Front Camera Eye Tracking for Mobile VR

Panagiotis Drakopoulos*
Technical University of Crete

George Alex Koulieris†
Durham University

Katerina Mania‡
Technical University of Crete

ABSTRACT

User fixations is a fast and natural input method for VR interaction. Previous attempts for mobile eye tracking in VR were limited due to low accuracy, long processing time and the need for hardware add-ons such as anti-reflective lens coating and IR emitters. We present an innovative mobile VR eye tracking methodology, utilizing only the captured images of the front-facing (selfie) camera through the headset's lens, without any modifications. The system enhances the low-quality camera-captured images that suffer from low contrast and poor lighting by applying a pipeline of customized low level image enhancements to suppress obtrusive reflections due to the headset lenses. We proceed to calibration and linear gaze mapping between the estimated iris centroids and physical pixels on the screen resulting to iris tracking in real-time. A preliminary study confirms that the presented eye tracking methodology performs comparably to eye trackers in commercial VR headsets when the eyes move in the central part of the headset's field of view.

Keywords: Eye-tracking, Mobile VR, Gaze-contingent Displays

1 INTRODUCTION

Input methods for interaction in mobile VR, i.e. a mobile phone placed in an inexpensive cardboard or plastic case (Figure 1), are limited to uncomfortable head-tracking controlling a pointer on the screen or sometimes, using a side button of the case. Prior work has demonstrated that eye fixations is a natural input method for interaction with head-worn displays [6]. In this poster, we propose an innovative mobile VR eye tracking methodology, utilizing only the captured images of the front-facing (selfie) camera without any modifications or additional hardware. Early study data show that this methodology results to accuracy which is comparable to eye trackers in commercial VR headsets in the central Field-of-View (FoV) of about 20°of visual angle. Our system tracks the left eye based on the smartphone's camera input.

State of the art eye tracking relies on the enhanced pupil contrast provided by the infrared (IR) light of specialized eye tracking devices. When IR is absent, screen content can either produce obtrusive reflections on the headset's lens, or eye detection could fail when the iris is barely visible.

A preliminary attempt for smartphone-based eye tracking in mobile VR relied on either coating the headset lenses with an anti-reflective layer [4] or comparing the reflections of the on-screen content on the surface of the eye, known as Purkinje images, to calibrated images to infer an estimated gaze position [3]; the latter did not work in real-time. More recently, coarse gaze tracking on a smartphone-based VR headset was presented by using a Convolutional Neural Network (CNN) resulting to relatively low accuracy of nearly 10° with calibration, which is prohibitive for real-time gaze-based interaction [1].

*e-mail: panosdrak@hotmail.com
†e-mail: georgios.a.koulieris@durham.ac.uk
‡e-mail: k.mania@ced.tuc.gr

Figure 1: Unmodified HMD with smartphone.

Our eye tracking methodology for mobile VR enhances the low-quality, poor lit camera-captured images by combining a pipeline of low level image enhancements to suppress reflections due to the headset lenses. We devise an efficient circle fitting geometric algorithm to obtain an accurate estimation of the iris center based on customizing a Hough circle transform to be highly sensitive to circular features in the image such as the iris. Purpose-specific calibration and gaze mapping algorithms are designed and implemented to convert the detected eye center co-ordinates to screen coordinates resulting in real-time eye tracking.

2 BACKGROUND

In mobile headsets, eye gaze can be either tracked [8] or predicted [7]. Eye tracking techniques can be classified as **feature-based** and **model-based** [8]. Feature-based detect a key feature, e.g. the dark pupil, by relying on pixel intensity levels or intensity gradients. Based on the iris-appearance hypothesis for dark-pupil eye tracking, the pupil is the darkest element to detect. Model-based approaches find the best fitting circle or ellipse of the pupil at a higher computational cost than feature-based, however, they are prohibitive for real-time eye tracking.

Eye-tracking on the infrared is based on a trade-off between runtime performance and accuracy by combining model and feature-based approaches so that stable iris detection is achieved. Our VR mobile eye tracking system deploys a purpose-specific feature-based algorithm to determine the approximate position of the eye and a computationally efficient model-based part in the form of a circle fitting algorithm to obtain an accurate estimation of the iris center.

Calculating the user's point of gaze requires a conversion of the eye-tracker's estimated iris center to screen locations accomplished by a mapping function, determined per user through calibration, i.e. fixation on a set of presented target points of known positions. Our initial lab tests corroborate previous work on mapping functions [2], [5] as high order polynomials did not improve accuracy, nor did increasing the amount of calibration points. Driven by our requirement for high accuracy and low latency, we employ the computationally more efficient linear mapping function to infer the gaze point.

3 System Overview

We now describe the main system components of our mobile VR eye tracking system in order to address the challenges analyzed above. An Apple iPhone 6S was employed as the eye-tracking device. Its 4.7" display has a resolution of 1334x750, 326ppi. The device comes with a 5MP front camera with 720p@30fps video capture capability, along with a 1.84GHz dual-core CPU and 2GB RAM. We placed the mobile phone in a commodity head mounted case for mobile VR with head straps, without IR lighting or any modification.

