Eye Pose Estimation and Tracking Using Iris as a Base Feature

Dmitry Shmunk[1], **Member, IEEE**
[1]Almalence MCHJ, Tashkent, 100015, Uzbekistan

Corresponding author: Dmitry Shmunk (e-mail: dmitry.shmunk@ almalence.com).

ABSTRACT A novel, fast, and robust method for 3D eye pose tracking that leverages the anatomical constancy of the human iris to improve accuracy and computational efficiency is proposed. Traditional pupil-based methods suffer from limitations due to pupil size variability, decentering, and the need for complex corrections for refraction through the corneal bulge. In contrast, the iris, due to its fixed size and direct visibility, serves as a more reliable feature for precise eye pose estimation. Our method combines key advantages of both model-based and regression-based approaches without requiring external glint-producing light sources or high computational overheads associated with neural-network-based solutions. The iris is used as the primary tracking feature, enabling robust detection even under partial occlusion and in users wearing prescription eyewear. Exploiting the consistent geometry of the iris, we estimate gaze direction and 3D eye position with high precision. Unlike existing methods, the proposed approach minimizes reliance on pupil measurements, employing the pupil's high contrast only to augment iris detection. This strategy ensures robustness in real-world scenarios, including varying illumination and stray light/glints/distortions introduced by corrective eyewear. Experimental results show that the method achieves low computational cost while maintaining state-of-the-art performance.

INDEX TERMS Eye tracking, pupil detection, iris detection, gaze tracking, image processing.

I. INTRODUCTION

Eye tracking technology has emerged as a necessary component in applications spanning virtual reality (VR), augmented reality (AR) and mixed reality (MR) – collectively XR. These technologies rely on the precise determination of eye pose and gaze direction to create immersive experiences and optimally render the scene. Despite its importance, robust and accurate eye tracking remains a challenge, primarily because of physiological variability.

A good overview of existing methods can be found in [1], section ‘Introduction’ and [2], section ‘State of The Art’.

Current state-of-the-art eye tracking methods can be broadly categorized into three approaches: pupil tracking, glint-based methods, and neural network-based approaches. Each has its strengths and weaknesses:

• Methods based on pupil tracking leverage the high contrast between the pupil and surrounding structures, making it easy to locate. However, variability in pupil size due to illumination changes, variations in decentering relative to the visual axis which is not just an individual characteristic but also depends on pupil radius, and its positioning behind the corneal bulge which refracts the light rays complicate accurate 3D localization and gaze estimation.

• Glint-based techniques that utilize reflections from controlled light sources, can yield reliable 3D eye position estimates. One of the well-known examples of such an approach is PCCR (Pupil Center Cornea Reflections) method. However, these methods are often disrupted by prescription eyewear glints and optical distortions or environmental stray lighting, significantly limiting their robustness.

• Neural network-based approaches excel in analyzing complex eye imagery; however, their computational costs are prohibitive for real-time applications, and they typically provide limited information about eye position relative to the visual axis.

Emerging from these limitations is the concept of iris-based eye tracking. Unlike pupil, the iris maintains a constant shape and size across different lighting conditions and individuals, making it an ideal candidate for accurate gaze estimation. Additionally, the direct visibility of the iris eliminates the need for corrective procedures to account for corneal refraction. However, the challenges of low contrast and partial occlusion due to eyelids necessitate additional image processing and

modeling techniques to realize the full potential of iris-based tracking.

The proposed method aims to integrate the strengths of iris-based tracking with computationally efficient algorithms to achieve a precise and robust eye pose estimation. Pseudo-polar rasterization of the iris results in simple procedure of iris pose and dimensions extraction, which is highly efficient computationally. This approach avoids reliance on external glint-producing light sources and high-computational-cost neural networks. Exploiting the constant geometry of the iris, the method demonstrates significant advancements in eye tracking accuracy and reliability, even in real-world conditions such as users wearing prescription glasses or under variable illumination.

This paper contributions lie in providing accurate 3D eye and gaze estimation with low computational cost utilizing iris-centric eye tracking, enabling new possibilities for eye tracking in consumer devices and professional applications.

II. PROPOSED METHOD

A brief outline of a tracking method using the iris as a base feature is as follows:

- Find pupil location.
- Estimate the pupil ellipse to roughly locate the iris.
- Refine the location and shape of the iris and find the iris edge (limbus) via pseudo-polar iris rasterization.
- The refined location and shape of the iris edge are then utilized to fit the 3D eye model and extract the pose of an eye (x/y/z position of eyeball center and gaze vector).

A. OVERVIEW

The proposed iris-based eye tracking method is designed to estimate the 3D eye pose and gaze direction with high precision and computational efficiency. Unlike traditional pupil- or glint-based approaches, this method primarily leverages the geometric constancy of the human iris, offering advantages in accuracy and independence from lighting conditions.

The method begins with pupil detection, utilizing its high contrast to establish a preliminary reference point for further steps. Once the pupil's approximate position was determined, the iris parameters are identified and processed as the primary feature for pose estimation. By using the iris, which maintains a fixed size and is directly visible without optical distortion, the method avoids challenges such as pupil dilation, decentering, and refraction effects caused by the corneal bulge.

Processing steps involve pupil ellipse fitting, precise boundary detection of the iris, and 3D reconstruction of the eye. This method integrates image processing with geometric modeling, employing pseudo-polar coordinate transformations to rasterize the iris. Temporal coherence is then used to refine estimations across sequential frames, enhancing robustness and stability.

