

# Ultra-Low Power Gaze Tracking for Virtual Reality

Tianxing Li, Qiang Liu, and Xia Zhou

Department of Computer Science, Dartmouth College, Hanover, NH  
{tianxing,qliu,xia}@cs.dartmouth.edu

## ABSTRACT

Tracking user’s eye fixation direction is crucial to virtual reality (VR): it eases user’s interaction with the virtual scene and enables intelligent rendering to improve user’s visual experiences and save system energy. Existing techniques commonly rely on cameras and active infrared emitters, making them too expensive and power-hungry for VR headsets (especially mobile VR headsets).

We present LiGaze, a low-cost, low-power approach to gaze tracking tailored to VR. It relies on a few low-cost photodiodes, eliminating the need for cameras and active infrared emitters. Reusing light emitted from the VR screen, LiGaze leverages photodiodes around a VR lens to measure reflected screen light in different directions. It then infers gaze direction by exploiting pupil’s light absorption property. The core of LiGaze is to deal with screen light dynamics and extract changes in reflected light related to pupil movement. LiGaze infers a 3D gaze vector on the fly using a lightweight regression algorithm. We design and fabricate a LiGaze prototype using off-the-shelf photodiodes. Our comparison to a commercial VR eye tracker (FOVE) shows that LiGaze achieves  $6.3^\circ$  and  $10.1^\circ$  mean within-user and cross-user accuracy. Its sensing and computation consume  $791\mu\text{W}$  in total and thus can be completely powered by a credit-card sized solar cell harvesting energy from indoor lighting. LiGaze’s simplicity and ultra-low power make it applicable in a wide range of VR headsets to better unleash VR’s potential.

## CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile devices**; *Virtual reality*; • **Computer systems organization** → *Sensors and actuators*;

## KEYWORDS

Gaze tracking, virtual reality, visible light sensing

### ACM Reference format:

Tianxing Li, Qiang Liu, and Xia Zhou. 2017. Ultra-Low Power Gaze Tracking for Virtual Reality. In *Proceedings of SenSys '17, Delft, Netherlands, November 6–8, 2017*, 14 pages.

DOI: 10.1145/3131672.3131682

## 1 INTRODUCTION

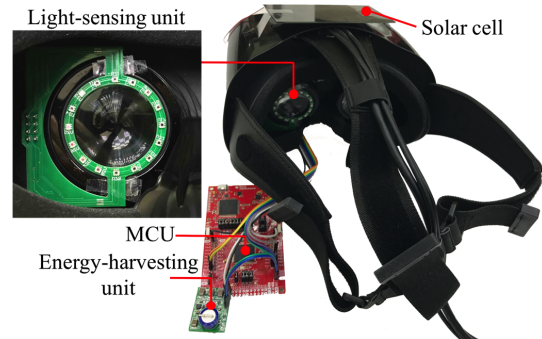
Virtual reality (VR) emerges as a promising next computing platform. Taking the form of head-mounted displays (HMD), modern

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*SenSys '17, Delft, Netherlands*

© 2017 ACM. 978-1-4503-5459-2/17/11...\$15.00

DOI: 10.1145/3131672.3131682



**Figure 1:** LiGaze integrated into a FOVE VR headset. LiGaze uses only passive photodiodes on a light-sensing unit to track user gaze. It can be powered by a credit-card sized solar cell (atop the headset) harvesting energy from indoor lighting.

VR headsets provide immersive, realistic simulation of the 3D physical world and are poised to transform how we interact, entertain, and learn. With the advent of affordable, accessible VR headsets (e.g., Google Daydream, Cardboard, Samsung Gear VR), VR is gaining popularity and projected to be a multi-billion market by 2025 [2].

Our work in this paper focuses on a feature crucial to VR: gaze tracking, i.e., determining user’s eye fixation direction. Not only does gaze tracking allow users to interact with the content just by glances, it also can greatly improve user’s visual experience, reduce VR sickness, and save systems (display) energy. The energy saving can be achieved by foveated rendering [23, 52], which progressively reduces image details outside the eye fixation region. Such energy saving is particularly beneficial for mobile VR headsets without external power cords.

