EyeSpyVR: Interactive Eye Sensing Using Off-the-Shelf, Smartphone-Based VR Headsets

KARAN AHUJA, Carnegie Mellon University, USA
RAHUL ISLAM, Indian Institute of Information Technology Guwahati, India
VARUN PARASHAR, Manipal University, India
KUNTAL DEY, IBM Research, India
CHRIS HARRISON and MAYANK GOEL, Carnegie Mellon University, USA

Low cost virtual reality (VR) headsets powered by smartphones are becoming ubiquitous. Their unique position on the user's face opens interesting opportunities for interactive sensing. In this paper, we describe *EyeSpyVR*, a software-only eye sensing approach for smartphone-based VR, which uses a phone's front facing camera as a sensor and its display as a passive illuminator. Our proof-of-concept system, using a commodity Apple iPhone, enables four sensing modalities: detecting when the VR head set is worn, detecting blinks, recognizing the wearer's identity, and coarse gaze tracking - features typically found in high-end or specialty VR headsets. We demonstrate the utility and accuracy of EyeSpyVR in a series of studies with 70 participants, finding a worn detection of 100%, blink detection rate of 95.3%, family user identification accuracy of 81.4%, and mean gaze tracking error of 10.8° when calibrated to the wearer (12.9° without calibration). These sensing abilities can be used by developers to enable new interactive features and more immersive VR experiences on existing, off-the-shelf hardware.

CCS Concepts: • **Human-centered computing → Virtual reality**; *Interaction techniques*;

Additional Key Words and Phrases: VR, periocular biometrics, user identification, eye tracking, gaze tracking, blink detection, personalized service delivery on VR

ACM Reference Format:
Karan Ahuja, Rahul Islam, Varun Parashar, Kuntal Dey, Chris Harrison, and Mayank Goel. 2018. EyeSpyVR: Interactive Eye Sensing Using Off-the-Shelf, Smartphone-Based VR Headsets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 57 (June 2018), 10 pages. https://doi.org/10.1145/3214260

1 INTRODUCTION

Smartphone-based virtual reality (VR) headsets have become increasingly popular since the introduction of the Google Cardboard [14] in 2014. These devices adapt a user's existing smartphone into a powerful VR headset, taking advantage of the phone's existing sensing and rendering capabilities to enable rich VR experiences, without the expense and complexity of a dedicated VR headset.

Eye sensing has been shown to be a powerful and useful input modality in VR experiences, ranging from foveated rendering of complex scenes, to social VR applications and gaze-responsive non-player characters (NPCs) in games [4, 12, 37]. However, existing VR eye sensing approaches require special illumination and sensors

Authors' addresses: Karan Ahuja, Carnegie Mellon University, Pittsburgh, USA ; Rahul Islam, Indian Institute of Information Technology Guwahati, Assam, India; Varun Parashar, Manipal University, Jaipur, India; Kuntal Dey, IBM Research, Bangalore, India; Chris Harrison; Mayank Goel, Carnegie Mellon University, Pittsburgh, USA kahuja@cs.cmu.edu.

2474-9567/2018/6-ART57 $15.00
https://doi.org/10.1145/3214260

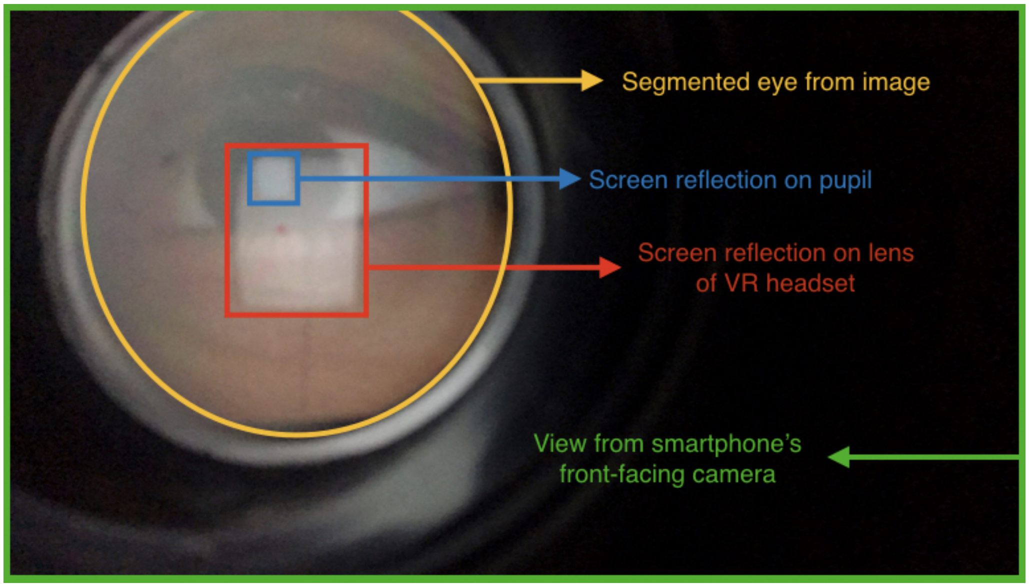

Fig. 1. Example image from a smartphone's front facing camera when placed into a VR headset.

