2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops

Visible-Spectrum Gaze Tracking for Sports

Bernardo R. Pires, Myung Hwangbo, Michael Devyver, Takeo Kanade
Carnegie Mellon University
Pittsburgh, PA
{bpires,myung,devyver,tk}@cs.cmu.edu

Abstract

*In sports, wearable gaze tracking devices can enrich the viewer experience and be a powerful training tool. Because devices can be used for long periods of time, often outside, it is desirable that they do not use active illumination (infra-red light sources) for safety reasons and to minimize the interference of the sun. Unlike traditional wearable devices, in sports, the gaze tracking method must be robust to (often dramatic) movements of the user in relation to the device (*i.e. *the common assumption that, because the device is wearable, the eye does not move with regards to the camera no longer holds.) This paper extends a visible-spectrum gaze tracker in the literature to handle the requirements of a motor-sports application. Specifically, the method presented removes the original assumption (in the original method) that the eye position is fixed, and proposes the use of template matching to allow for changes in the eye location from frame to frame. Experimental results demonstrate that the proposed method can handle severe changes in the eye location and is very fast to compute (up to 60 frames per second in modern hardware.)*

1. Introduction

The question "where was he looking at?" has been asked countless times by both coaches and fans. Analysis of the gaze direction of athletes can give an insight into their perception of the sport and the motivations of their actions (*e.g.*, whenever a soccer or football player passes to the wrong teammate, was it because of poor judgment or because he did not "see" the correct teammate?)

Recent video-based eye tracking (video-oculography) systems can be divided into appearance-based and shape-based methods. Appearance-based methods map the entire eye image directly to gaze position [10], and are generally less accurate when compared with shape-based methods, which track specific portions of the eye anatomy such as the corneal reflection [7], pupil contour [8], and iris contour [4]. (See [3] for a recent survey of the eye-tracking literature.)

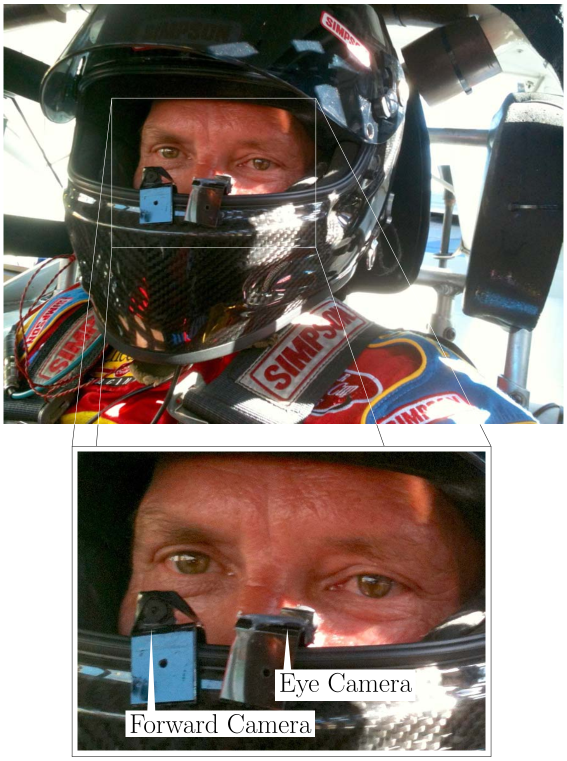

Figure 1. Early prototype for initial data collection in a NASCAR race circuit. This prototype consists of a forward facing camera to observe the environment from the driver's point of view and an eye facing camera for estimation of the gaze direction. Both cameras are connected to a laptop computer for data recording.

