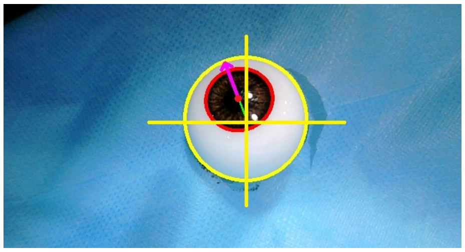


Fig. 6: The true orientation of the eye is determined by comparing the relative positions of the iris and sclera centers.

IV. Experiments

To validate the stability and accuracy of the proposed method, three sets of experiments are designed. The first experiment compares the iris recognition performance. The second experiment evaluates the accuracy of eye-gaze direction estimation and the ability to resolve ambiguities. Finally, in the third experiment, a camera is mounted on a robotic arm. The proposed method is used to compute the eye's position and orientation, enabling the end-effector of the robotic arm to follow the eye's movements in real-time. The proposed approach runs on a computer equipped with an Intel i5-10400 processor and an NVIDIA Titan V GPU for YOLO inference.

A. 2D Iris Recognition

In this experiment, the camera remains stationary while the eye model rotates around a fixed axis. The goal of the experiment is to compare the different methods for iris recognition and tracking in the camera's field of view. Five different methods are evaluated: our proposed method, the RANSAC-based Pupiltrack [24], the learning-based GazeML method [25], the CNN-based Pupil-Locator [26](Abbreviated as Locator), and the 3D Eye Tracker [27].

The experiments are conducted under two different resolution settings. The first setting uses a resolution of 1920 × 1080 at 30Hz, with the camera positioned approximately 100mm from the eye. However, under these conditions, 3D Eye Tracker failed to detect the iris. To address this, the camera is adjusted to a position approximately 50mm from the eye, with the resolution set to 640 × 360 at 30Hz. At this closer distance, Locator struggled to detect the iris. Additionally, GazeML required landmarks to identify the iris, YOLO is employed to predefine the region containing the eye for more effective detection.

We set the eyeball rotation duration to 10 seconds, rotating around an axis parallel to the y-axis of the camera frame, from -30° to $+30^\circ$, resulting in a total of 300 frames for analysis. A tracking failure is defined as a deviation exceeding 30 pixels. Tab. I presents the number of frames in which each recognition method lost tracking out of the 300 frames. Cases where recognition was not possible are denoted by a "-" symbol. The frame loss statistics for each method are displayed in Tab. I.

TABLE I: The number of lost frames in iris detection tasks in 300-frame videos at two different resolutions.

	Resolution 1920×1080	Resolution 640×360
Ours	0	0
GazeML[25]	3	16
Pupiltrack[24]	5	10
Locator[26]	-	0
3D Eye Tracker[27]	7	-

As shown in Tab. I, all methods except ours experienced varying degrees of tracking loss. By using YOLO to define the region of the eye and employing an adaptive thresholding approach, our method is able to consistently detect the iris's position in the image under various lighting conditions.

Fig. 7 illustrates the tracking failures of other methods under conditions with lighting variations and shadows. The Locator (Fig. 7c) performs well when the camera is positioned at a greater distance from the eye (Resolution 1920×1080), but exhibits significant tracking loss at closer distances (Resolution 640×360). Both Pupiltrack (Fig. 7b) and 3D Eye Tracker(Fig. 7d) are highly susceptible to interference from shadows. The GazeML (Fig. 7a) tends to deviate in recognition process, leading to unstable positioning.

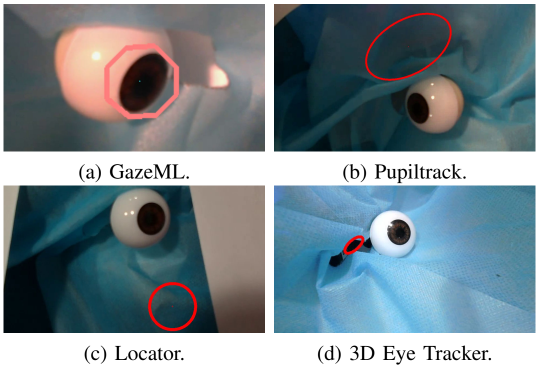

(a) GazeML.
(b) Pupiltrack.
(c) Locator.
(d) 3D Eye Tracker.

Fig. 7: The instances of frame loss for each method.

Fig. 8 depicts the positional shifts of various recognition techniques within the camera's coordinate system throughout the eye movement. The horizontal and vertical axes represent pixel measurements. For clarity, only the coordinates of points where the tracking remains uninterrupted are shown. It should be noted that Pupiltrack is omitted from the figure due to its coordinates closely aligning with those of our proposed method when tracking is uninterrupted.