[10, 17], or even instrument the eye itself [39], which is incompatible with a commodity smartphone approach. Furthermore, commercial eye trackers such as FOVE [10] and Tobii eye tracking for HTC Vive [38] generally cost hundreds to thousands of dollars. Ideally, we want a way to robustly sense a wearer's eye using *only* a phone's built-in hardware.

In this work, we show how a smartphone inserted into a VR headset can capture a user's left periocular region through its front-facing camera, as shown in Figure 1. Additionally, we can utilize the screen as an illumination source. Note that unlike e.g., traditional gaze estimation approaches, illumination from the screen does not appear as a well-defined "*eye glint*", and instead appears as a large reflection on the pupil (Figure 1, highlighted in blue). Furthermore, the screen also produces a reflection on the inexpensive optics of commodity VR headsets (often plastic lenses of poorer optical clarity; Figure 1, outlined in red). Figure 2 offers other challenging examples stemming from poor VR headset misalignment, camera autofocus issues, and low illumination due to dark screen contents.

Nonetheless, we show that these images of the eye can enable four interesting sensing modalities: worn/not-worn detection, blink detection, user identification, and coarse gaze estimation. Our approach must contend with a number of technical challenges, including non-uniformity of the light source, occlusion of the pupil by reflections, and inconsistent sensor placement. We describe our software-only eye sensing implementation — which we call EyeSpyVR — and present the results of a series of studies involving 70 participants. We demonstrate useful eye-sensing capabilities without requiring additional or specialized hardware, allowing advanced features to be enabled with little more than a software update. We envision VR application developers leveraging these techniques to open new interactive possibilities and more immersive VR experiences on commodity VR headsets.

2 RELATED WORK

We now briefly review the literature that most directly intersects with our focus of work — blink detection, gaze tracking, and eye-based authentication and identification — especially systems with a low-cost dimension.

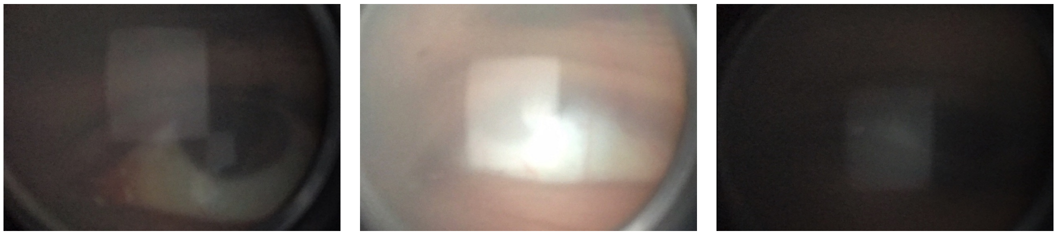

Fig. 2. Examples of challenging images, from left to right: headset misaligned, image out of focus, and low illumination.

2.1 Blink Detection

Blink detection has been used in a wide range of applications. For example, [19] used it to measure drowsiness, with a particular focus on driver safety. Voluntary blinks have also been used extensively as a control mechanism in the accessibility literature (see e.g., [25]). Blink patterns have also been used to measure engagement and attention [35], as well as infer activity [18]. These example applications can now be enabled in low-cost VR experiences with our approach. Also related to our technical approach are computer-vision-based techniques, such as Lalonde *et al.*[26], which detected blinks using SIFT based features, and [3], which used active appearance models. Most related to our work is LiGaze [27], which instrumented a VR headset with four photodiodes to detect blinks by analyzing reflected light intensity. Lastly, Electrooculography (EOG) sensing has also been explored for interactive use [36].

2.2 Gaze Tracking

Gaze sensing has been used extensively in the field of HCI [33], most often to infer a user's area of attention. While there has been much work on gaze estimation using specialized hardware [16, 29], recent work has also demonstrated gaze tracking on mobile platforms, underscoring that the requisite computer power is available [1, 23]. To achieve high precision, head mounted systems are popular, as they offer close and stable views of the eyes (see e.g., [11]). More specifically in the VR domain, [32] showcases various interaction techniques powered by eye-gaze in immersive VR experiences. There are also commercial systems, such as SMI and FOVE, that offer high-quality gaze estimation for specialty applications. Please also see [21], which provides a comprehensive survey of eye-gaze tracking in consumer platforms. In the research literature, systems such as LiGaze [27], Parallel Eyes [22] and EyeContact [39] all rely on special hardware - photodiodes, infrared cameras and special contact lenses respectively. In contrast, we use smartphones that users already own, and thus offer a zero-cost solution.