Among the more accurate shape-based methods, corneal reflection and pupil contour normally require infrared ray (IR) active illumination. These methods can be highly accurate, but are not adequate for extended daily use [6] and are susceptible to interference from other IR sources, namely the sun. Non-active illumination methods usually focus on the detection of the iris contour. Unlike IR, visible-spectrum gaze tracking is more challenging due to reflections and specularities introduced by arbitrary illumination,

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.146

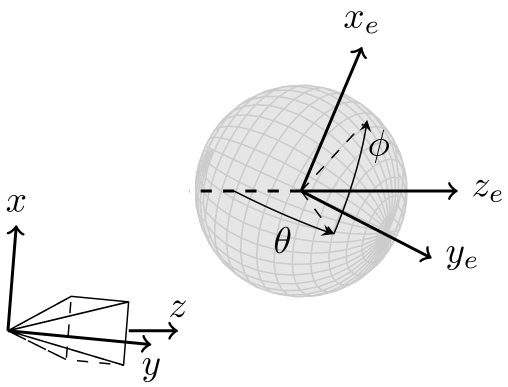

Figure 2. Eye model. The eye sphere is represented in gray; the camera coordinates are (x, y, z); the eye coordinates are (x_e, y_e, z_e); and the eye surface is parameterized by (θ, ϕ).

variability in the iris color between users, and variability in the eye texture due to changes in eye irrigation. Thus it is often more computationally complex and requires use of advanced computer vision methods.

One of the earliest approaches [5] to this problem used RANSAC to fit an ellipse to the iris but was not robust to changes in illumination and occlusions. The methods in [12] and [11] also use ellipse fitting to estimate the iris but use three-dimensional (3D) models to improve robustness. However, they either require the estimation of a large number of parameters per frame [12] or require complex calibration procedures [11].

More recently, the "Unwrapping the Eye" method proposed in [9] uses a 3D model to represent the eye and uses that model to estimate the planar appearance of the eye surface (thus "unwrapping" the eye.) Because the iris is a circle on the eye surface, iris detection can be obtained by a circle fitting procedure instead of the more computationally demanding ellipse fitting used by other methods. This method was demonstrated to out-perform other visible-light gaze tracking methods and to be competitive with commercial IR based solutions [9].

Despite their ability to track the gaze of the user, none of the visible light gaze tracking methods in the literature is suitable to applications in sports because they do not allow for significant movements in the eye location in relation to the camera. I.e., they degrade very rapidly if the user's head moves in relation to the wearable device. This paper proposes an extension to the method in [9] so that the eye location is tracked from frame to frame, thus dramatically increasing the gaze tracking robustness in practical sports applications.

Organization of the Paper Section 2 gives a brief overview of the "unwrapping" the eye method that serves as the basis of the proposed method. Section 3 describes the proposed method as an extension of [9] that uses template matching to determine the eye corners and thus is robust movement of the user in sports applications. This section also details the choice of optimal template size. Section 4 details the comparison of the proposed method against the state of the art. Finally, Section 5 concludes the paper.

2. Overview of the Method in [9]

The "Unwrapping the Eye" method models the eyeball as a three-dimensional sphere and the iris as a two-dimensional circle on the sphere. Iris detection is achieved by determining the center of the iris on the surface of the eye. Since this method is used as the basis to the paper's contribution, we briefly review the "Unwrapping the Eye" method here for completeness.

Eye Model Following the assumptions of [9], (x, y, z) is the camera coordinate system and the eye coordinate system (x_e, y_e, z_e) is defined so that its origin coincides with the eye center $X_c = [x_c\, y_c\, z_c]^{\mathrm{T}}$; the axis z and z_e are aligned; and the angle between x and x_e is γ (see Figure 2). If r is the radius of the eye sphere, the surface of the eye can be parameterized by (θ, ϕ) such that (see [9] for detailed derivation:)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}_z(-\gamma)\,\mathbf{R}_y(\phi)\,\mathbf{R}_x(\theta) \begin{bmatrix} 0 \\ 0 \\ -r \end{bmatrix} + X_c \qquad (1)$$

where the rotation matrices are defined by:

$$\mathbf{R}_z(-\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

$$\mathbf{R}_y(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \qquad (3)$$

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \qquad (4)$$

Calibration Equation 1 shows that the eye model is fully parameterized by the γ angle, the eye radius r, and the eye center X_c. If the eye is fixed with regards to the camera, these parameters remain constant and can be determined during calibration of the system. This is achieved by determining, on the eye image $\mathcal{I}(u, v)$, the corners of the eye $\left(v_l^0, u_l^0\right)$ and $\left(v_r^0, u_r^0\right)$. Assuming that the corners are aligned with the y_e axis of the 3D eye model, they correspond to $(\theta, \phi) = (\pm\pi/2, 0)$. The authors in [9] show that, assuming that the camera parameters are known, this information is sufficient to determine the γ parameter precisely and the X_c and r model parameters up to a scale factor.

