Unwrapping the Eye for Visible-Spectrum Gaze Tracking on Wearable Devices

Anonymous WACV submission

Paper ID 124

Abstract

Wearable devices with gaze tracking can assist users in many daily-life tasks. When used for extended periods of time, it is desirable that such devices do not employ active illumination for safety reasons and to minimize interference from other light sources such as the sun. Most non active-illumination methods for gaze tracking attempt to locate the iris contour by fitting an ellipse. Although the camera projection causes the iris to appear as an ellipse in the eye image, it is actually a circle on the eye surface. Instead of searching for an ellipse in the eye image, the method proposed in this paper searches for a circle on the eye surface. To this end, the method calibrates a three-dimensional eye model based on the location of the corners of the eye. Using the 3D eye model, an input image is first transformed so that the eye’s spherical surface is warped into a plane, thus “unwrapping” the eye. The iris circle is then detected on the unwrapped image by a three-step robust circle-fitting procedure. The location of the circle corresponds to the gaze orientation on the outside image. The method is fast to calibrate and runs in real-time. Extensive experimentation on embedded hardware and comparisons with alternative methods demonstrate the effectiveness of the proposed solution.

1. Introduction

Wearable gaze tracking devices that can be used in daily-life have many applications. By giving insight into the user’s perception and intent, such devices can help prevent human error when driving or operating complex machinery, improve the quality of life for users with reduced cognitive abilities, and facilitate human-computer interaction.

Recent video-based eye tracking (video-oculography) systems can be divided into appearance-based and shape-based methods:

- Appearance-based methods use the entire eye image as a descriptor and map the image descriptor directly to gaze position [7]. Because they rely on the image intensity, these methods are particularly sensitive to changes in illumination and scene geometry (i.e., changes in eye or head position with regards to the camera.) and are less accurate when compared with shape-based methods.

- Shape-based methods track specific portions of the eye anatomy such as the corneal reflection [5], pupil contour [6], and iris contour (corneal limbus) [2]. Corneal reflection and pupil contour methods usually require infrared ray (IR) active illumination, whereas iris contours can be detected with visible spectrum imaging.

IR methods can achieve high-accuracy (less than one degree) but require active illumination and thus are susceptible...
to interference from other IR sources, namely the sun. This limits their use to (mostly) indoors activities. Additionally, IR methods are not desirable for extended daily use, especially for the elderly and children [4], even if the intensity of IR meets the safety standards.

Using visible spectrum images, on the other hand, requires vision techniques to determine the iris contour (or, less frequently, the pupil contour.) This is a challenging and open problem due to the complex anatomy of the eye. The major hurdles for visible-light eye tracking include occlusion introduced by the eyelids and eyelashes, presence of reflection and specularities introduced by arbitrary illumination, variability in the iris color between users, variability in the sclera texture due to changes in eye irrigation, and the fact that small changes in the camera location relative to the eye cause complex changes in the scene geometry.

Most visible-spectrum methods in the literature, model (implicitly or explicitly) the iris as an ellipse and map its center to the gaze position. One of the early visible-light based iris tracking methods proposed ellipse-fitting with a RANSAC scheme [7], but still lacked robustness to illumination change and occlusions. More recently, Wu et al. [10] improved robustness to occlusion by introducing a three-dimensional (3D) model that accounts for the eyeball, iris and eyelids. Although viable, this method requires tracking as many as seven parameters for every frame, which limits its robustness. In 2011, Tsukada et al. [9] used a 3D model to restrict the ellipse tracking, requiring the tracking of only two parameters at run-time. Although competitive with commercial methods, this method is sensitive to occlusion and requires a complex calibration of a large number of parameters. See [1] for a recent survey of the eye-tracking literature.

The method proposed in this paper relies on the observation that, although the iris appears as an ellipse on the image, it actually corresponds to a circle on the eye surface. Thus, the method replaces the traditional task of fitting an ellipse on the image, by the task of fitting a circle directly on the eye surface. This approach greatly reduces the complexity of problem by reducing the number of parameters in the search space (from 5 parameters for an ellipse to 3 parameters for a circle,) without diminishing the richness of the model. Because circle fitting requires less parameters, it is usually faster, more robust, and less prone to overfit the data, especially in degenerate conditions were only a portion of the iris is observed.