2.3 Identification and Authentication

Human eyes are highly varied and can be used to identify and authenticate users. Typically the ocular region is used, which includes the pupil, sclera, conjunctival vasculature and iris. Geometric [30] and deep learning [2] based approaches have both proven effective (see [34] for a survey), and studied extensively for smartphone use [2, 6, 20, 34]. The literature on authentication in VR experiences is more nascent. Traditional PINs and passwords have been explored [13], as have as bio-acoustics [28]. Closest to our work is [40], which use a head worn system to track eye movement behavioral biometrics. However, to our knowledge, no VR work has leveraged eye image features for authentication or identification.

3 IMPLEMENTATION

Our prototype system uses an iPhone 7 inserted into a $10 VR Box headset [5], seen in Figure 3. To simplify development and debugging, we configured the iPhone to stream its front camera to a desktop computer over WiFi, which performs video processing and classification.

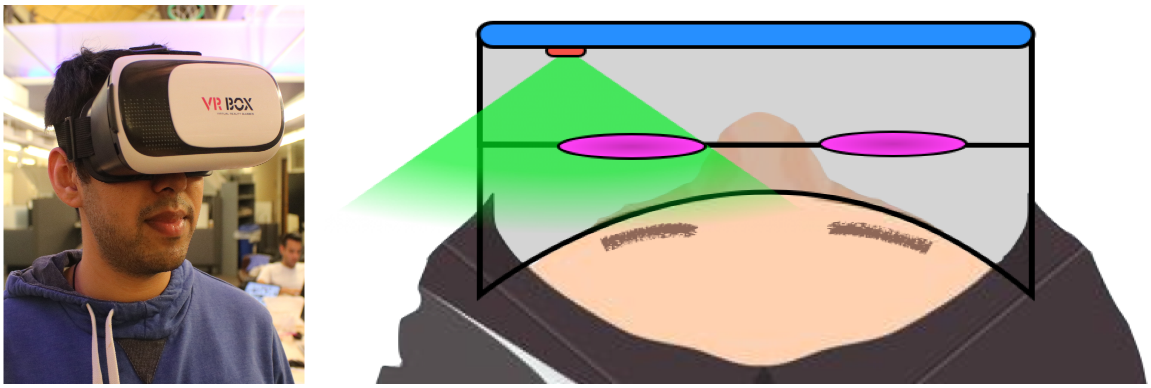

Fig. 3. Left: User wearing VR Box headset. Right: Top down schematic view of the VR headset when worn. The smartphone (blue) camera (red) has a field of view (green) that encompasses one of the headset's two lenses (pink), allowing it to see one of the wearer's eyes.

Our processing pipeline begins by detecting whether an eye is present in the camera's field of view. For this decision, we train a linear-kernel SVM classifier on the RGB histogram of the image (768 features total). If the system detects that an eye is present, meaning that the headset is worn, the system then segments the eye from the image for later processing. This is done by using the Hough Circle transform [9] to identify the eye's lens, after which the eye region is cropped.

Once the eye is segmented, we extract more detailed features that help us detect gaze direction and perform biometric identification. These eye features are computed using a convolutional neural network (CNN). CNNs have been successfully applied to many image processing problems, including eye-gaze direction in distant camera images [23]. In general, CNNs require large amounts of data in order to provide good results [24]. To circumvent this data requirement, we leverage an existing model architecture and weights from [2], which used periocular eye images captured from smartphone cameras to perform biometric identification. This network is state of the art for mobile ocular biometrics on the VISOB [34] and MICHE [7] databases. Using this model as a starting point, we then train our own custom CNNs for blink detection, user identification, and gaze detection using Stochastic Gradient Descent with standard back propagation and momentum, with a learning rate of 0.01 for all layers, and a batch size of 128 for 1000 epochs. This model architecture is shown in Figure 4.

For blink detection, our CNN is trained as a binary classifier: eye opened or closed. For user identification, we train the model as a multiclass classifier, and using the last layer as an eye representation vector. This means no training is needed to enroll a new user; only the user's eye representation needs to be calculated through a forward pass. Then, using this representation vector, we can compute the cosine similarity between eyes, and in turn, classify a wearer as either genuine (using a database of known eyes) or as an impostor by thresholding the score (calibrated to minimize the equal error rate). We also note that it is common for VR devices to be used among families and close circles of friends. For such scenarios, we train a linear-kernel SVM on the eye representation vectors of enrolled users to perform identification among a small set of users.