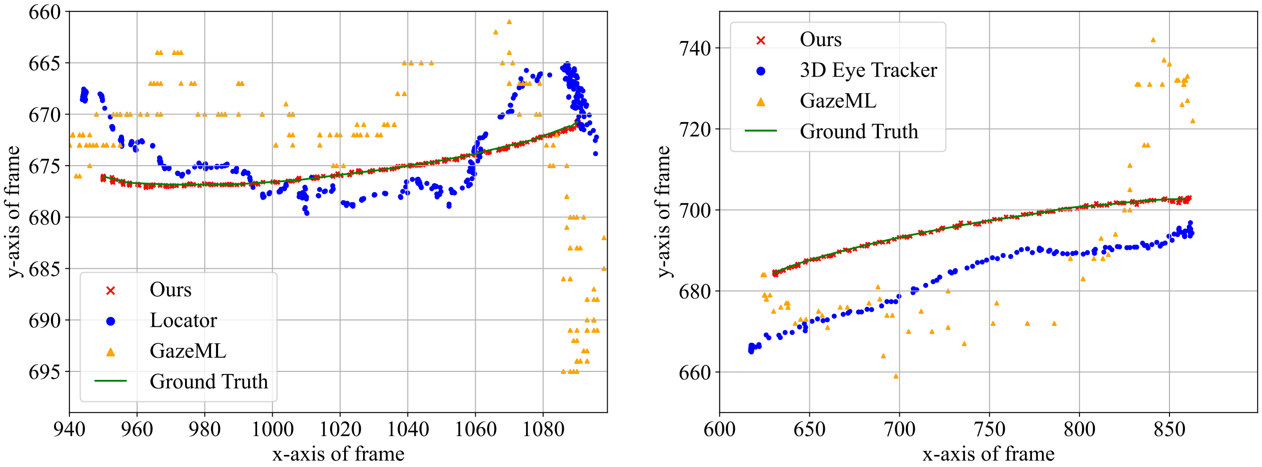
As shown in Fig. 8, our method closely matches the Ground Truth during the tracking process. In contrast, the two learning-based methods (GazeML and Locator) demonstrate less stability. The detailed tracking results are presented in Tab. II.

Tab. II presents the mean and standard deviation of pixel deviation from the ground truth for different methods at two resolutions. The mean metric represents the accuracy of each tracking method, while the standard deviation metric indicates the stability of the tracking. Since points with significant deviation (considered as tracking loss) are excluded from the statistics, our method and the Pupiltrack method show similar performance. The GazeML method exhibits a higher level of jitter, while the Locator and 3D Eye Tracker methods are relatively more stable but exhibit lower adaptability.

B. 3D Eyeball Orientation Estimation

In this experiment, we firstly set the eyeball to rotate 30 degrees around an axis parallel to the x-axis of the camera frame. Then, we control the eyeball to rotate from -30° to $+30^\circ$ around an axis parallel to the y-axis of the camera frame while calculating the normal vector of the eyeball relative to the camera.

In this experiment, we do not include a comparison with Locator, as it cannot provide a 3D eyeball orientation. Additionally, the 3D Eye Tracker requires a pre-established



(a) Iris center at resolution 1920×1080.

(b) Iris center at resolution 640×360.

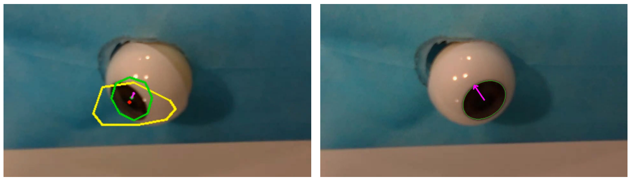
Fig. 8: The instances of frame loss for each recognition method.

TABLE II: The mean and standard deviation of the error from the ground truth during tracking, measured in pixels.

	Resolution 1920×1080		Resolution 640×360	
	Mean	Standard	Mean	Standard
Ours	0.119	0.103	0.749	0.278
GazeML	8.631	6.591	17.019	10.439
Pupiltrack	0.095	0.087	0.814	0.258
Locator	3.423	2.421	-	-
3d Eye Tracker	-	-	11.704	3.835

model to compute the orientation, which is not suitable for our current experimental setup.