To calibrate the eye model, the proposed method only requires knowledge of the location of the eye corners in the image. Using this information, the method determines the location and radius of the eyeball sphere and creates a 3D model of the eye. At run-time, the image pixels corresponding to the eye surface are sampled and warped into a new image. This image is denoted as being the unwrapped image since it corresponds to taking the spherical surface of a sphere and “unwrapping” it into a plane (see Figure 1).

Iris detection is achieved by searching for a circle on this unwrapped image.

Circle estimation is done in three steps. First, a coarse initial estimate is obtained using a circular version of the Hough Transform; second, features from the circumference of the iris are extracted by searching radial paths starting from the iris center; and third, a robust method of circle fitting is used to estimate the circle that best fits the detected features. Once the location of the iris has been determined, the method creates a map that relates the iris location on the eye image with the gaze location on the outside image. The whole procedure is simple to implement, fast to calibrate, and runs in realtime.

2. Proposed Approach

Eye Model Figure 2 illustrates the eye model. The method represents the eyeball as a three-dimensional sphere and the iris as a two-dimensional circle on the sphere. Because the system is intended to be used with wearable devices (i.e., glasses, goggles or helmets), it assumes that the center the eyeball sphere is fixed. Let $X_C = [x_C, y_C, z_C]^T$ be the coordinate of the eyeball center in the camera coordinate system $(x, y, z)$.

On the eyeball center, the axes $(x_e, y_e, z_e)$ are defined so that the $x_e y_e$ plane is parallel to the image plane (i.e., $z_e$ is parallel to $z$.) The method assumes that the eye corners are on this plane and defines the $y_e$ axis so that it is aligned with them. Thus, the camera and the eyeball axes are related by a rotation about the $z$ axis, plus a translation from the center of the camera to the center of the eye. The angle $\gamma$ is defined as be the rotation along the $z$ axes between the camera and the eye axes as denoted in Figure 3.
Figure 3. Left: Rotation between the \(xy\) camera axes and the \(xe, ye\) eyeball axes. For clarity, the figure assumes that there is no translation between axis and that the \(z\) axis is collinear to \(ze\) with both axes pointing into the paper. Right: Parametrization of the surface of the eyeball sphere (the eye ball sphere is omitted from the figure for clarity.)

The angles \((\theta, \phi)\) parametrize the surface of the eyeball sphere (see Figure 3). Defining \(r\) to be the radius of the eyeball sphere, the \((xe, ye, ze)\) coordinates of a point on the surface of the eyeball sphere are:

\[
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix}
= R(\gamma) \begin{bmatrix}
  x_e \\
  y_e \\
  z_e \\
\end{bmatrix} + X_C \quad \Leftrightarrow \quad (1)
\]

\[
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix}
= \begin{bmatrix}
  \cos \gamma & -\sin \gamma & 0 \\
  \sin \gamma & \cos \gamma & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x_e \\
  y_e \\
  z_e \\
\end{bmatrix} + \begin{bmatrix}
  x_C \\
  y_C \\
  z_C \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_e \\
  y_e \\
  z_e \\
\end{bmatrix}
= \begin{bmatrix}
  cos \phi & 0 & -\sin \phi \\
  0 & 1 & 0 \\
  \sin \phi & 0 & \cos \phi
\end{bmatrix} \begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos \theta & -\sin \theta \\
  0 & \sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix}
  0 \\
  -r
\end{bmatrix}
\quad \Leftrightarrow \quad (2)
\]

The same point will have the following coordinates on the camera axes:

\[
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix}
= Rz(\gamma) \begin{bmatrix}
  x_e \\
  y_e \\
  z_e \\
\end{bmatrix} + X_C \quad \Leftrightarrow \quad (3)
\]

\[
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix}
= R(\theta, \phi, \gamma) \begin{bmatrix}
  0 \\
  0 \\
  -r
\end{bmatrix} + X_C
\]

where \(R(\theta, \phi, \gamma)\) is defined to be \(Rz(\gamma) R_y(\phi) R_x(\theta)\).