Finally, for gaze tracking, we train our model as both a classifier and as a regressor. In the former case, we train a model as a multiclass classifier to predict one of five gaze locations on the screen. For the regressor, we train a model with a continuous output to minimize the mean gaze angular error. To calibrate per user, we first train a master, person-independent CNN for gaze angle, and then fine-tune the CNN for each user by freezing the weights of the convolutional layers and adjusting only the last layer. This allows us to make rapid updates to a pre-trained model using only a small amount of user data. To collect user data, we capture periocular images of users while they wear the VR headset and follow a red dot with their eyes, which appear in one of five positions.

As noted previously, our current approach runs on a laptop CPU (dual core i7, 2.8 GHz, no GPU acceleration), which facilitated rapid development and testing. On this laptop, we achieve 31.1 FPS for eye region segmentation. Worn detection is very fast, under 1 ms of execution time. Once the system detects the VR headset is worn, the blink, identification and gaze processes execute, which run at 22 FPS. Our camera stream runs at 30 FPS, so we are just under the native frame rate when all processes are running. We have no doubt this could run at interactive speeds on modern smartphones with some tuning; other work has already demonstrated such speeds with similarly complex CNNs [23]. Also, our models are 82 MB in size, making it practical to load onto a smartphone. Thus in a commercial implementation, we envision all processing occuring on the phone.

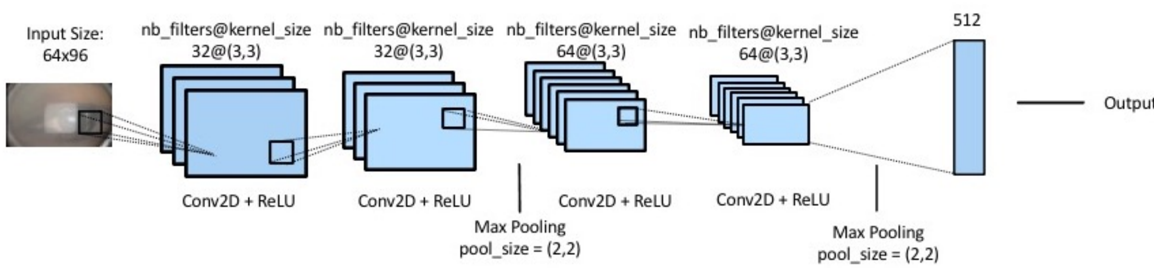

Fig. 4. Architecture of our CNN models. We train different models for detecting blinks, gaze location and user identification.

4 EVALUATION

We performed a series of studies to evaluate the accuracy and performance of EyeSpyVR. In all cases, we use an Apple iPhone 7 and VR Box headset. During data collection, participants were allowed complete freedom to move around and no restrictions were placed on their head movements. When donning the headset, we told participants to find a position that was most comfortable for them, and encouraged them not to adjust the headset once data collection began (note that between sessions, the headset was removed). Participants were drawn from the local population; all had used smartphones, but most did not have any experience with VR systems. All participant data was collected indoors in a lab setting.

4.1 Study 1: Worn Detection & Blink Detection

For our first study, investigating worn detection and blink detection, we recruited 10 participants (2 female, mean age 26, SD 3.3). To start, participants were asked to wear the VR headset and the VR interface instructed the user to keep their eyes open and look straight ahead. After a brief pause, ten images were captured at 200 ms intervals. The VR interface then instructed the user to close their eyes. After a brief delay, ten images of the user's closed eyes were taken. This completed one phase of data collection, which was repeated twice more. Between phases, participants removed the VR headset to add variability and realism to the data set.

We also collected 2000 images when the headset was not worn across several environments, including indoors with the lights on, indoors with the lights off, outdoors while standing, and outdoors while walking.

4.2 Study 2: User Identification & Gaze Estimation

For our second study, which evaluated user identification and gaze estimation, we recruited 63 participant (12 female, mean age 31, SD=6.5). Each participant completed two sessions, 10 minutes each, on consecutive days. Data from two participants had to be dropped due to poor/incorrect headset placement (*i.e.*, the eye was only partially visible), and one other participant was dropped due to poor illumination, resulting in data from 60 participants.