3.1 Improving iris visibility

Initially, images are captured by the smart phone's camera of head-worn mobile VR at 20 fps. Directly applying circle fitting does not accurately detect the iris center, as the images suffer from poor contrast and obtrusive reflections. We apply a series of image enhancement operations that result in a more salient iris (Figure 2).

3.1.1 Reflection suppression

Salient highlights can be detected as an abrupt peak in high intensity values in the image's histogram. We suppress the intensity values of this region by averaging with nearby pixels that are not considered to be part of the reflection.

3.1.2 Histogram equalization

Eye images suffer from poor contrast, due to low light. Their intensities are cramped in a small range of values. Histogram equalization allow areas of low contrast to gain a higher contrast, by spreading out the most frequent intensity values, resulting in a more even distribution.

3.2 Isolating the eye region

Front camera images (Figure 1b) are contaminated with elements such as the HMD frame, the lenses etc. Our goal is to detect the eye region and the iris within this region by applying circle fitting.

3.2.1 Iris contour

Captured images vary in brightness, reflections and iris saliency. We configured the Hough gradient function, which is a cost-efficient version of the Hough transform, to be highly sensitive to circular features in the image to mitigate function failures. The operation results in $n > 1$ candidate circles with false positives. In order to determine the best match of the contour of the iris, we compute a confidence value for each circle, as a metric to describe how likely it is that it represents the iris.

3.2.2 Iris Region of Interest (RoI)

Having located the iris, we isolate the RoI in which the iris moves. We do this by presenting targets at the extreme ends of the screen and instruct a viewer to fixate on them. This process ensures that the iris survives cropping in all possible eye positions. We find the centroids of the fitted circles and determine the RoI's position as the average of these centroids, so that the RoI encloses all possible iris positions. This operation significantly increases the iris detection speed, as the system searches for the iris on only a small part of the original image. The RoI is calculated only once per user before calibration.

3.2.3 Eye tracking

Having isolated the RoI, we perform image enhancements to improve the visibility of the iris and reapply our circle fitting technique to determine the iris center in the final cropped image. The centroid of the isolated iris is used for calibration, remapping and eye tracking. Once the RoI is determined, the system proceeds to calibration,

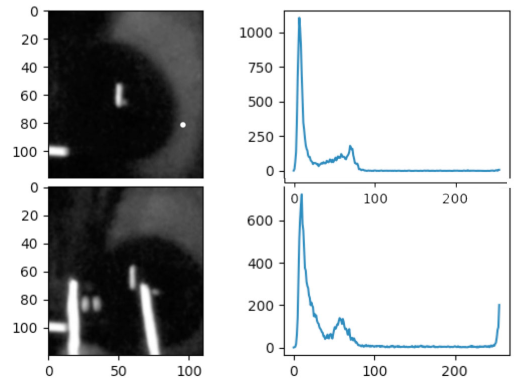

Figure 2: Image histogram with: no lens reflections (top), strong lens reflections (bottom) with a visible peak of high intensity values.

during which users are required to fixate on a set of target points. We apply a linear remapping between the estimated iris centroids and physical pixels on the screen to infer the final gaze point on the screen.

4 Evaluation and Conclusion

We propose iris detection and pupil tracking for eye-tracked head-worn mobile VR, based on the front-facing camera and processing of current smart phones with no modifications. Accuracy and precision of our front camera based mobile eye tracker was examined. Early results indicate that our eye tracker performs best and comparably to eye trackers in commercial VR headsets when the eyes move in the central part of the headset's FoV (about 20°of visual angle). Mean accuracy and precision decreases with eccentricity.

References

[1] K. Ahuja, R. Islam, V. Parashar, K. Dey, C. Harrison, and M. Goel. Eyespyvr: Interactive eye sensing using off-the-shelf, smartphone-based vr headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):57, 2018.

[2] J. J. Cerrolaza, A. Villanueva, and R. Cabeza. Study of polynomial mapping functions in video-oculography eye trackers. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 19(2):10, 2012.

[3] S. W. Greenwald, L. Loreti, M. Funk, R. Zilberman, and P. Maes. Eye gaze tracking with google cardboard using purkinje images. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pp. 19–22. ACM, 2016.

[4] H. Hakoda, W. Yamada, and H. Manabe. Eye tracking using built-in camera for smartphone-based hmd. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pp. 15–16. ACM, New York, NY, USA, 2017. doi: 10.1145/3131785.3131809

[5] P. Kasprowski, K. Hareżlak, and M. Stasch. Guidelines for the eye tracker calibration using points of regard. In *Information Technologies in Biomedicine, Volume 4*, pp. 225–236. Springer, 2014.

[6] G. A. Koulieris, K. Akşit, M. Stengel, R. Mantiuk, K. Mania, and C. Richardt. Near-eye display and tracking technologies for virtual and augmented reality. In *Computer Graphics Forum*, vol. 38, pp. 493–519. Wiley Online Library, 2019.

[7] G. A. Koulieris, G. Drettakis, D. Cunningham, and K. Mania. C-lod: Context-aware material level-of-detail applied to mobile graphics. In *Computer Graphics Forum*, vol. 33, pp. 41–49. Wiley Online Library, 2014.

[8] D. Li, D. Winfield, and D. J. Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pp. 79–79, 2005.