The eye model parameters can be divided into two sets. First, the parameters that describe the eye sphere: the rotation \(\gamma\) with relation to the camera, the center of the eyeball \((x_C, y_C, z_C)\), and the radius of the eyeball \(r\). And second, the parameters that describe the iris on the unwrapped \((\theta, \phi)\) plane, namely its location \((\theta_1, \phi_1)\) and radius \(r_1\). The objective of the calibration of the eye model is to determine the eye sphere parameters (namely, the \(\gamma\) angle, and the center and radius of the eyeball) which are assumed fixed.

3. Calibration of the Eye Model

The calibration method presented here only requires the knowledge of the location of the two eye corners in the image, denoted as \((u_L, v_L)\) and \((u_R, v_R)\). These locations can be obtained manually (by asking the user to click the eye corners,) or automatically (by, for example, using descriptive features tailored for the detection of the eye corners.) Note that, in the eyeball coordinate system the eye
corners have coordinates \((0, \pm r, 0)\).

Assuming that the intrinsic parameters of the eye camera are known, and letting \(K\) be the invertible matrix of camera parameters, one has the following relationship between the location of the corners in the image and in the eyeball coordinate system (where we assume the standard perspective camera model, use \((u_{LR}, v_{LR})\) to denote either the left or right corners, and have \(\lambda \neq 0\):

\[
\begin{bmatrix}
u_{LR} \\
v_{LR} \end{bmatrix} = \lambda K \begin{bmatrix}
\cos \gamma & -\sin \gamma & 0 \\
\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x_C \\
y_C \\
z_C
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} \tag{4}
\]

Defining \((u_{LR}', v_{LR}')\) to be such that:

\[
\begin{bmatrix}
u_{LR}' \\
v_{LR}' \end{bmatrix} = K^{-1} \begin{bmatrix}
u_{LR} \\
v_{LR} \end{bmatrix}
\]

one can write:

\[
\begin{bmatrix}
u_{LR}' \\
v_{LR}' \end{bmatrix} = \lambda \begin{bmatrix} x_C \mp \sin \gamma \\
y_C \pm \cos \gamma \\
z_C
\end{bmatrix}, \quad \lambda \neq 0 \tag{6}
\]

which allows one to find the following relationships:

\[
x_C = \frac{1}{2\lambda} (u_L + u_R) \\
y_C = \frac{1}{2\lambda} (v_L + v_R) \\
z_C = \frac{1}{\lambda}
\]

\[
r = \frac{1}{2\lambda} \sqrt{(u_L' - u_R')^2 + (v_L' - v_R')^2}
\]

\[
\gamma = \arctan \left( \frac{-u_L' - u_R'}{v_L' - v_R'} \right) \quad v_L' \neq v_R'
\]

Thus, simply knowing the eye corners’ location in the image is sufficient to fully determine \(\gamma\) and to determine the remaining fixed parameters up to a constant scaling factor.

## 4. Iris Detection

### Unwrapping the eye

After the eye model is calibrated, the location where each point on the \((\theta, \phi)\) plane lies in the eye image is determined by using the standard perspective camera model and Equation (4) (where \(\lambda \neq 0\)):

\[
\begin{bmatrix}
u(\theta, \phi) \\
v(\theta, \phi) \end{bmatrix} = \lambda K \begin{bmatrix}
R(\theta, \phi, \gamma) & 0 \\
0 & -r
\end{bmatrix} + X_C \tag{8}
\]

Based on this information, the method resamples the eye image \(I(u, v)\) to obtain the pixel values corresponding to the \((\theta, \phi)\) plane thus creating the unwrapped image \(I(\theta, \phi)\) such that

\[
I(\theta, \phi) = I(u(\theta, \phi), v(\theta, \phi)) \tag{9}
\]

![Figure 5](image1)

Figure 5. Top: Original eye images \(I(u, v)\). Bottom: Unwrapped images \(I(\theta, \phi)\) resampled from the original images. Note that, on the unwrapped images, the iris has a circular shape, the eyelids are approximately horizontal and the eyelashes are approximately vertical.