This study consisted of three capture sessions. Between Sessions 1 and 2, there was a 2 minute break with the headset off. Between Sessions 2 and 3, there was a one day break, adding additional variability. Within each session, we ask the wearer to gaze at a red dot presented on the screen in one of five locations: top-left (-18° yaw left, 10° pitch up), top-right (+18°, +10°), center (0°, 0°), bottom-left (-18°, -10°), and bottom-right (+18°, -10°). For each requested dot, we wait one second, and then we capture 5 images of the eye at 300 ms intervals. Not only does this data allow us to evaluate gaze estimation, but also provided a diverse set of images for testing user identification at different gaze angles (*i.e.*, as opposed to looking just straight ahead). An example of this data can be seen in Figure 5, where different parts of the iris and conjunctival vasculature is visible for each gaze point. In total, we collected $5 \times 5 = 25$ images per session, $\times$ 3 sessions, resulting in a total of 4500 images from our 60 participants.

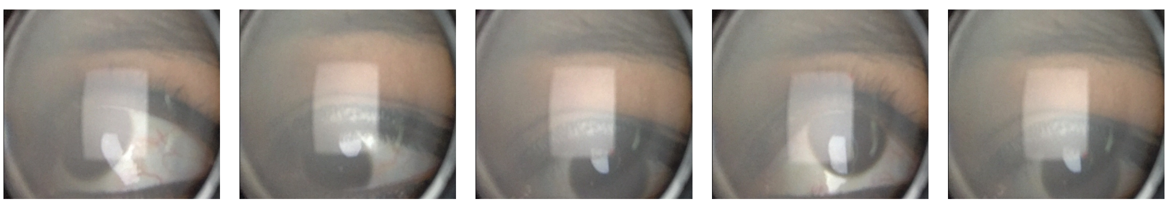

Fig. 5. Example images of a user gazing at our five test targets.

5 RESULTS

5.1 Worn Detection

To evaluate worn *vs.* not worn detection, we used two datasets. The first set contained images we captured when the headset was not worn by any participant (2000 images). The second set included all of the images from our first, ten-person user study (600 images). We trained and tested our worn-detection classifier using leave-one-person-out cross-validation. Across all users, accuracy was 100%, suggesting immediate feasibility.

5.2 Blink Detection

To evaluate the performance of our blink classifier, which predicts if the eyes are opened or closed, we again used a leave-one-person-out cross-validation analysis. This test-train procedure was repeated once per test participant, and the resulting accuracies were averaged. Across our ten participants, our classifier achieved a mean accuracy of 95.3% (SD = 0.11). Our average precision was 94.9%, with an average recall of 96.0%. This compares favorably to the next closest work, LiGaze [27], which reported 83% precision and 80% recall.

5.3 User Authentication

EyeSpyVR supports both user verification, which determines if a user is genuine, and user identification, which distinguishes a person from among a small pool of potential users.

For user verification, we trained the system with user enrollment. We first train a model as a multi-class classifier on a set of 40 randomly-selected users (out of our 60 study participants). We use Sessions 1 and 2 as training (2000 images in total) and Session 3 for validation (1000 images in total). We use the resulting trained CNN to compute eye representation vectors, which are then used for cosine similarity (see Section 3 for more details). We then use this model, and the remaining (i.e., holdout) 20 users for testing our verification system. Specifically, we use Session 1 and Session 2 data (1000 images) as enrollment and Session 3 data (500 images) for verification. The results are plotted in Figures 6 ("All Gaze Points") and 7. The Area Under Receiver Operating Curve (AUROC) attained by our system is 0.84 (for reference, MicheNet is 0.74). If we limit enrollment and testing to matched gaze direction instances (e.g., straight ahead gaze images with straight ahead gaze images), AUROC increases to 0.87 (MicheNet similarly increases to 0.77). See "Matched Points" in Figure 6 and Figure 8.

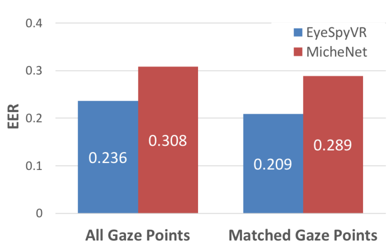

Fig. 6. Equal Error Rates Verification.

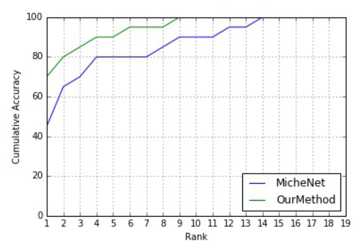

Fig. 7. Cumulative Match Curve for All Gaze Points.

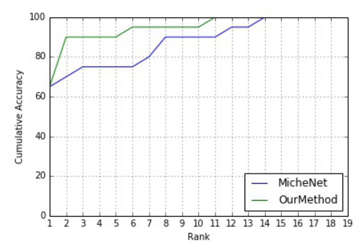

Fig. 8. Cumulative Match Curve for Matched Gaze Points.