Finally, the 3D coordinates of the iris and eyeball centers are calculated using the known camera parameters and anatomical constraints. These coordinates are transformed into a head-mounted display (HMD) coordinate system, and the gaze vector is computed as a line from the eyeball center to the iris center, further corrected for the angle between the optical and visual axes. The final outputs of the method are normalized gaze vector and physical positions of eyeball and iris centers measured in millimeters.

The key innovations of this method lie in its ability to:

- Reliably detect and utilize the iris as a stable tracking feature, enabled by introduction of a novel pseudo-polar iris rasterization.
- Using the iris as the primary tracking feature makes it possible to tolerate optical distortions and occlusions without requiring specialized lighting or neural network-based processing.
- Maintain computational efficiency suitable for real-time applications in resource-constrained environments.

B. ROUGH PUPIL POSITION ESTIMATION

The first step of the proposed eye tracking method is to detect the rough position of the pupil. Although the pupil is not used as the primary reference for gaze estimation, its high contrast with the surrounding features makes it an ideal starting point. This stage is designed to provide a computationally efficient and noise-resilient estimate of the pupil location.

The input image is first processed to eliminate small, irrelevant features such as glints, eyelashes, and noise while preserving the edges of larger, critical features, such as the pupil and iris. This is achieved using a fast edge-preserving filter that enhances the robustness of the subsequent steps. The filter used is a simplified separable bilateral-like implementation. The mean values of the neighboring pixels before and after pixel being filtered are used to clamp its value. Two passes performed: vertical and horizontal. This preprocessing step ensures the system remains robust under diverse lighting conditions and variable image qualities.

To segment the pupil from the surrounding regions, the intensity histogram of the preprocessed image is analyzed by first computing the cumulative distribution of pixel intensities, selecting a threshold based on the minimal expected area of the pupil, and flagging pixels with intensities below this threshold as potential pupil regions, creating a binary image. Such adaptive thresholding accounts for variations in lighting and pupil contrast, enabling the method to remain effective across various setups.

The threshold image undergoes a two-pass process to refine the estimation of the pupil position:

First pass: Computing weighted vertical segments:

- Continuous vertical segments of flagged pixels are identified in the binary image.
- Each segment is assigned a weight proportional to the gradient value at its boundaries in the original image. The weight reflects the likelihood of the segment

corresponding to a sharp edge characteristic of the pupil. Non-pupil areas, which typically lack sharp edges, are therefore assigned lower weights which minimizes false positives.

Second pass: Finding maximum-area spot via horizontal sum:

- The horizontal sums of the weighted vertical segments are computed.
- The pixel with the maximum sum is selected as the approximate pupil center.

The pixel identified in the second pass is considered the rough center of the pupil and serves as the input for subsequent stages, including iris detection and 3D reconstruction.

The use of lightweight filtering and fast pupil location detection ensures the system operates in real time. Adaptive thresholding and gradient-based weighting make the method resilient to noise and false-positive detections.

C. ELLIPSE FITTING TO THE PUPIL EDGE

Once the rough position of the pupil has been determined, the next step involves modeling its contour through ellipse fitting. This process refines the pupil location and geometry, enabling accurate segmentation and serving as a foundation for subsequent iris-based calculations. The ellipse fitting procedure is designed to handle partial occlusions, noise, and variability in the input data while maintaining computational efficiency.

Based on the rough center of the pupil, a region of the pupil area is extracted from the input image. The initial pupil area is defined as a set of flagged pixels within a rectangular region centered around the identified rough pupil position, with a size sufficient to encompass the entire pupil. This is followed by dynamic expansion. Additional flagged pixels near the initial area are included if they fall within a predefined distance threshold. This step accounts for potential additional pupil segments separated by occlusions or contrast variability. The dynamic expansion step is repeated until no additional pixels flagged.

Subsequently, the pupil area undergoes contour extraction to identify the pupil's boundary as a set of edge points. The contour of the area surrounding the center is extracted as a list of points. The point is considered to belong to a contour if it has non-flagged immediate neighbor points.

The obtained point list is filtered: points creating inward-curved edge, which are unlikely to belong to the convex pupil boundary, are removed from the list. The removal is repeated multiple times, possibly shrinking the list with each iteration. This ensures that the contour data reflects the true edge of the pupil, free from artifacts and noise.

A direct least-squares ellipse fitting algorithm from [3] is applied to the filtered contour points to model the pupil shape. To handle partial occlusion, fitting is performed on multiple combinations of point subsets, selecting the two largest continuous groups of points. The choice of two groups is supported by the typical situation in which the pupil is partially occluded by the top, bottom or both eyelids. The fitting parameters that minimized the geometric error between the ellipse and the full list of boundary points are selected.

Output of the procedure: The ellipse's center, major and minor axes, and rotation angle are computed from algebraic ellipse parameters.

The ellipse parameters are checked against the expected ranges for pupil size and shape and consistency with the detected pupil center.

D. REFRACTION AND PERSPECTIVE CORRECTION FOR TRUE PUPIL ELLIPSE PARAMETERS

Accurate pupil position estimation requires correction of the distortion caused by corneal refraction of light as it passes through the optical elements of the human eye. Displacements of features closer to or farther from the imaging plane should also be performed because of the camera perspective projection. Without accounting for these effects, the perceived pupil position and shape deviate from their true physical parameters, introducing systematic errors into the model.

The human cornea, with a refractive index of approximately 1.3375, causes light entering the eye to bend significantly. This refraction introduces two effects: (1) distortion of pupil shape: the apparent shape of the pupil deviates from its true geometry, appearing as an expanded ellipse in 2D image projections. (2) Offset of the pupil center projection on the image plane. The center of the apparent pupil shifts relative to its true position owing to the bending of light rays.

Refraction and perspective corrections are critical for mitigating these effects and ensuring that the model accurately represents the true physical parameters of the eye.