The purple arrows in Fig. 9 represent the eye orientations estimated by different algorithms. The GazeML method suffers from instability when detecting the iris position, which adversely affects the accuracy of its eye orientation estimation (see Fig. 9a). As a result, the estimated normal vector significantly deviates when the eye deflection angle is large. On the other hand, Pupiltrack lacks a mechanism to resolve ambiguity, leading to instances where the direction is incorrectly identified as the opposite (see Fig. 9b).

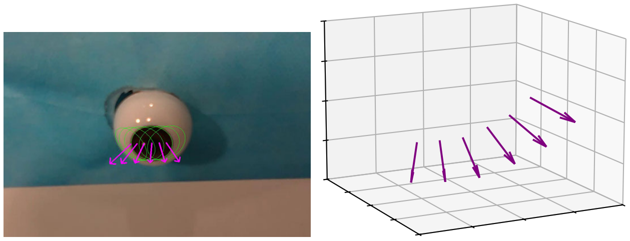


(a) GazeML.

(b) Pupiltrack.

Fig. 9: Errors in eye orientation estimation.

Profit from stable tracking, our method also demonstrates consistent performance in estimating eye orientation. Furthermore, by resolving the ambiguity issue, our approach avoids misjudgment of the eye's direction. Even when the significant changes occur in the projection's rotation angle (πs), the method accurately identifies the eye's orientation throughout the rotation process. The experimental results are illustrated in Fig. 10.



(a) Orientation in 2D.

(b) Orientation in 3D.

Fig. 10: Eyeball position and orientation in 2D image and 3D coordinate system.

Fig. 10 illustrates the estimated eye orientation and position using our proposed method. The purple arrows indicate the estimated eye orientation. Fig. 10a shows the estimated orientation and position of the eye plotted in the camera frame while Fig. 10b displays the estimated eye position and orientation in a 3D coordinate system. From the results, it can be concluded that our method exhibits both stability and accuracy in estimating the eye's position and orientation.

In Tab. III, we present the estimated normal vectors and their corresponding ground truth values at 5-degree intervals. Angle metric denotes the rotation angle of the eye, Estimation metric represents the estimated normal vector. Ground Truth metric indicates the actual normal vector of the eye. Error metric shows the angular deviation.

As shown in Tab. III, our method not only resolves the ambiguity issue but also achieves a small deviation between the estimated and true angles, with an average error of 0.584°. Generally, larger errors occur when the deflection angle approaches to ±30°. This can be attributed to the fact that, as the deflection angle increases, the ratio between the major and minor axes changes more drastically, thereby amplifying the error.

TABLE III: Comparison of estimated normal vectors and ground truth normal vectors at various eye rotation angles, along with the corresponding angular errors.

Angle	Estimation(x,y,z)			Ground truth(x,y,z)			Error
$-30°$	0.428	0.514	-0.742	0.433	0.5	-0.750	0.968°
$-25°$	0.364	0.507	-0.780	0.365	0.5	-0.784	0.514°
$-20°$	0.299	0.501	-0.811	0.296	0.5	-0.813	0.303°
$-15°$	0.227	0.494	-0.838	0.224	0.5	-0.836	0.418°
$-10°$	0.150	0.491	-0.857	0.150	0.5	-0.852	0.572°
$-5°$	0.073	0.490	-0.868	0.075	0.5	-0.862	0.659°
0°	0.006	0.494	-0.869	0.0	0.5	-0.866	0.517°
5°	-0.076	0.493	-0.866	-0.075	0.5	-0.862	0.406°
10°	-0.150	0.493	-0.856	-0.150	0.5	-0.852	0.451°
15°	-0.224	0.497	-0.838	-0.224	0.5	-0.836	0.188°
20°	-0.292	0.505	-0.811	-0.296	0.5	-0.813	0.351°
25°	-0.365	0.515	-0.775	-0.365	0.5	-0.784	1.010°
30°	-0.429	0.518	-0.739	-0.433	0.5	-0.750	1.241°

C. Eye Tracking with a Surgical Robotic Arm

In this experiment, we mounted a camera on a 7-degrees robotic arm and rotate the eye model to allow the robotic arm to track the eye's movement. This setup is designed to test the stability and accuracy of our proposed method for eye tracking.

The experimental setup is shown in Fig. 1, where a RealSense camera is mounted at the end of a robotic arm with an eye model rotated by a motor. In the world coordinate system, the eye rotates around the x-axis from -30 degrees to +30 degrees. In this experiment, the end-effector of the robotic arm is controlled to perform rotational or translational movements, ensuring that the camera remains oriented directly towards the iris while maintaining a fixed distance from it. We record the position and orientation of the robotic arm's end effector at 10-degree intervals, compared them with the computed position and orientation of the eye, and plotted the results in Fig. 11, coordinates are in millimeters.