The dominating methodology to gaze tracking relies on (infrared) cameras, often accompanied by infrared (IR) LEDs to illuminate eyes. The combination of IR LEDs and cameras raises concern on energy consumption and form factor. Recent hardware advances have led to modular designs of eye tracker that can be included to mobile and wearable devices. These commercial eye trackers (e.g., Tobii, SMI), however, are still expensive (e.g., \$10K for integrating a Tobii eye tracker into a VR headset) and power-hungry (requiring external battery packs). As a result, most VR headsets today shy away from integrating gaze tracking and resort to head direction as a coarse, often incorrect, estimate of user’s gaze.

In this paper, we seek a low-power, low-cost approach that can lower the barrier to integrate gaze tracking into various VR headsets including those for mobile VR. We consider a minimalist sensing approach, which eliminates active IR LEDs and replaces cameras with a few low-cost (\$2 each), small (e.g.,  $2\text{ mm}^2$ ) photodiodes. Exploiting the fact that VR screen light is the sole and constant light source within the space of a VR headset, we reuse screen light to track user gaze. Specifically, we place a few (e.g., 16) photodiodes at the boundary of each VR lens (Figure 1), so that each photodiode senses screen light reflected by user’s eyeball in a certain direction.

Given that our pupil is a hole in the iris center and absorbs incoming light, when screen light strikes the pupil region, most light rays are absorbed from the photodiode’s point of view, weakening the reflected light perceived by photodiodes in that direction. Thus, pupil movement affects the spatial pattern of changes in reflected screen light. Such pattern can be exploited to infer gaze direction.

Although pupil’s light absorption has been exploited by prior methods [19, 42], we face unique challenges in VR, mainly because of the uncontrolled nature of our light source (screen light). VR content is colorful and dynamic, resulting in screen light and hence reflected screen light that are spatially nonuniform and time-varying. Such inherent variations make it hard to identify changes in reflected light caused by pupil movement. Furthermore, user diversity presents additional challenges in inferring gaze. Our experiments show that eye and skin characteristics all affect the actual impact of pupil’s light absorption on sensed reflected light. The gaze tracking algorithm needs to be customized for individual users. Finally, blink can interfere with sensing reflected light and tracking gaze.

To address these challenges, we design LiGaze. The core of LiGaze is to estimate reflected light changes associated with screen content and then identify changes caused by pupil movement. To do so, we design a circuit board with photodiodes embedded back to back on both sides, where one side faces the screen sensing incoming screen light in different directions and the other faces user’s eyeball sensing reflected screen light. We characterize the relationship between the front and back sensors’ readings assuming a center pupil. This characterization is then used to derive features related to changes of reflected light caused by pupil movement. We apply supervised learning (boosted trees [12, 21]) to learn the correlation between our features and the 3D gaze vector, and infer gaze vectors on the fly using regression. To handle user diversity, we calibrate our characterization and apply bootstrap aggregating [10] to fine-tune the trained model for a specific user. Furthermore, LiGaze detects blinks in a parallel process. Given that a blink causes similar changes in reflected light at most photodiodes, LiGaze leverages this spatial correlation to detect blink occurrences.