The refracted (observed) ellipse axes ratio (φ_{refr}) from the true ellipse axes ratio (φ_{corr}) can be derived using equation 3.5a given in [4]:

$$\varphi_{refr} = 0.99 \cdot cos\left(\frac{acos(\varphi_{corr})+5.3}{1.121}\right). \quad (1)$$

For near-to-eye camera, it is also important to account for perspective projection effects. The camera ray hitting imaging plane at some point (x, y) is not perpendicular to this plane. This results in the observed ellipse ratio deviating from the true ellipse ratio (observed from infinity). The camera ray angle should be projected onto a plane coinciding with the minor ellipse axis and perpendicular to the imaging plane to obtain the angle to the minor ellipse axis Θ:

$$\theta = atan\left(\frac{\sqrt{x^2+y^2}}{f_{px}} cos\,(\beta)\right), \quad (2)$$

where angle between ellipse minor axis and the vector from the center of the imaging plane towards (x, y) is denoted by β. Tangent of the camera angle is obtained by dividing the distance $\sqrt{x^2+y^2}$ from the imaging plane center, by the camera focal length expressed in pixels: $f_{px} = W/[2 \cdot tan(HFOV/2)]$, where W is the frame width and $HFOV$ is the camera horizontal field of view. Multiplying

by $cos\,(\beta)$ yields the tangent of the angle between the projected camera ray and the ellipse minor axis.

Factoring in the correction of the camera perspective projection, ignoring the 5.3 constant angle (which is incorporated into the visual axis angle 'α' later on) and neglecting the 0.99 scaler, equation (1) for the corrected ellipse ratio can be re-written as:

$$\varphi_{corr} = cos\big[acos\big(\varphi_{refr} \cdot cos(\theta)\big) \cdot 1.121 - \theta\big]. \quad (3)$$

Multiplying the observed ratio by $cos(\theta)$ corrects for the camera ray angle (effectively transforming the image plane to the equivalent view under perpendicular rays), while subsequently subtracting θ rotates the un-refracted angle back into the image plane. The minor ellipse axis is then recalculated as the major axis (unchanged) multiplied by the corrected ratio φ_{corr}.

The pupil center shift (Δ, in millimeters) from the true pupil ratio due to refraction can be approximated by the following empirical formula:

$$\Delta_{refr} = acos(\varphi_{corr}) \cdot 0.32 + [acos(\varphi_{corr}) \cdot 0.58]^2. \quad (4)$$

The above equation was obtained by performing a ray-tracing simulation in Zemax optical design software and fitting a second-order polynomial across different observation angles. The simulation used a pupil diameter of 4mm and refraction index of 1.3375.

Furthermore, the pupil is slightly closer to the eyeball center (approximately 0.3-0.7mm, depending on the subject and accommodation state [4]) than the iris-sclera boundary. When captured at an angle, it causes iris and pupil center separation on the imaging plane. This introduces additional pupil center shift Δ_{persp}:

$$\Delta_{persp} = (P_{eye} - I_{eye}) \cdot sin[acos(\varphi_{corr})], \quad (5)$$

where I_{eye} and P_{eye} are distances between eyeball center and centers of the iris and pupil, respectively.

Individual anatomic decentration of the eye pupil should also be accounted for.

Note that the above corrections to the detected pupil pose only improve the central point placement and scaling factors for the pseudo-polar iris rasterization in the next step and do not affect the precision of the estimation of the eye pose.

E. IRIS POSE ESTIMATION IN 2D IMAGE SPACE

The iris is a stable anatomical feature with a fixed size and consistent geometry, making it a reliable reference for estimating eye pose. The next step of the method is to detect the iris boundary (limbus) in the 2D image space, which includes rasterization in the pseudo-polar coordinate space, boundary detection, and final pose calculation.

To localize the iris, the input image is scanned in pseudo-polar coordinates, a transformation centered on the corrected pupil ellipse. The 'pseudo' here reflects the fact that the scan is performed along a nearly elliptic path rather than a circular path, accounting for pupil ellipse ratio and further adjusted for the perspective projection of the camera.

RASTERIZATION DETAILS

- The rasterization origin is the pupil center (after refraction correction).
- Radial scanning traces concentric ellipses outward from the pupil edge towards the sclera, providing the initial (x, y) pixel coordinates in the camera frame.
- The (x, y) coordinate is further corrected for the camera perspective by first computing the height (h) of the rasterized point (height of projection of the iris point in 3D space to the image plane) above (or below) the imaging plane:

$$h = sign \cdot R \cdot \sqrt{1 - \left(\frac{r}{a}\right)^2}, \quad (6)$$

where R is the radial distance of the scan (horizontal offset in the rasterized iris image), r is the distance from the pupil center to (x, y), a denotes ellipse semi-major axis, and *sign* determines whether the point is above or below the imaging plane and is determined by measuring the angle between major ellipse axis and the angle of the scan: if it is above $\pi/2$ and below $3\pi/2$: +1; otherwise: -1.

The coordinates are then adjusted according to the height and camera focal length (derived from the right triangle formed by the camera ray, height of the iris point h, and image plane):

$$\Delta x = \frac{h \cdot x}{h + f_{px}}. \quad (7)$$

Algorithm 1 outlines the iris rasterization procedure. A practical choice of the angular sector is $[-\pi/6, \pi/6]$. The scaling factor 'scl' is computed first; it specifies how the coordinate vector must be scaled as a function of angle, to transform the unit circle into the target ellipse. The image plane coordinates (x, y) are then obtained from the iris center (x0, y0), and each raster pixel at (r, α) is assigned accordingly. θ and φ are the ellipse parameters (rotation and axes ratio), and Δx, Δy are calculated according to the equation (7).