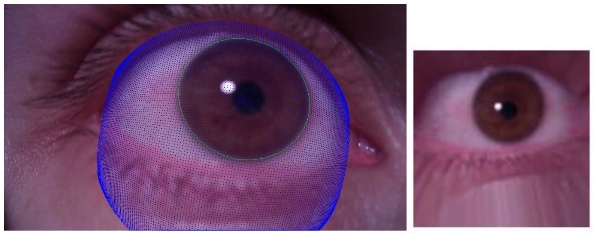

Figure 3. Left: original eye images $\mathcal{I}(u,v)$ with the 3D model superimposed. Right: "unwrapped" image $I(\theta,\phi)$, which is a planar representation of the eye surface.

Eye Surface Image At run-time, the method creates, for every image from the eye camera $\mathcal{I}(u,v)$, a new image $I(\theta,\phi)$ such that the pixels in the 3D eye surface are warped to a plane. The authors in [9] refer to this new image as the "unwrapped" image since the spherical eye surfaced is unwrapped into a plane.

Using equation (1), the 3D location (x,y,z) of the eye surface for each value of (θ,ϕ) can be determined. Assuming the intrinsics of the eye camera are known, the standard perspective projection model is used to determine the projection of every (x,y,z) point into the image plane (u,v). Thus, one can write u and v as functions of θ and ϕ and obtain the planar representation of the eye surface $I(\theta,\phi)$ - the "unwrapped" image - by re-sampling the eye image $\mathcal{I}(u,v)$:

$$I(\theta,\phi)=\mathcal{I}(u(\theta,\phi),v(\theta,\phi)) \qquad (5)$$

See Figure 3 for an example of both the eye image and the "unwrapped" eye surface image.

Iris Detection The iris detection algorithm uses exclusively the "unwrapped" eye surface image and takes advantage of the fact that the iris is circular in this image. Thus, the method can replace the traditional ellipse fitting with a simpler, faster, and more robust circle fitting procedure.

Because the iris is circular, the method starts by using the circular Hough transform to coarsely estimate the center of the iris. Based on this information, the method searches for features by determining the maximum of the image gradient in radial paths from the iris center (the maximum of the gradient corresponds to the edges of the iris.) Finally, a robust circle fitting procedure is used to determine the exact iris location. See [9] for details.

3. Proposed Method

The method in [9] works well for stable applications, such as when the user is sitting and gazing at a screen, but it is not able to cope with situations where the user's activities may cause the camera to move with relationship to the head (and consequently) the eye of the wearer. See Figure 4 for an example. The method presented here expands on [9] to remove this limitation. In particular the eye corners (or other distinctive features visible in the camera feed and annotated by the operator) are tracked and the 3D eye model is adjusted from frame to frame. This allows the method to obtain a consistent "unwrapped" eye surface image and thus achieve high quality iris detection.

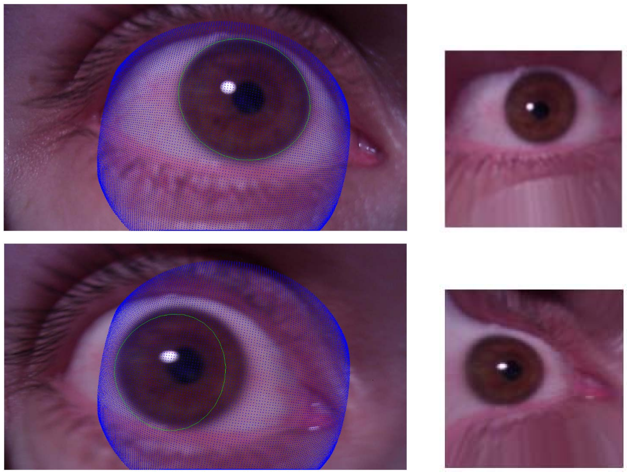

Figure 4. On the original method, the eye model does not change after calibration, which causes errors if the eye position changes with regards to the wearable device. Top: Initial image used for calibration and corresponding "unwrapped" eye surface image. Bottom: The user's eye moved in the image which causes the 3D model to sample the eye surface image incorrectly (note the distortion on the "unwrapped" eye surface image on the right.)