Finally, we tested our system's performance for user identification, simulating family units of size four, following the procedure in [15]. Similar to the verification study, we train our model on 40 random participants to learn eye representations, and keep 20 participants' data for testing. We then simulated 1000 virtual families, in which we randomly pick four participants (out of the 20 holdout participants) to become a family. We train on family members' data from Sessions 1 and 2 (200 images) and test on Session 3 (100 images). This process revealed an overall identification accuracy of 81.4% across 1000 simulated rounds. Although falling short of commercial-level accuracy, we do believe this result illustrates that useful signal is present and future work could unlock superior accuracies.

5.4 Gaze Estimation

We first evaluated our person-independent gaze tracking accuracy using leave-one-person-out cross-validation, achieving a mean accuracy of 90.8% for predicting gaze across our five targets. We then evaluated the accuracy of the same model, but calibrated with per-user data. Specifically, for each user, we trained the CNN generically on 59 participants' data, same as a leave-one-person-out testing approach. Then, we fine-tuned the CNN on data from a participant's Sessions 1 and 2, and test on data from Session 3 (captured a day later). Overall, this achieves a mean accuracy of 95.3% on the five gaze targets.

To evaluate our eye gaze regressor, we measured the angular error between the reported gaze position and the actual gaze position, and averaged the angular error across all trials. We obtained a person-independent gaze accuracy of 12.9° using leave-one-person-out cross validation. After per-user calibration (see above), we achieved

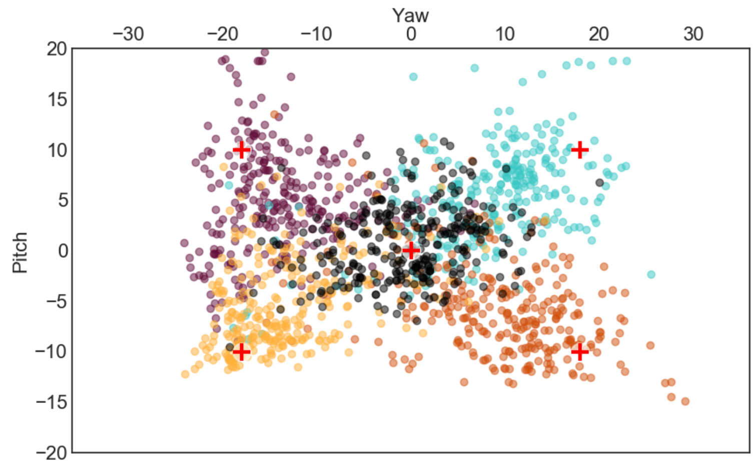

Fig. 9. Per-user calibrated gaze accuracy across five points. Red cross-hairs are ground truth.

an angular error of 10.8° when trained on Sessions 1 and 2, and tested on Session 3. The latter per-user calibrated results are plotted in Figure 9 (all paticipants' data superimposed).

As before, we can compare our results to LiGaze [27], which achieved 10.1° cross-user accuracy using a VR headset instrumented with photodiodes. However, we note that devices using special-purpose infrared emitters and cameras, such as FOVE[10], offer accuracies on the order of 1°, but cost at least hundreds of dollars. EyeContact[39] offers even higher accuracies (0.09°), though this requires wearing a special contact lens.

6 DISCUSSION

We envision EyeSpyVR enhancing a wide variety of VR experiences (see also Related Work). For instance, worn detection can be used as a clutch, triggering interactive content the moment the headset is worn, creating more personal and cinematic experiences. Likewise, content or game-play could be automatically paused upon removal of the headset. Blink and gaze tracking are a potent combination. They could, for example, be used to infer cognitive state, such as attention and boredom, which could e.g., dynamically alter game elements. Blink and gaze are also powerful social signals, and could also be used to create more authentic social VR experiences, where avatars match their wearer's eye behavior and perhaps even allowing for mutual gaze between remote participants.

Commodity smartphone VR systems typically lack accessory controllers found in their higher-end counterparts. Thus gaze tracking (in concert with a selection mechanism like dwell) could offer an additional input means [8, 32]. Gaze tracking could also be valuable in creating accessible VR experiences for those with motor disabilities. Additionally, a major limitation of smartphone VR headsets is their limited battery and rendering power (compared to dedicated VR headsets tethered to powerful desktop computers). To mitigate this, worn detection could be used to preserve limited battery power, rendering no content when no user is looking. By tracking gaze with EyeSpyVR, foveated rendering [31] could allow for lower-fidelity rendering imperceptible to the user, again saving power and execution time.

Finally, user identification could be used to secure VR devices, applications and files from unwanted interlopers. It could also be used to automatically load profiles for users on a shared device, *e.g.*, favorite programs and resuming media and games at user-specific points. Likewise, parameters such as interocular distance and motion sickness mitigations could be loaded seamlessly without user intervention.