Algorithm 1 Pseudo-polar iris rasterization

Input: CameraFrame
Output: Raster

for r = min_iris **to** max_iris **do**
  for α = -π/6 **to** -π/6 **do**
    γ ← **atan2**[**sin**(α - θ), φ·**cos**(α - θ)]
    scl ← $\sqrt{\mathbf{cos}(\boldsymbol{\gamma})^2 + (\boldsymbol{\varphi} \cdot \mathbf{sin}(\boldsymbol{\gamma}))^2}$
    x ← **x0** + **r** · **scl** · **cos**(α) - Δx
    y ← **y0** + **r** · **scl** · **sin**(α) - Δy
    Raster(**r** - **min_iris**, α + π/6) ← **CameraFrame**(x, y)
  end for
end for

OUTPUT

Two raster images are generated: one for the sector of the iris towards the left of the pupil and one towards the right of the

pupil (Fig.1). The raster images encapsulate the brightness profile that corresponds to the transition at the iris-sclera boundary (Fig.2).

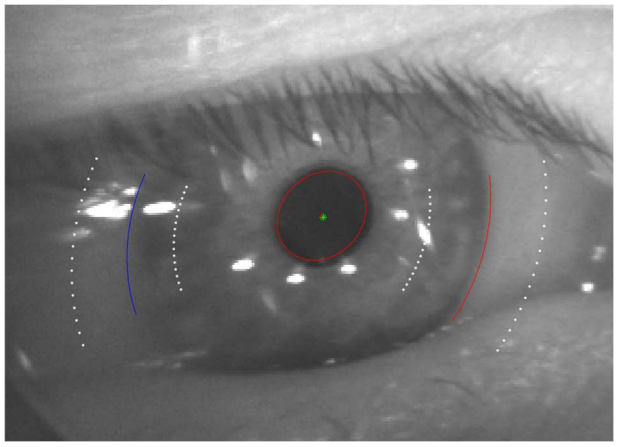

FIGURE 1. Left and right areas of the iris scans mapped onto the input frame with detected limbus boundaries drawn (white dots delimit the start and the end of the rasterized region). Note the ellipse deviation from the observed pupil position – this represents pupil pose corrected for the refraction and decenter.

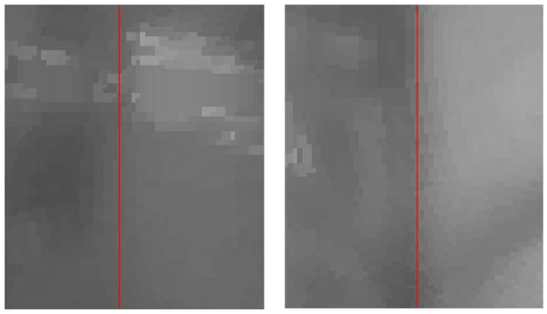

FIGURE 2. Iris scans after the pseudo-polar rasterization (left and right sectors, respectively).

The boundary between the iris and sclera is identified by analyzing the brightness gradient in the raster images. A sudden step (marked with red line in Fig.2) in brightness indicates the transition from the iris to the sclera (limbus). If no step above the threshold is found, the iris is considered to be overly occluded, and the iris pose estimation from this particular raster scan is discarded.

The iris radius can be computed from the location of the brightness step in the rasterized iris scans. The dynamic pupil decenter in the horizontal direction is estimated by computing the iris radius difference from both left and right iris scans. The dynamic pupil decenter in the vertical direction is estimated from the slight deviation of the step profiles from the vertical direction (slant). From this, the precise center of the iris and its radius in pixels is determined. Finally, the bow-like shapes of the rasterized brightness step boundaries can be used to precisely adjust the remaining iris ellipse parameters (minor/major axes and rotation angle). This information provides a precise 2D pose of the iris in the image space.

F. IRIS CENTER LOCATION IN 3D SPACE

Accurately determining the centers of eye features in 3D space is essential for reliable gaze estimation and 3D eye pose tracking. The human iris has a near-constant diameter of approximately 11.6 mm across individuals ([12], section '2.2.3 The Iris'), with minimal variation due to age and ethnicity. This anatomical feature provides a robust reference for determining the 3D position of the iris center.

The 2D iris center is projected into 3D space using the following steps:

- A projection ray is computed from the normalized image coordinates through the camera's optical center and the 2D iris center. This ray represents the possible locations of the iris center in 3D space along the line of sight.
- The actual 3D position of the iris center is determined by intersecting the projection ray with a virtual plane at a depth consistent with the iris diameter. Scaling by the known iris diameter ensures that the reconstructed iris center matches the physical size and position of the iris in space.
- Temporal Smoothing. When multiple frames are available, the position of the iris center is averaged over time using a weighted smoothing function to reduce jitter.

G. EYEBALL CENTER ESTIMATION

The eyeball center serves as a stable reference point, and its location relative to the iris center allows for precise estimation of the gaze direction. This section outlines the process for estimating the eyeball center in both the 2D image and 3D physical spaces, leveraging geometric modeling and temporal data integration.

The human iris is anatomically fixed at a consistent distance of approximately 8 mm (key eye parameters from here on are derived from [10, Fig.2] if no other reference is provided) from the center of the eyeball. This constant distance enables the estimation of the eyeball center based on the 3D position of the iris.

The key assumptions are as follows:

- The eyeball approximates a perfect sphere with a diameter of ~24 mm.
- The iris is positioned on the surface of the eyeball.

ESTIMATION IN 2D IMAGE SPACE

When only a single input frame is available, the eyeball center is estimated in the 2D image space using the geometric relationship between the iris and eyeball. Using the known offset of the iris center from the eyeball center, a 2D projection is made, utilizing the iris ellipse ratio to approximate the eyeball's center in 2D.