3.1. Movement of the Camera

Although the eye location, when observed from the eye camera, can change significantly, the distortions are limited by the geometry of the wearable device used. In particular, for cameras mounted on helmets or eyeglasses, the motion corresponds to a rotation centered on the head of the user (see Figure 5.) Because the distance from the eye to the camera remains approximately constant and because the radius of the camera rotation is much larger than the distance to the eye, the movement distortions in the eye image can be accurately modeled by a translation.

Mathematically, the argument above states that, on the camera reference coordinates (x,y,z), the movement of the eye center X_c with regards to the eye camera can be modeled by:

$$X_c=\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}=\begin{bmatrix} x_c^0+x_c(t) \\ y_c^0+y_c(t) \\ z_c^0 \end{bmatrix} \qquad (6)$$

where $X_c=\left[x_c^0\ y_c^0\ z_c^0\right]^{\mathrm{T}}$ corresponds to the original eye

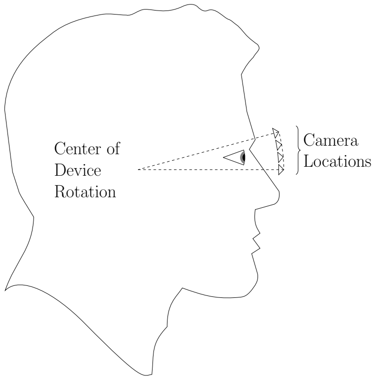

Figure 5. Although the eye can move significantly with regards to the camera (see Figure 4,) the type of distortions is constrained by the geometry of the wearable device.

center location determined during calibration and the movement is modeled by the terms $x_c(t)$ and $y_c(t)$.

Using equation 1, and assuming an upper triangular intrinsic camera matrix, equation 6 implies that locations of the corners of the eye can be modeled by:

$$\left\{\begin{array}{l} u_l = u_l^0 + u(t) \\ u_r = u_r^0 + u(t) \end{array}\right. \qquad \left\{\begin{array}{l} v_l = v_l^0 + v(t) \\ v_r = v_r^0 + v(t) \end{array}\right. \tag{7}$$

which shows that the camera movement is modeled by a translation in the eye image with time varying parameters $u(t)$ and $v(t)$.

3.2. Eye Corner Detection

During calibration, in addition to highlighting the user's eye corners, the operator selects the range of expected translations to be observed due to the movements of the user's head in relation to the wearable device. The corner information is used to select a region of the image as a template for each eye corner $\mathcal{T}_l\left(u,v\right)$ and $\mathcal{T}_r\left(u,v\right)$. At run-time, template matching [2] is used to determine the correct location of the eye corners.

Template matching works by comparing all segments in the region of interest selected by the operator to the corresponding eye corner template, and determining the best match. For the left corner, we have:

$$\left(\hat{u}_l, \hat{v}_l\right) = \arg\max_{\left(u_l, v_l\right)} S\left(\mathcal{I}\left(u+u_l, v+v_l\right), \mathcal{T}_l\left(u,v\right)\right) \tag{8}$$

where the optimization is done for every eye image but the time index t was omitted for compactness, and S is a similarity measure between the two image segments. The result for the right corner is equivalent.

To determine the best measure of template similarity $S\left(\mathcal{I}\left(u+u_l, v+v_l\right), \mathcal{T}_l\left(u,v\right)\right)$ we compare four different measures (all summations are done over the range of (u,v) values where the template was sampled:)

- Squared Difference (sqdiff):

$$S\left(\bullet\right) = -\sum_{u,v}\left[\mathcal{I}\left(u+u_l, v+v_l\right) - \mathcal{T}_l\left(u,v\right)\right]^2 \tag{9}$$

- Normed Squared Difference (sqdiffNormed):

$$S\left(\bullet\right) = -\frac{\sum_{u,v}\left[\mathcal{I}\left(u+u_l, v+v_l\right) - \mathcal{T}_l\left(u,v\right)\right]^2}{\sqrt{\sum_{u,v}\mathcal{I}^2\left(u+u_l, v+v_l\right)\cdot\sum_{u,v}\mathcal{T}_l^2\left(u,v\right)}} \tag{10}$$