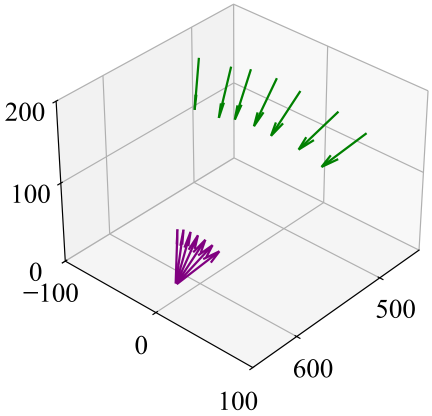


Fig. 11: The position and orientation of the eye model and the camera.

In Fig. 11, the coordinate system is based on the world coordinate frame, with units in millimeters. The purple arrows indicate the orientation of the eyeball and the position of the iris center, which are computed based on the motor rotation values along with the dimensions of the eyeball and iris. The green arrows represent the position and orientation of the camera controlled by the robotic arm. As the eyeball rotates, the end effector of the robotic arm, guided by our method, also rotates around the eyeball. Throughout the entire process, the camera maintains a consistent orientation facing the eyeball and keeps a relatively fixed distance from it. The specific data is presented in Tab. IV.

TABLE IV: Comparison of estimated normal vectors and ground truth normal vectors at various eye rotation angles, along with the corresponding angular errors.

Eyeball angles	Distance(mm)	Error angles
$-30°$	212.189	3.750°
$-20°$	206.088	2.918°
$-10°$	208.020	1.792°
0°	208.424	1.187°
10°	207.224	0.681°
20°	211.029	3.255°
30°	212.557	0.967°

Tab. IV presents the distance and orientation angle errors between the camera and the eye when the eye is rotated to seven different angles. In the table, ”eyeball angles” represents the angles of rotation of the eye around the x-axis in the world coordinate system, ”distance” indicates the distance between the camera and the eye, and ”error angles” refer to the angle between the eye's orientation and the camera's orientation (with the camera orientation inverted to calculate the angle). Throughout the entire process, the average distance between the camera and the eye was 209.362 mm, and the average error angle was 2.078°. According to Tab. III, approximately 1.494° of error can be attributed to the position and orientation control of the robotic arm's end-effector. The table shows that under our method's control, the surgical robotic arm can maintain a consistent distance from the eye and continuously keep the camera aligned with the eye.

V. Conclusion

In this study, we propose a robust method for estimating the 3D orientation of the eyeball from 2D images. Our approach effectively addresses the ambiguity in determining the eyeball's orientation by leveraging adaptive thresholding and the accurately detect the iris region under varying lighting conditions. Through extensive experiments, we demonstrate that our method outperforms other existing approaches in both stability and accuracy. Our method shows consistent performance across different resolutions and lighting conditions. Additionally, our approach accurately estimates the eyeball's orientation with minimal errors, particularly when the rotation angles are small, and resolves ambiguity by comparing the relative positions of the iris and sclera centers. Overall, our method offers a significant improvement in tracking and orientation estimation of the eyeball, which could be beneficial for various applications in computer vision and human-computer interaction.

References

[1] Ning Wang, Xiaodong Zhang, Danail Stoyanov, Hongbing Zhang, and Agostino Stilli. Vision-and-force-based compliance control for a posterior segment ophthalmic surgical robot. *IEEE Robotics Autom. Lett., RAL*, 8(11):6875–6882, 2023.

[2] Zhibin Fan, Jiayuan Wang, Mengyao Liu, Jiahui Yang, Jie Zhao, and He Zhang. Ophthalmic surgical robot design and calibration methods. In *IEEE International Conference on Robotics and Biomimetics, ROBIO*, pages 1–6, 2023.

[3] Xingchi He, Marcin Balicki, Peter Gehlbach, James Handa, Russell Taylor, and Iulian Iordachita. A multi-function force sensing instrument for variable admittance robot control in retinal microsurgery. In *IEEE International Conference on Robotics and Automation, ICRA*, pages 1411–1418, 2014.

[4] Anuradha Kar and Peter Corcoran. A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. *IEEE Access*, 5:16495–16519, 2017.

[5] Shreya Ghosh, Abhinav Dhall, Munawar Hayat, Jarrod Knibbe, and Qiang Ji. Automatic gaze analysis: A survey of deep learning based approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence, TPAMI*, 46(1):61–84, 2023.