If multiple input frames were already observed, the individual estimations of the eyeball center in 2D space can be averaged (with weighting proportional to how confident the detection was in each particular frame). It is worth noting that there is no unique center of rotation because of the positioning and operation of the extra-ocular muscles [4]. The centers of rotation for the horizontal and vertical movements of the eye are at different distances from the iris: approximately 12.5mm

and 15.3mm behind the cornea, respectively ([16], section 'Results'). This fact should be accounted for, and stabilization should be performed separately for the horizontal and vertical coordinates of the eyeball center.

ESTIMATION IN 3D SPACE

When the iris center is reconstructed in 3D space, the eyeball center can be estimated directly through back-projection. The eyeball center is positioned along the ray extending from the iris center through a 3D plane tangent to the eyeball surface. A fixed 8 mm offset is used to accurately locate the center. When multiple input frames are available, the eyeball center estimates are averaged over the frames. Confidence weighting is applied to assign greater importance to frames with high-quality iris detections. A similar separate per-coordinate approach as in 2D stabilization is applied owing to the absence of a unique center of rotation.

H. GAZE VECTOR ESTIMATION

The final step in the eye tracking pipeline is the computation of the gaze vector, which determines the direction of the user's gaze relative to the coordinate system of the HMD or other reference frames. The gaze vector is derived from the spatial relationship between the eyeball center and iris center, both of which have been accurately estimated in the previous steps.

To compute the gaze vector in the context of an HMD, the 3D coordinates of the eyeball and iris centers are transformed into the device's coordinate system.

The gaze vector is defined as a directional vector originating from the eyeball center and pointing toward the iris center. However, the visual axis of the eye does not coincide with the optical axis [4]. The horizontal and vertical angular differences between these axes (angle α) should be determined during the individual eye-tracking calibration. Alternatively, typical 5° nasal and 2° upward can be used as a fallback [4]. In summary, gaze vector computation involves the following steps:

- Compute the direction vector. The direction vector is computed as the difference between the 3D coordinates of the iris and eyeball centers given in the device coordinate system.
- Normalize. The gaze vector is normalized to unit length for consistency across devices and applications.
- Adjust for the visual axis angle α. The gaze vector is multiplied by the rotation matrix computed from α.

I. CALIBRATION

The key eye model parameters introduced above, such as rotation distances, pupil decentration, and the angle α, can vary slightly across subjects. A method benefits from subject-specific calibration of these parameters.

A five-point calibration is recommended, with reference targets positioned at the center, and at locations directly above, below, left, and right of the center. This configuration enables estimation of the following eye model parameters:

- Centers of horizontal and vertical eye rotation: derived from the geometric relationship between the iris center positions registered on the image plane and the known angular offsets of the reference targets.
- Pupil parameters (decentration and non-circularity): obtained by comparing the detected pupil ellipse parameters with the iris ellipse parameters estimated via pseudo-polar rasterization.
- Angle α (between the visual and optical axes): computed by comparing the uncorrected gaze vector obtained during calibration with the ground-truth gaze direction toward each reference target.

A reference MATLAB implementation of the complete calibration procedure is available in [17].

III. RESULTS

A. 3D EYE POSE RECONSTRUCTION PERFORMANCE

GROUND TRUTH DATASETS

The NVGaze dataset [8], [9] with ground-truth gaze directions for real-world eye images and full 3D eye pose ground truth for synthetic eye images was utilized to assess the accuracy and precision of the proposed method.

The subset used for gaze direction estimation is: "Off-axis camera view inside VR headset (HTC Vive Pro + PupilLabs cameras)" with an off-axis camera position. The subset includes images from subjects with variations in gender, ethnicity, age, and eye shape. Some of the subjects include eyeliner, eyeshadow, mascara, eyeglasses, and contact lenses. For each subject, the data includes varying gaze directions and pupil sizes.

The NVGaze synthetic dataset was used to assess the 3D eyeball center position estimation. We had to resort to synthetic data because this is the only dataset providing the true ground truth eyeball 3D center position and not the estimated one [13].

Eye tracking performance assessment (precision and accuracy) was performed according to the methodology outlined in the Tobii whitepaper [5] for VR/AR headsets and wearables.

GAZE ACCURACY AND PRECISION

The gaze estimation accuracy and precision over nine subjects (the sets with enough frames to perform calibration over) are summarized in Tables 1 and 2, respectively. According to the recommendations in [5], precision is measured as the root-mean-square deviation (RMSD) of the angular differences between the mean gaze direction and each gaze direction for each frame. The accuracy is measured as the angular difference - the offset (in degrees) - between the estimated and actual directions.

The achieved average accuracy is within 2° to 3° for the majority of the subjects, with some exceptions.

For example, subject '09' represents a difficult case for the method. It has high levels of pupil occlusion of over 50% due

to unusually close-to-camera eye placement and the presence of correction glasses.

The accuracy and precision of the method are comparable to those of the model-fitting [1] and the PCCR method [14]. Although not a direct comparison, a similar angular gaze error of 1.68° is reported in [1].

The last column of Table 1 is the 'generalization error' as defined in [8]. I.e., the absolute gaze error between the ground truth and the estimated values after applying a per-subject affine calibration transform. The method achieves a generalization error of 2.35°, which is comparable to the 2.1°-3.1° error range reported for the neural network-based approach in [8], evaluated on the same dataset.

TABLE 1
GAZE DIRECTION ACCURACY,
DIFFERENCE FROM GROUND TRUTH IN DEGREES.