- Normed Cross Correlation (ccorrNormed):

$$S\left(\bullet\right) = \frac{\sum_{u,v}\left[\mathcal{I}\left(u+u_l, v+v_l\right)\cdot\mathcal{T}_l\left(u,v\right)\right]}{\sqrt{\sum_{u,v}\mathcal{I}^2\left(u+u_l, v+v_l\right)\cdot\sum_{u,v}\mathcal{T}_l^2\left(u,v\right)}} \tag{11}$$

- Normed Pearson Correlation (ccoefNormed): where the similarity is as in equation (11) but the image $\mathcal{I}$ and template $\mathcal{T}$ are replaced by their mean subtracted counterparts $\mathcal{I}' = \mathcal{I} - \bar{\mathcal{I}}$ and $\mathcal{T}' = \mathcal{T} - \bar{\mathcal{T}}$ (see [1] for more details.)

The comparison of similarity measures was performed on a dataset of 250 frames extracted from a video recorded with a helmet equipped with visible-spectrum miniature cameras. The helmet is moved with relationship to the user's eye to simulate natural conditions during the practice of the sport. The template is extracted from the initial frame manually, and the similarity measures are compared with the ground truth for all other frames.

Figure 6 presents the percentage of correct eye corner detections (number of times that the detection is within 5% of the eye radius from the ground truth divided by the total number of analyzed frames) and the time to compute the best match. In both cases results are presented as a function of the template size (as a percentage of the eye radius.)

The best similarity measure and template size strike a balance between fast computation times and large percentage of correct detections. From Figure 6, we determine that the Normed Cross Correlation measure with a template size of approximately 4.5% of the eye radius achieves the best trade-off.

4. Experimental Results

The complete iris detection method was implemented in C++ and uses OpenCV [1] for some of the basic vision tasks. In practical applications the template matching task runs in parallel with the iris detection so as to not reduce the frame rate (note that, from Figure 6, we know that the template matching tasks takes less than $200ms$, *i.e.* runs at approximately 5 frames per second.) After each template matching operation the eye model is recomputed and the iris detection procedure is updated appropriately.

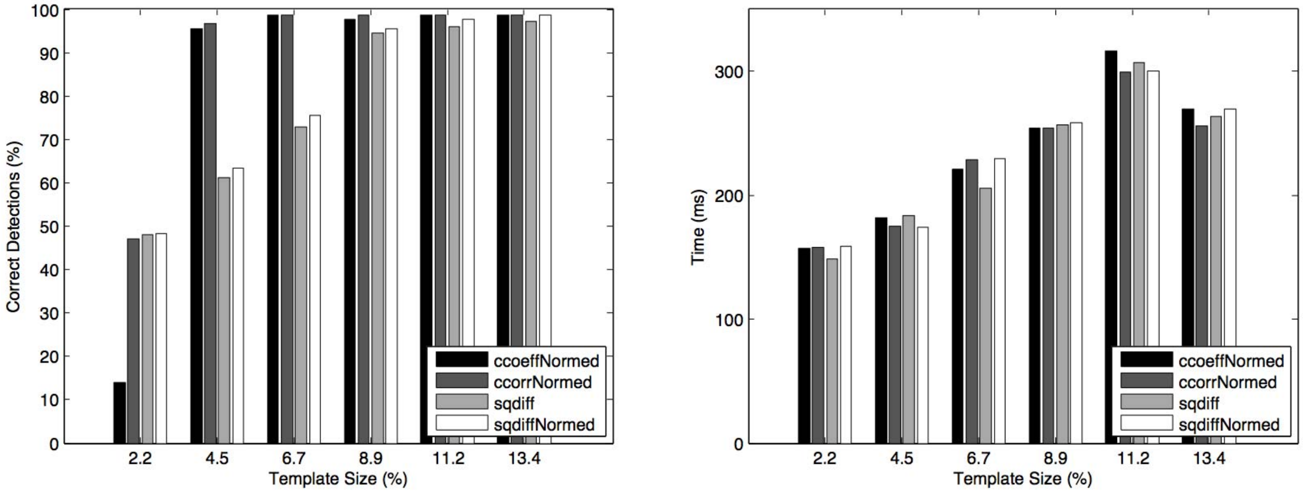

Figure 6. Comparison of similarity measures. Left: Percentage of correct eye corner detections (defined as a detections where the distance from the ground truth is less than 5% of the eye radius) as a function of the template size (as a percentage of the eye radius.) Right: Time to compute the template detection also as a function of the template size.