![Figure 6](image2)

Figure 6. Voting procedure for the Hough Transform. Left, the points with gradient above a threshold are the only ones to vote. Right: Each point votes for multiple locations of the center of the iris along the opposite direction of its gradient.

Figure 5 shows a few of the \(I(\theta, \phi)\) images created by the method.

The iris has the shape of a circle on the \((\theta, \phi)\) plane, regardless of the gaze direction. Thus, the method models the iris as a circle with three parameters, the center \((\theta_1, \phi_1)\) and a radius \(r_1\):

\[(\theta, \phi) \in \text{iris} \iff (\theta - \theta_1)^2 + (\phi - \phi_1)^2 \leq r_1^2 \tag{10}\]

In theory, the iris radius \(r_1\) is constant for every frame in the eye video and could be determined during the calibration of the eye model. In practice, however, we found that, because the eyeball is not a perfect sphere and because there can be small errors in the eye calibration calibration, the apparent radius of the iris in the \((\theta, \phi)\) plane is not constant. Therefore the method does not attempt to estimate a fixed value for \(r_1\) and instead allows it to vary within a small range \(r_1 \in R_1\).

### Initial Iris Estimate using Hough Transform

The Hough Transform estimates the model parameters through a voting procedure where each observation votes for the values of the parameters that agree with it. The method uses the Hough Transform to allow each point on the iris circum-
estimate the center \((\theta_1, \phi_1)\) and radius \(r_1\) of the circle for each group. The method then selects the estimate of parameters that produces the most number of inliers. To count the number of inliers for each parameter estimate, the method thresholds the distance of each point to the circle as illustrated on the right side of Figure 7.

Once the method estimates the center of the iris in the \((\theta, \phi)\) plane, it is trivial to use Equation (4) to determine the location of the iris on the eye image, which we denote by \((u_I, v_I)\), as well as the pixels corresponding to the detected circle. These points are highlighted in Figure 8.

5. Gaze Estimation

To estimate the gaze, the method assumes that there is a linear model that relates the iris location on the eye image \((u_I, v_I)\) to the gaze location on the output image, which is denoted by \((u_G, v_G)\). I.e., the method assumes that:

\[
\begin{bmatrix}
    u_G \\
    v_G
\end{bmatrix} = \begin{bmatrix}
    u_I \\
    v_I
\end{bmatrix} + b \Leftrightarrow \begin{bmatrix}
    u_G \\
    v_G
\end{bmatrix} = \begin{bmatrix}
    a_{11} & a_{12} \\
    a_{21} & a_{22}
\end{bmatrix} \begin{bmatrix}
    u_I \\
    v_I
\end{bmatrix} + \begin{bmatrix}
    b_1 \\
    b_2
\end{bmatrix}
\]

Figure 8. Example of detected iris edge points overlaid on the original input images.

Figure 9. Eye (top) and outside (bottom) images obtained for multiple locations of the target screen. These images are used to calibrate the gaze model.

The method starts by computing the gradient of a Gaussian smoothed version of \(I(\theta, \phi)\), which we denote by \(G(\theta, \phi)\). We let \(|G(\theta, \phi)|\) be the magnitude of the gradient and \(\Psi(\theta, \phi)\) be its orientation. Each point that has a gradient above a threshold “votes” for the radii and center along the direction to its gradient (see Figure 6.) i.e., if \(|G(\theta', \phi')| > G_{thr}\) then the point \((\theta', \phi')\) votes for the following \(\theta_I, \phi_I, r_I\) combinations:

\[
\forall r_I' \in R_I : \text{vote for } \begin{cases}
    \theta_I = \theta' - r_I' \cos [\Psi(\theta, \phi)] \\
    \phi_I = \phi' - r_I' \sin [\Psi(\theta, \phi)] \\
    r_I = r_I'
\end{cases}
\]

(11)

After all the points vote, the votes are tallied into bins with each vote also counting towards its neighbors in the \(\theta_I, \phi_I, r_I\) directions. Finally, the initial estimate of the circle parameters corresponds to the bin with the highest number of votes.