Subject	FoV degree range from center				Average, full FoV	Gen. error
	0-10	10-20	20-25	25-30		
01	1.19	1.57	2.13	2.90	1.93	1.40
02	1.46	2.66	3.30	3.51	2.77	1.32
03	1.45	2.69	2.57	2.71	2.51	2.30
04	1.33	1.73	2.20	2.78	1.94	1.26
05	1.59	1.72	2.12	2.77	1.94	1.55
06	1.91	2.17	2.78	2.68	2.39	1.72
07	2.43	3.87	2.99	4.77	3.64	2.96
08	4.16	3.57	3.49	3.89	3.74	2.75
09	6.76	7.82	6.49	7.32	7.17	5.89
All	2.48	3.09	3.12	3.70	**3.11**	**2.35**

TABLE 2
GAZE DIRECTION PRECISION,
RMSD FROM GROUND TRUTH IN DEGREES.

Subject	FoV degree range from center				Average, full FoV
	0-10	10-20	20-25	25-30	
01	0.62	0.90	0.98	1.22	0.95
02	0.84	0.97	0.95	1.19	0.99
03	1.05	1.72	2.17	2.36	1.84
04	0.80	0.90	0.87	0.99	0.89
05	1.07	1.22	1.43	1.78	1.33
06	1.19	1.52	1.49	1.36	1.45
07	1.48	1.94	1.66	1.45	1.79
08	1.73	1.66	1.86	1.93	1.77
09	1.83	1.75	1.90	2.57	1.99
All	1.18	1.40	1.48	1.65	**1.44**

TABLE 3
DATASET PROPERTIES AND ACHIEVED ACCURACY.

Subject	% of iris reflectivity	Eyewear presence	% of frames >1/5th pupil occluded	Accuracy
01	22	N	1	1.93
02	29	N	2	2.77
03	35	N	2	2.51
04	35	Y	0	1.94
05	26	N	0	1.94
06	14	N	0	2.39
07	41	Y	12	3.64
08	26	Y	0	3.74
09	31	Y	25	7.17

Per-subject dataset properties (iris reflectivity, presence of eyewear, and the percentage of frames in which more than one-fifth of the pupil area is occluded) together with the achieved accuracy, are summarized in Table 3. Because of the limited number of participants, definitive trends cannot be established; nevertheless, several observations can be made:

- No statistically significant dependence of gaze estimation precision on the iris lightness was detected. This is likely due to the lower variability of the iris lightness in the IR wavelength range compared with the visible range.
- The presence of eyewear is not a significant factor in the method accuracy (mean error: 2.3° without eyewear vs. 2.8° with eyewear).
- A persistently high level of pupil occlusion leads to a significant reduction in accuracy.

EYEBALL CENTER 3D POSITION ACCURACY AND PRECISION

To assess the eye center position estimation precision as well as the robustness of the method to 'slippage' (displacement of near-eye display over user's head over time), a synthetic dataset, generated with NVGaze dataset tools, with sinusoidally varying position of the eyeball center in 3D space was used. Both the eye gaze direction and eyeball position varied simultaneously. The speed of eyeball displacement was set to 1 mm/s. The achieved eyeball-center estimation accuracy under such harsh conditions is 0.81 mm.

Fig.3 illustrates the track of the eye gaze. A total of 1600 camera frames were generated, which amounts to approximately 27 seconds at 60fps. Fig.4 depicts estimated eyeball 3D position vs true position during these 27 seconds.

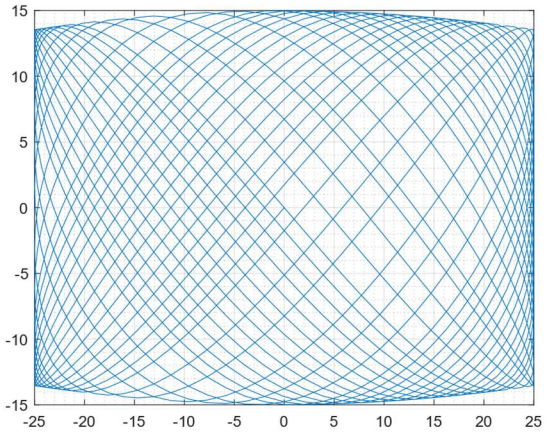

FIGURE 3. Eye gaze direction coverage (degrees horizontally/vertically from the center) during 3D eyeball position slippage estimation test.

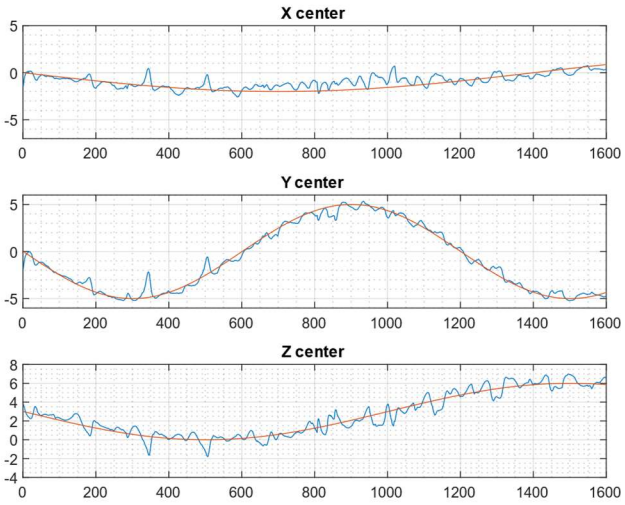

FIGURE 4. Estimated eyeball 3D position (slippage) vs true position. Vertical axis is in mm, horizontal axis – frame number.