To examine LiGaze’s feasibility and performance, we build a LiGaze prototype using off-the-shelf, low-cost photodiodes. We design and fabricate a thin (<1 mm), ring-shaped printed circuit board to host 16 photodiodes (OPT3001) on each side. These sensors are connected to a low-power micro-controller (MSP432) that aggregates sensor data, infers a 3D gaze vector, and detects blink. We add the prototype to a commercial VR headset (FOVE) with eye-tracking functionality and evaluate our tracking performance. Based on our experiments with 30 participants, our key findings are as below:

- LiGaze achieves  $6.3^\circ$  and  $10.1^\circ$  mean within-user and cross-user accuracy in tracking gaze, which translates to 100%, 99.5%, and 91.7% accuracy in differentiating 4, 9, and 16 regions on a screen;
- LiGaze generates a 3D gaze inference within 7.8 ms on average, with spatial precision of  $5.2^\circ$  and  $5.9^\circ$  within and cross users;
- LiGaze consumes  $791\mu W$  including both sensing and computation and thus can be completely powered by a credit-card sized solar cell harvesting energy from indoor lighting;
- LiGaze detects blinks with 83% precision and 80% recall overall.

The main contribution of our work is a demonstration that a few passive photodiodes can enable gaze tracking for VR. We explore

the tradeoff between performance and energy consumption of gaze tracking in the context of wearable devices. Our work follows the spirit of recent works [40, 41] and yet differs in that we focus on the VR scenario and exploit the VR setup to further drive down the energy consumption. The simplicity and ultra-low power of our approach can help bring gaze tracking to a wide range of VR headsets and better unleash VR’s potential.

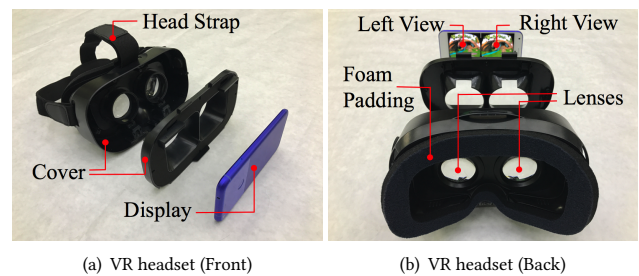
## 2 LIGAZE: RATIONALE AND CHALLENGES

Tailored to wearable VR, LiGaze reuses screen light inside a VR headset and senses screen light reflected by our eyes. Next we first describe the inner structure of modern VR headsets. We then introduce LiGaze’s design rationale and challenges.

### 2.1 Inner Structure of VR Headsets

Modern VR headsets are classified into two types based on the head-mounted displays (HMDs): 1) tethered HMDs (e.g., HTC Vive, Oculus Rift), displays built in the headset and connected to powerful desktop servers for rendering, and 2) mobile HMDs, which reuse screens of mobile phones slotted into the VR headset (e.g., Google Cardboard, Daydream, Samsung Gear VR). Tethered HMDs offer better visual quality, thanks to the computation power of external servers allowing more advanced rendering. However, they are constrained in mobility due to the need of tethered cords. Mobile VR, on the other hand, is self-contained offering full mobility, and yet suffers from relatively lower visual quality and limited battery life.

Despite the differences in HMDs, VR headsets share a similar inner structure: with a display/screen in the front, screen light passes through a pair of lenses (36 – 43 mm in diameter) positioned very closely (1 – 2 cm) to eyes. The lenses divide the screen content into two slightly different 2D images tailored to the left and right eye. By angling the 2D images, the pair of lenses helps to create a 3D virtual scene perceived by the user. Figure 2 shows the main components of a mobile VR headset. The foam padding and head strap block ambient light from entering the headset to create an immersive virtual scene. They also improve the comfort of wearing the headset. Next we explain how to exploit the VR setup to enable gaze tracking using only a few light sensors (i.e., photodiodes).

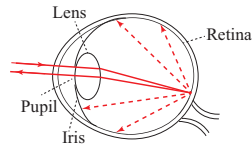


**Figure 2: VR headset structure.** A VR headset contains a display, a cover, a pair of lenses, a foam padding, and a head strap.