[6] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence, TPAMI*, 46(12):7509–7528, 2024.

[7] Dmitri Model and Moshe Eizenman. User-calibration-free remote eye-gaze tracking system with extended tracking range. In *Proceedings of the 24th Canadian Conference on Electrical and Computer Engineering, CCECE*, pages 1268–1271, 2011.

[8] Jianfeng Li and Shigang Li. Eye-model-based gaze estimation by RGB-D camera. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 606–610, 2014.

[9] Xiaolong Zhou, Haibin Cai, Zhanpeng Shao, Hui Yu, and Honghai Liu. 3d eye model-based gaze estimation from a depth sensor. In *IEEE International Conference on Robotics and Biomimetics, ROBIO*, pages 369–374, 2016.

[10] André Meyer, Martin Böhme, Thomas Martinetz, and Erhardt Barth. A single-camera remote eye tracker. 4021:208–211, 2006.

[11] Carlos Hitoshi Morimoto, Arnon Amir, and Myron Flickner. Detecting eye position and gaze from a single camera and 2 light sources. In *16th International Conference on Pattern Recognition, ICPR*, pages 314–317, 2002.

[12] Shreya Ghosh, Munawar Hayat, Abhinav Dhall, and Jarrod Knibbe. MTGLS: multi-task gaze estimation with limited supervision. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV*, pages 1161–1172, 2022.

[13] Manuel J. Marín-Jiménez, Vicky Kalogeiton, Pablo Medina-Suarez, and Andrew Zisserman. Laeo-net++: Revisiting people looking at each other in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence, TPAMI*, 44(6):3069–3081, 2022.

[14] Yihua Cheng, Yiwei Bao, and Feng Lu. Puregaze: Purifying gaze feature for generalizable gaze estimation. In *Conference on Artificial Intelligence, AAAI*, pages 436–443, 2022.

[15] Rakshit Kothari, Shalini De Mello, Umar Iqbal, Wonmin Byeon, Seonwook Park, and Jan Kautz. Weakly-supervised physically unconstrained gaze estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 9980–9989, 2021.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 779–788, 2016.

[17] Yuk-Hoi Yiu, Moustafa Aboulatta, Theresa Raiser, Leoni Ophey, Virginia L Flanagin, Peter Zu Eulenburg, and Seyed-Ahmad Ahmadi. Deepvog: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of neuroscience methods*, 324:108307, 2019.

[18] Sergio Canu. Eye detection-gaze controlled keyboard with python and opencv p. 1. *Pysource, Jan*, 7, 2019.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 815–823, 2015.

[20] Andy Catruna, Adrian Cosma, and Emilian Radoi. Crossgaze: A strong method for 3d gaze estimation in the wild. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2024.

[21] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *International Conference on Computer Vision. ICCV*, 2019.

[22] Aayush K Chaudhary, Rakshit Kothari, Manoj Acharya, Shusil Dangi, Nitinraj Nair, Reynold Bailey, Christopher Kanan, Gabriel Diaz, and Jeff B Pelz. Ritnet: Real-time semantic segmentation of the eye for gaze tracking. In *IEEE/CVF International Conference on Computer Vision Workshop, ICCVW*, pages 3698–3702, 2019.

[23] Shervin Dehghani, Michael Sommersperger, Junjie Yang, Mehrdad Salehi, Benjamin Busam, Kai Huang, Peter Gehlbach, Iulian Iordachita, Nassir Navab, and M Ali Nasseri. Colibridoc: An eye-in-hand autonomous trocar docking system. In *International Conference on Robotics and Automation, ICRA*, pages 7717–7723, 2022.

[24] Nithin Philip. pupiltrack-nystagmus. [Online]. Avaliable: https://github.com/nphilip1098/pupiltrack-nystagmus, 2021.

[25] Seonwook Park, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In *ACM symposium on eye tracking research & applications*, pages 1–10, 2018.

[26] Shaharam Eivazi, Thiago Santini, Alireza Keshavarzi, Thomas Kübler, and Andrea Mazzei. Improving real-time cnn-based pupil detection through domain-specific data augmentation. In *ACM symposium on eye tracking research & applications*, pages 1–6, 2019.

[27] Alexander Plopski, Jason Orlosky, Yuta Itoh, Christian Nitschke, Kiyoshi Kiyokawa, and Gudrun Klinker. Automated spatial calibration of hmd systems with unconstrained eye-cameras. In *IEEE international symposium on mixed and augmented reality, ISMAR*, pages 94–99, 2016.