B. COMPUTATIONAL EFFICIENCY

The method was implemented in both MATLAB and native C code [17]. The C version uses single-precision floating-point arithmetic and runs single-threaded. To enhance computational efficiency, it leverages compiler-assisted auto-vectorization (e.g., GCC versions 12 and above).

The majority of the computational cost is incurred during the initial eye-pupil detection stage because these operations require a full scan of the input frame. The pseudo-polar iris scan generation and analysis is quite fast because the region of the iris is well-defined after initial pupil detection.

In terms of memory footprint, the method requires less than twice the size of the input frame for the intermediate buffer storage. For example, with a 640 x 480 input frame, the total memory allocation is approximately 450 KB.

Overall, the method achieves the following impressive processing times:

- On a 3GHz 12th generation Intel CPU the single 640 x 480 frame processing takes 350µs on a single core.
- On XR-specific **Snapdragon XR2**, 640 x 480 camera frame: 465µs on a single core. This means that **stereo eye tracking at a 120 Hz** refresh rate utilizes under **2% of the total CPU** at less than **0.5ms latency**.

Compared with the published PCCR performance figures [14] (reported at 4-8 ms per frame), the proposed implementation achieves a lower, sub-millisecond processing time. However, a well-optimized PCCR implementation should achieve comparable performance on modern hardware, given the simple geometric relationship between glint position and gaze direction [15].

No direct comparison with neural network-based methods is possible, as those are typically executed on GPUs or other specialized hardware. Nevertheless, based on the processing-time reported in [8] (496µs to estimate gaze, excluding pupil localization) and the approximate difference in computational throughput between a 12th-generation Intel CPU (~16 GFLOPS per core) and an Nvidia Titan V GPU (~12 TFLOPS), the proposed method's computational requirements are roughly three orders of magnitude lower.

IV. DISCUSSION

The proposed iris-based eye tracking method demonstrates significant advancements in the field, paving the way for fast and robust 3D eye-pose estimation. This section evaluates the strengths of the method, addresses its limitations, and explores future directions.

A. GENERAL OBSERVATIONS

The method performs best when using an off-axis camera. This is due to the fact that a very minor change in the ratio of the iris ellipse axes when the eye is looking directly into the camera can lead to a significant variation in the estimated iris angle. Consequently, such positioning introduces a high level of noise in the estimation of the eyeball center location. For on-axis camera locations, PCCR-based methods are generally more suitable.

Eye tracking systems that use an on-axis, through-the-lens camera location tend to produce lower accuracy in estimating the 3D position of the eyeball. This is due to the fact that on-axis, through-the-lens systems require high focal lengths (effectively narrow FOV). In such setups, the eye image change little with variations in the distance between the eye and camera, increasing the error in distance registration.

Although the method may lag slightly behind PCCR-based approaches in simple scenarios (e.g., users without eyewear and under constant illumination), it demonstrates superior precision and robustness under challenging lighting conditions where dynamic pupil decentering may occur or when eyewear is present. Eyewear can generate double reflections and displace reflection locations, which often confuse PCCR methods.

B. STRENGTHS OF THE PROPOSED METHOD

LEVERAGING ANATOMICAL CONSTANCY

The fixed geometry and size of the iris, irrespective of illumination or physiological changes, provide a robust basis for tracking. Unlike the pupil, which varies in size and position, the iris offers consistent features for accurate detection and pose estimation. The partial visibility of the iris is managed through the selection of appropriate areas for analysis and temporal data integration, enabling robust tracking even in scenarios where eyelids or eyelashes occlude parts of the iris.

ROBUSTNESS ACROSS CONDITIONS

By eliminating the need for glints or specialized lighting, this method is resilient in diverse environments, making it suitable for wearable devices. The method remains effective for users wearing prescription glasses or contact lenses, which is a common limitation of glint-based approaches.

COMPUTATIONAL EFFICIENCY

The lightweight computational framework ensures that the system operates in real time, even on resource-constrained platforms such as standalone HMD devices. Compared to neural networks-based eye tracking, the proposed method achieves at least two orders of magnitude reduction in computing resources.

C. CHALLENGES AND LIMITATIONS

SENSITIVITY TO IRIS CONTRAST VARIABILITY

The method's reliance on iris boundary detection renders it sensitive to contrast variability among individuals (e.g., lighter irises). Future studies could incorporate adaptive preprocessing techniques to dynamically normalize contrast variations.

PARTIAL IRIS VISIBILITY IN EXTREME CASES

Although the method performs well under moderate occlusions, extreme cases in which less than 10% of the iris edge is visible result in reduced accuracy. Similarly, as detailed in Table 3, severe pupil occlusion produces a comparable decrease in performance.

CALIBRATION DEPENDENCY

As with any other camera-based eye tracking method, per-individual eye tracking calibration is required to obtain the most precise results. However, the method is resilient to the common problem of slippage and does not require recalibration upon HMD repositioning on the user's face.

ASSUMPTION OF FIXED ANATOMICAL DIMENSIONS

The method relies on average anatomical dimensions (e.g., 11.6 mm iris diameter) for 3D calculations. Individual deviations from these averages may introduce errors. An individual calibration procedure mitigates this issue.

D. COMPARISON WITH EXISTING METHODS

PUPIL-BASED APPROACHES

The proposed method avoids the inaccuracies associated with pupil size variability and decentering, ensuring consistent 3D position estimation. In contrast, pupil-based methods often benefit from higher contrast, having an edge under difficult lighting conditions.

GLINT-BASED TECHNIQUES

Glint-based methods (PCCR) excel in controlled environments; however, falter under natural lighting and with optical wear. The iris-based eye tracking method offers a clear advantage in such scenarios utilizing robust iris edge detection through pseudo-polar rasterization.