In order to evaluate the performance of the proposed method, we compared it against the original method in [9] on a different set of 250 frames (again, taken with a helmet wearable device, simulating the head movements with regards to the helmet.) To fully explore the properties of the proposed method, the corners were detected on every frame at the expense of slower frame rate. However, the performance was very similar to the real-time system. See Figure 7 for a summary of the results.

We first note that, to produce better results, it is often useful to use the last few frames of history to "stabilize" the iris detection result (*e.g.*, by using an exponentially decaying average filter.) Under normal circumstances, this improves the quality of the detection at the expense of a small lag when there is large eye movement. During testing we have discovered that omitting this stabilizing operation improves results for the original method [9]. This can be explained by the fact that, since this method assumes the eye is fixed, the movements of the eye ball with regards to the camera are actually interpreted as very rapid movements of the iris on a fixed eye ball. As would be expected, since the proposed method tracks the eye ball, it observes fairly constrained iris movement and omitting the stabilization step on this case reduces (slightly) the performance.

Figure 7 shows that, even in the best case, the proposed method always outperforms the original method in [9]. It is important to note, however, that the proposed method still has a failure case when the distortion is so large that the corners are not detected. These cases however, can be mitigated by modeling the corners' movement over time or using more advanced matching method.

Finally, Figure 8 illustrates the advantages of the proposed method by showing its output for the images in Figure 4.

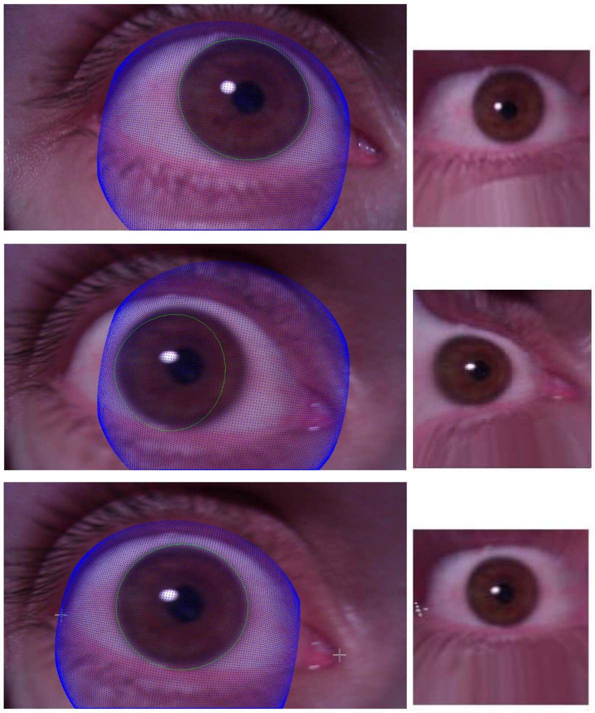

Figure 8. The proposed method can accurately track the user's eyeball and is robust to movements of the wearer's head with relation to the eye camera.