Detection of Iris Circumference Features Based on the initial estimate of the circle location, the method detects features in a way inspired by the OpenEyes project. Starting at the estimate of the circle center, it samples \(I(\theta, \phi)\) along paths flowing outwards of the center as illustrated on the left side of Figure 7 (note that the method only considers paths close to the horizontal direction to minimize the interference of the eyelids on the feature detection process.) To detect features, the method searches for the maximum of the gradient along each of these paths.

Final Iris Estimation through Robust Circle Fitting The method uses an algorithm proposed by G. Taubin [8] to estimate the circle that best fits the features detected in the previous step. This method produces accurate circle fits even if data points are observed only within a small arc, but is sensitive to the presence of outliers.

To circumvent this limitation, the method uses the RANSAC algorithm. In particular, the method randomly selects small groups of features and uses Taubin’s method to...
To determine the parameters of the linear relationship, $A$ and $b$, the gaze model is calibrated by presenting the user with a target screen as shown on Figure 9. The user is asked to click on the red circle multiple times in different locations on the screen.

Each time $t \in \{1, \ldots, T\}$ that the user clicks on the red circle, the system determines the iris location on the eye image $(u_t^f, v_t^f)$ as well as the circle location on the outside image $(u_t^c, v_t^c)$. By assuming that the user is looking at the target when it clicks on it, this procedure determines $T$ relationships between the iris and the gaze locations. Based on this relationships, the method uses RANSAC to determine the parameters, $A$ and $b$, of the gaze model.

6. Experimental Results

Datasets Used The performance of the proposed method was evaluated against the Open Eyes method [3] as well as the method proposed by Tsukada et al. [9]. We used two datasets: Dataset 1 corresponds to the dataset used in [9] and consists of 123 frames. Dataset 2 consists of 480 frames divided into train (80 frames) and test (400 frames) sequences. Dataset 2 is more challenging because it contains an greater amount of eye movement covering a broader range of gaze orientations. Ground truth for both datasets was labeled by a human operator that hand-fitted an ellipse (5 parameters) to the iris contour for each frame. None of the datasets contains images where the eye is blinking (closed or nearly closed).

Evaluation of Results To measure the quality of the iris detection, Tsukada et al. [9] use an area score based on the overlap between the detected and the ground truth ellipse. Let $E$ denote the detected ellipse, and $E_{\text{ground}}$ denote the ground truth ellipse. The area score $S_{\text{area}}$ is given by:

$$S_{\text{area}} = 1 - \frac{\text{Area}(E \cap E_{\text{ground}})}{\text{Area}(E \cup E_{\text{ground}})} \quad (13)$$

Most of the gaze tracking algorithms (including OpenEyes, Tsukada’s and the proposed method) do not use the ellipse parameters to estimate the gaze. Instead, the gaze is computed solely based on the location of the center of the ellipse. Thus, the overlap area score has a limited usefulness when evaluating the quality of iris detection.

To address this limitation, we propose a distance score based on the euclidean distance between the center of the detected ellipse $E_C$ and the center of the ground truth ellipse $E_{\text{ground}}$ normalized by the average of the major and minor axes of the ground truth ellipse (denoted by $E_{\text{ground}}^a$ and $E_{\text{ground}}^b$, respectively):

$$S_{\text{dist}} = \frac{|E_C - E_{\text{ground}}|}{\frac{1}{2} (E_{\text{ground}}^a + E_{\text{ground}}^b)} \quad (14)$$

For both score types, a lower value corresponds to better iris detection. Figure 10 illustrates the difference between the two scores. In particular, note that it is possible to have very poor area score, while having the ellipse center at the best location and consequently having a very good distance score (which reflects the fact that this detection would result in accurate gaze estimation.) On the other hand, it is not possible to have a poor distance score while having a good area score since, once the detected ellipse center is far away from the ground truth ellipse, the overlap between the ellipses will always be poor. This fact is illustrated by Figure 11, which shows the best area score attainable for multiple values of the distance score.