Although EyeSpyVR brings new capabilities to light for inexpensive, commodity hardware, there are several important limitations. Foremost, while our accuracy results are encouraging, they are not at levels seen in dedicated eye sensing systems. Thus low cost comes with reduced performance, which developers (and users) will have to accommodate. Additionally, due to the horizontal placement of smartphones in most commodity VR headsets, only one eye can be seen by the front facing camera. For many applications, this is sufficient, as users typically blink with both eyes and gaze approximately in the same direction. However, it may produce inaccurate gaze data if users look at very close virtual objects. This limitation might be overcome by placing a fish-eye lens in front of the smartphone camera to expand its field of view to both eyes. We also found that some smartphone VR headsets occlude the front facing camera, often with some type of bracket to secure the phone. Similarly, some smartphones place their front facing cameras at the edges, which may provide a suboptimal view of the eyes.

7 CONCLUSION

We have presented our work on EyeSpyVR, an eye sensing approach that works with low cost, smartphone-based VR headsets, enabling a suite of useful eye sensing modalities, including worn/not worn detection, blink detection, user identification and eye gaze tracking. This is achieved with no special hardware, and most off-the-shelf smartphone-based VR headsets should be compatible with our approach. We devised a robust, generalizable CNN-based method, learning features from images of the periocular region as captured by a smartphone camera. We ran two user studies, including 70 participants, to validate our prototype implementation and technical approach, demonstrating the feasibility of EyeSpyVR.

REFERENCES

[1] Karan Ahuja, Ruchika Banerjee, Seema Nagar, Kuntal Dey, and Ferdous Barbhuiya. 2016. Eye center localization and detection using radial mapping. In *ICIP*. IEEE, 3121–3125.

[2] Karan Ahuja, Rahul Islam, Ferdous A Barbhuiya, and Kuntal Dey. 2017. Convolutional neural networks for ocular smartphone-based biometrics. *Pattern Recognition Letters* 91 (2017), 17–26.

[3] Ioana Bacivarov, Mircea Ionita, and Peter Corcoran. 2008. Statistical models of appearance for eye tracking and eye-blink detection and measurement. *IEEE transactions on consumer electronics* 54, 3 (2008).

[4] Jeremy N Bailenson, Nick Yee, Jim Blascovich, Andrew C Beall, Nicole Lundblad, and Michael Jin. 2008. The use of immersive virtual reality in the learning sciences: Digital transformations of teachers, students, and social context. *The Journal of the Learning Sciences* 17, 1 (2008), 102–141.

[5] VR Box. 2018. VR Box Headset. (2018). https://www.amazon.com/Virtual-Reality-Headset-Glasses-Smartphones/dp/B01IXLEJCM

[6] Tiago de Freitas Pereira and Séastien Marcel. 2015. Periocular biometrics in mobile environment. In *Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 1–7.

[7] Maria De Marsico, Michele Nappi, Daniel Riccio, and Harry Wechsler. 2015. Mobile Iris Challenge Evaluation (MICHE)-I, biometric iris dataset and protocols. *Pattern Recognition Letters* 57 (2015), 17–23.

[8] Heiko Drewes and Albrecht Schmidt. 2007. Interacting with the Computer Using Gaze Gestures. In *Human-Computer Interaction – INTERACT 2007*, Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 475–488.

[9] Richard O Duda and Peter E Hart. 1972. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* 15, 1 (1972), 11–15.

[10] FOVE. 2018. FOVE eye tracker. (2018). https://www.getfove.com/

[11] John M Franchak, Kari S Kretch, Kasey C Soska, and Karen E Adolph. 2011. Head-mounted eye tracking: A new method to describe infant looking. *Child development* 82, 6 (2011), 1738–1750.

[12] Maia Garau, Mel Slater, Vinoba Vinayagamoorthy, Andrea Brogni, Anthony Steed, and M Angela Sasse. 2003. The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 529–536.

[13] Ceenu Goerge, Mohamed Khamis, Emanuel von Zezschwitz, Marinus Burger, Henri Schmidt, Florian Alt, and Heinrich Hussmann. 2017. Seamless and Secure VR: Adapting and Evaluating Established Authentication Systems for Virtual Reality. In *USEC*.

[14] Google. 2014. Google Cardboard. (2014). https://vr.google.com/cardboard/

[15] Anhong Guo, Robert Xiao, and Chris Harrison. 2015. Capauth: Identifying and differentiating user handprints on commodity capacitive touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*. ACM, 59–62.

[16] Dan Witzner Hansen and Qiang Ji. 2010. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence* 32, 3 (2010), 478–500.