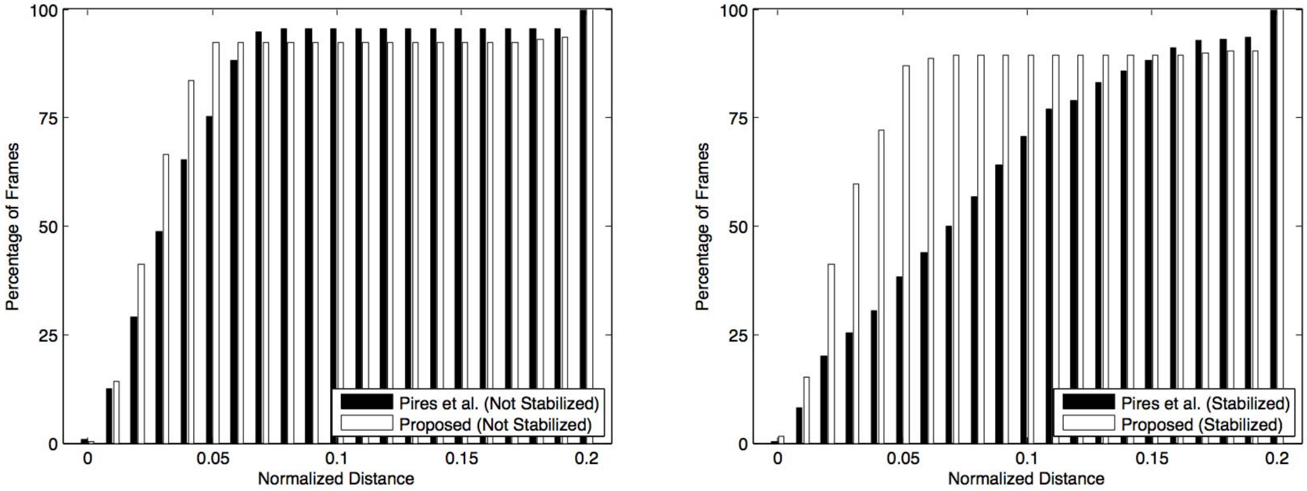

Figure 7. Percentage of frames where the distance from the detected eye center to the ground truth is below a given normalized distance (*i.e*. distance divided by the eye radius.) Left: Previous history is not used to stabilize the method results. Right: Previous history is used to stablize results.

5. Conclusion

There are many potential applications of gaze tracking technology in sports. But, in order for the technology to be widely used, it is necessary to solve a set of problems specific to the sports industry. Non-active illumination based gaze tracking that copes with movement of the user with regards to the wearable device is desirable for field applications. This paper proposes such a method by extending of a visible-spectrum gaze tracker in the literature to cope with the necessary motion distortions. In particular, the proposed method uses template matching to track the corners of the eyes of the user and incorporates this information into the iris detection procedure. The resulting method is fast to compute and can handle large changes in the eye location.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4

[2] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. John Wiley & Sons, 2009. 4

[3] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3):478–500, Mar. 2010. 1

[4] D. W. Hansen and A. E. C. Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181, Apr. 2005. 1

[5] D. Li and D. J. Parkhurst. Open-source software for real-time visible-spectrum eye tracking. In *Proceedings of the COGAIN Conference*, pages 18 – 20, 2006. 2

[6] B. Noris, J.-B. Keller, and A. Billard. A wearable gaze tracking system for children in unconstrained environments. *Computer Vision and Image Understanding*, 115(4):476–486, Apr. 2011. 1

[7] T. Ohno and N. Mukawa. A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, ETRA '04, pages 115–122, New York, NY, USA, 2004. ACM. 1

[8] A. Pérez, M. L. Córdoba, A. García, R. Méndez, M. L. Muñoz, J. L. Pedraza, and F. Sánchez. A precise eye-gaze detection and tracking system. In *WSCG'03*, 2003. 1

[9] B. R. Pires, M. Devyver, A. Tsukada, and T. Kanade. Unwrapping the eye for visible-spectrum gaze tracking on wearable devices. In *IEEE Workshop on the Applications of Computer Vision WACV'13*, pages 369–376, Clearwater Beach, FL, USA, January 2013. 2, 3, 5

[10] K.-H. Tan, D. J. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Proceedings of the 6th IEEE Workshop on Applications of Computer Vision*, WACV '02, page 191, Washington, DC, USA, 2002. 1

[11] A. Tsukada, M. Shino, M. Devyver, and T. Kanade. Illumination-free gaze estimation method for first-person vision wearable device. In *2nd IEEE Workshop on Computer Vision in Vehicle Technology: From Earth to Mars (ICCV Workshops)*, pages 2084 – 2091, November 2011. 2

[12] H. Wu, Y. Kitagawa, T. Wada, T. Kato, and Q. Chen. Tracking iris contour with a 3d eye-model for gaze estimation. In *Proceedings of the 8th Asian conference on Computer vision - Volume Part I*, ACCV'07, pages 688–697, Berlin, Heidelberg, 2007. Springer-Verlag. 2