Successful vs Unsuccessful Iris Detection When the iris detection task is challenging (such as when the eyes are blinking or when the iris is close to the eye corners,) some of the methods fail to produce an estimate of the iris location, or output a location estimate that is so far away from the ground truth that the gaze estimate would be clearly incorrect. To illustrate this, we introduce the concept of successful versus unsuccessful iris detection. For the area score, we define a successful detection as a frame where the score is above 0.5 (the area of the detected ellipse overlaps at least

![Figure 10](image)

![Figure 11](image)
Table 1. Results of the iris detection on Dataset 1 (lower scores are better). The proposed method is more accurate than both OpenEyes and Tsukada’s methods.

Results for Dataset 1  All methods where able to successfully estimate the iris location for all frames in Dataset 1. Table 1 summarizes the performance of the evaluated methods. The table shows that, regardless of the score function used, the proposed method is more accurate (lower average error) than both the OpenEyes and Tsukada’s method.

Results on Dataset 2  Figures 12 and 13 summarize the results for Dataset 2 (for the area and distance scores, respectively). The results show that the proposed method outperforms Tsukada’s since it is able to detect the iris successfully in drastically more frames and has, in general, better performance on the frames where the iris is detected successfully. (The OpenEyes method was excluded from the comparison since it failed to produce an output for the vast majority of the frames in the dataset.) Figure 14 shows some examples of successful and failed detection.

Applications and Comparison to Commercial Products  The method runs at 15 frames per second on a modern computer and was applied to multiple indoors and outdoors scenarios, including office scenes, driving in natural conditions and driving on a NASCAR race track. Additionally, the method was coupled with a face detection and recognition system and text-to-speech technology that enabled it to say out loud the name of the person that the user was looking at. Figure 15 illustrates some of these applications. Note that it would not be possible to use an active-illumination method for gaze tracking in many of the applications tested.

Tsukada et al. compared their method to a commercial product and found it to be competitive to an active-illumination solution [9]. Although we did not perform a formal comparison with a commercial system, we believe our method is also competitive with such systems since it typically outperforms Tsukada’s.

7. Conclusion  This paper presented a new passive-illumination gaze tracking method based on the use of a 3D eye model (determined during calibration) to warp the spherical surface of the eye in the input image into a plane and thus “unwrap the eye”. After the eye is “unwrapped”, the location of the iris is determined by using a robust circle-fitting algorithm (instead of ellipse-fitting on the eye image.) The proposed approach reduces the search space of the problem from 5 to 3 dimensions without diminishing the richness of the underlying model and results in a fast, robust and accurate method. Extensive experimentation in real-world applications as well as comparison with current state of the art passive-illumination alternatives, demonstrate the superiority of the proposed method.

References

Train Dataset

Test Dataset

Figure 13. Distance Score $S_{area}$ for Dataset 2. Left: Distribution of scores for successful detection. Right: Count of frames with unsuccessful detection. The proposed method outperforms Tsukada’s. It is able to detect the iris successfully in drastically more frames and has better performance on the successful test frames.


Figure 14. Examples of method outputs for Dataset 2. First row: Accurate detection by both methods; Second row: Inaccurate detection (left) and unsuccessful detection (right) by both methods; Third row: Tsukada’s method fails, whereas proposed method succeeds.

Figure 15. Examples of applications. First row: Indoors scenes - tracking a ball thrown against a wall and held by another person. Second row: Outdoors driving scenes. Third row: Driving scenes in a NASCAR racing track. Forth row: Additional applications - Face detection and recognition and simultaneous two eye tracking.