[17] Steven Hickson, Nick Dufour, Avneesh Sud, Vivek Kwatra, and Irfan Essa. 2017. Eyemotion: Classifying facial expressions in VR using eye-tracking cameras. *arXiv preprint arXiv:1707.07204* (2017).

[18] Shoya Ishimaru, Kai Kunze, Koichi Kise, Jens Weppner, Andreas Dengel, Paul Lukowicz, and Andreas Bulling. 2014. In the blink of an eye: combining head motion and eye blink frequency for activity recognition with Google Glass. In *Proceedings of the 5th augmented human international conference*. ACM, 15.

[19] Takehiro Ito, Shinji Mita, Kazuhiro Kozuka, Tomoaki Nakano, and Shin Yamamoto. 2002. Driver blink measurement by the motion picture processing and its application to drowsiness detection. In *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*. IEEE, 168–173.

[20] Anil Jain, Lin Hong, and Sharath Pankanti. 2000. Biometric identification. *Commun. ACM* 43, 2 (2000), 90–98.

[21] Anuradha Kar and Peter Corcoran. 2017. A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms. *IEEE Access* (2017).

[22] Shunichi Kasahara, Mitsuhito Ando, Kiyoshi Suganuma, and Jun Rekimoto. 2016. Parallel Eyes: Exploring Human Capability and Behaviors with Paralleled First Person View Sharing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1561–1572.

[23] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2176–2184.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[25] Aleksandra Królak and Paweł Strumiłło. 2012. Eye-blink detection system for human–computer interaction. *Universal Access in the Information Society* 11, 4 (2012), 409–419.

[26] Marc Lalonde, David Byrns, Langis Gagnon, Normand Teasdale, and Denis Laurendeau. 2007. Real-time eye blink detection with GPU-based SIFT tracking. In *Computer and Robot Vision, 2007. CRV'07. Fourth Canadian Conference on*. IEEE, 481–487.

[27] Tianxing Li, Emmanuel S Akosah, Qiang Liu, and Xia Zhou. 2017. Ultra-Low Power Gaze Tracking for Virtual Reality. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 490–492.

[28] Rui Liu, Cory Cornelius, Reza Rawassizadeh, Ron Peterson, and David Kotz. 2017. Poster: Vocal Resonance as a Passive Biometric. In *Mobile Systems, Applications, and Services*. ACM, 160–160.

[29] Päivi Majaranta and Andreas Bulling. 2014. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*. Springer, 39–65.

[30] Unsang Park, Raghavender Reddy Jillela, Arun Ross, and Anil K Jain. 2011. Periocular biometrics in the visible spectrum. *IEEE Transactions on Information Forensics and Security* 6, 1 (2011), 96–106.

[31] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 179.

[32] Thammathip Piumsomboon, Gun Lee, Robert W Lindeman, and Mark Billinghurst. 2017. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *3D User Interfaces (3DUI), 2017 IEEE Symposium on*. IEEE, 36–39.

[33] Alex Poole and Linden J Ball. 2006. Eye tracking in HCI and usability research. *Encyclopedia of human computer interaction* 1 (2006), 211–219.

[34] Ajita Rattani, Reza Derakhshani, Sashi K Saripalle, and Vikas Gottemukkula. 2016. ICIP 2016 competition on mobile ocular biometric recognition. In *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 320–324.

[35] Tsugunosuke Sakai, Haruya Tamaki, Yosuke Ota, Ryohei Egusa, Shigenori Inagaki, Fusako Kusunoki, Masanori Sugimoto, Hiroshi Mizoguchi, et al. 2017. Eda-Based Estimation Of Visual Attention By Observation Of Eye Blink Frequency. *International Journal on Smart Sensing and Intelligent Systems* 10, 2 (2017), 296–307.

[36] Junichi Shimizu and George Chernyshov. 2016. Eye movement interactions in google cardboard using a low cost EOG setup. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 1773–1776.

[37] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Facevr: Real-time facial reenactment and eye gaze control in virtual reality. *arXiv preprint arXiv:1610.03151* (2016).

[38] Tobii. 2014. Tobii eye tracker for HTC Vive. (2014). https://blog.tobii.com/eye-tracking-vr-devkit-for-htc-vive-311cbca952df

[39] Eric Whitmire, Laura Trutoiu, Robert Cavin, David Perek, Brian Scally, James Phillips, and Shwetak Patel. 2016. EyeContact: scleral coil eye tracking for virtual reality. In *ISWC*. ACM, 184–191.

[40] Yongtuo Zhang, Wen Hu, Weitao Xu, Chun Tung Chou, and Jiankun Hu. 2018. Continuous Authentication Using Eye Movement Response of Implicit Visual Stimuli. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 177.

Received February 2018; revised April 2018; accepted April 2018.