NEURAL NETWORK APPROACHES

Neural network eye-tracking methods can handle complex eye imagery and adapt to various lighting conditions; however, they come at a very high computational cost. The iris-based approach achieves similar or better accuracy levels with significantly lower processing requirements.

E. FUTURE DIRECTIONS

The following directions of research may further enhance capabilities of the method:

- **Adaptive preprocessing**. The dynamic adjustment of parameters for contrast enhancement and noise reduction can improve the robustness of pupil area detection across diverse user groups and environments.
- **Enhanced 3D modeling**. Integration with advanced eye models that account for individual anatomical variations and dynamic behaviors could further refine 3D pose estimation.
- **Better temporal data utilization**. Advanced temporal models could enhance predictions in cases of transient occlusions or rapid eye movements.
- **Binocular tracking**. Extending the method to track both eyes simultaneously, sharing the intermediate data, would improve precision and enable stereo-based depth estimation.

V. CONCLUSION

The proposed iris-based eye tracking method offers a precise approach to eye pose estimation by combining anatomical insights with computational efficiency. Highly optimized low-complexity implementation allows for sub-millisecond processing times, even on low-power mobile CPUs. Precise eye position tracking enhances the user experience in AR/VR systems, enabling interaction and dynamic improvements in optimal scene rendering. The method's compatibility with near-eye displays and very low computational footprint make it particularly suitable for such applications.

REFERENCES

[1] L. Swirski and N. A. Dodgson. A fully-automatic, temporal approach to single camera, glint-free 3D eye model fitting. *Proceedings of ECEM*, 2013.

[2] F. Manuri, A. Sanna, C. P. Petrucci. PDIF: Pupil Detection After Isolation and Fitting. *IEEE Access*, vol. 8, pages 30826–30837 Feb 19 2020. doi: 10.1109/ACCESS.2020.2973005.

[3] V. Pratt. Direct Least-Squares Fitting of Algebraic Surfaces. *Computer Graphics (ACM)*, vol. 21, no, 4, pages 145–152, 1987. doi: 10.1145/37402.37420.

[4] D. A. Atchison and G. Smith. Optics of the Human Eye. *2nd edition, CRC Press*, 2023. doi: 10.1201/9781003128601.

[5] Eye tracking performance assessment — for VR/AR headsets and wearables. *Tobii white paper*, October 2023, Version 1.1. https://www.tobii.com/resource-center/reports-and-papers/eye-tracking-performance-assessment

[6] M. Tonsen, X. Zhang, Y. Sugano and A. Bulling. Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. *In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 139–142, 2016. doi: 10.1145/2857491.2857520.

[7] R. Kothari, Z. Yang, C. Kanan, R. Bailey, J. B. Pelz, and G. J. Diaz. Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities. *Scientific Reports*, vol. 10, no. 1, page 2539 (2020). doi: 10.1038/s41598-020-59251-5.

[8] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, D. Luebke. NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye Gaze Estimation. *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI) 2019*, May 2019. doi: 10.1145/3290605.3300780.

[9] E. Wood, T. Baltruaitis, X. Zhang, Y. Sugano, P. Robinson, A. Bulling. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. doi: 10.1109/ICCV.2015.428.

[10] P. Nogueira, M. Zankl, H. Schlattl, P. Vaz. Dose conversion coefficients for monoenergetic electrons incident on a realistic human eye model with different lens cell populations. *Physics in Medicine & Biology*, vol. 56, no. 21, pages 6919–6934, Nov 7 2011. doi: 10.1088/0031-9155/56/21/010.

[11] H. Gross, F. Blechinger, and B. Achtner. Handbook of Optical Systems, Volume 4: Survey of Optical Instruments. *Wiley-VCH Verlag GmbH*, 2008. doi:10.1002/9783527699247.

[12] G. Francois, P. Gautron, G. Breton, and K. Bouatouch. Image-Based Modeling of the Human Eye. *IEEE transactions on visualization and computer graphics*, vol. 15, no. 5, pages 815–827, Sep-Oct 2009. doi: 10.1109/TVCG.2009.24.

[13] W. Fuhl, G. Kasneci, and E. Kasneci. TEyeD: Over 20 million real-world eye images with Pupil, Eyelid, and Iris 2D and 3D Segmentations, 2D and 3D Landmarks, 3D Eyeball, Gaze Vector, and Eye Movement Types. *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Bari, Italy, 2021, pages 367-375, doi: 10.1109/ISMAR52148.2021.00053.

[14] S. Goñi, J. Echeto, A. Villanueva, and R. Cabeza. Robust Algorithm for Pupil-Glint Vector Detection in a Video-oculography Eyetracking System. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Cambridge, UK, 2004, pp. 941-944 Vol.4, doi: 10.1109/ICPR.2004.1333928.

[15] E. D. Guestrin and M. Eizenman, General theory of remote gaze estimation using the pupil center and corneal reflections, *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1124-1133, June 2006, doi: 10.1109/TBME.2005.863952.

[16] A. Ohlendorf, F. Schaeffel, and S. Wahl. Positions of the horizontal and vertical centre of rotation in eyes with different refractive errors. *Ophthalmic Physiol Opt.* 2022 Mar; 42(2):376-383. doi: 10.1111/opo.12940.

[17] D. Shmunk, 2025, “Fast, Iris-based eye tracking, implemented in MATLAB and native C”, GitHub, https://github.com/almalence/eyetracking

DMITRY SHMUNK received the M.Sc. degree in physics with minor in IT from Novosibirsk State University, in 1998. He is currently a Chief Technical Officer with Almalence, Tashkent, Uzbekistan. He is also an IEEE member and SPIE lifetime member.