### 2.2 The Light Absorption Effect

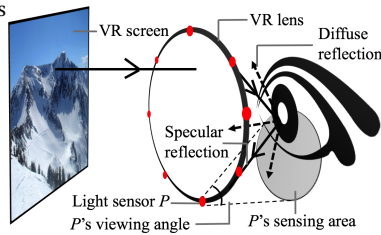
LiGaze leverages the light-absorption effect of our pupil when the observation point is not right next to the illumination source. In essence, the pupil is a hole in the center of iris, which allows

light rays to enter the retina. For an observation point off the axis of the illumination's optical path, light rays entering the pupil are absorbed either directly by eye's inner tissue, or after diffuse reflections at the retina by missing the exit from the pupil (Figure 3)<sup>1</sup>. Pupil's light absorption explains the dark pupil effect exploited by prior eye tracking methods [19, 42], where a camera is placed off the optical axis of the light source to observe a dark pupil.



**Figure 3: Light entering pupil is absorbed directly or after diffuse reflection at the retina.**

In the VR context, pupil's light absorption influences reflected screen light observed by photodiodes on the boundary of each VR lens (Figure 4). Here screen light passes through VR lenses, strikes eyeballs, and is reflected. Photodiodes (i.e., observation points) are placed to avoid obstructing the lens and thus off the axis of screen light. Each photodiode perceives reflected light rays within its viewing angle. Because of its short distance (e.g., 1 cm) to the eye and limited viewing angle (e.g.,  $\pm 45^\circ$ ), a photodiode perceives light reflected by only a region of the eyeball<sup>2</sup>. Take the bottom photodiode  $P$  (Figure 4) as an example, it senses screen light reflected by the bottom region of the eyeball. As a result, when pupil moves to the bottom, sensor  $P$  perceives larger decrease in reflected light because of pupil's light absorption; when pupil moves to other regions, photodiodes close to those regions perceive larger decrease in reflected light.



**Figure 4: A light sensor on a VR lens border senses light reflected by a region of the eye. Pupil absorbs light and its position affects the reflected light sensed by the photodiode.**

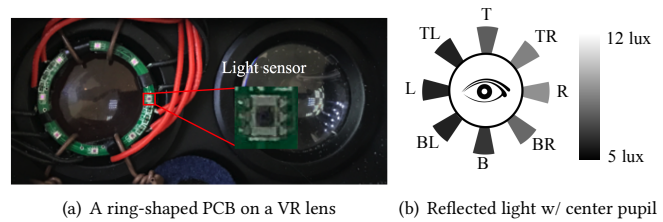
To examine this phenomenon in practice, we conduct experiments using low-cost photodiodes (OPT3001) and a mobile VR headset (DESKTEK V3) with a Pixel XL phone. We design and fabricate a ring-shaped printed circuit board (PCB) to host eight photodiodes and their circuits (Figure 5(a)). The PCB is narrow (2 mm) with a diameter matching the VR lens to avoid affecting the viewing of VR content. On the PCB, photodiodes are spread out evenly, sensing reflected light in the top (T), bottom (B), left (L), right (R), top-left (TL), top-right (TR), bottom-left (BL), and bottom-right (BR) directions around the eye. Each photodiode has a built-in analog-to-digital converter (ADC) that outputs light intensity readings at 10 Hz. We connect photodiodes to a micro-controller (MSP432) to collect sensor data. We seek to answer following questions:

**Q1: Can reflected screen light be sensed by light sensors?**

Our starting point is to examine whether low-cost photodiodes can sense screen light reflected by eyes and capture pupil's light-absorption effect. In our experiment, a participant wears the headset with our PCB attached to the left VR lens. The intensity of screen

<sup>1</sup>The light reflected by retina can only be sensed if the observation point aligns with the optical path of illumination source, since the eye acts as a retro-reflector [69].

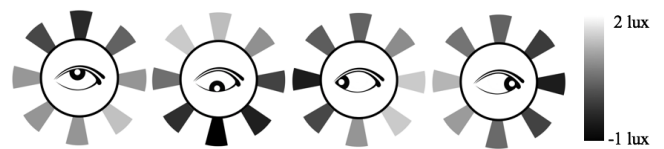
<sup>2</sup>A photodiode also perceives light diffused by other regions of the eyeball. However, the intensity of diffused light from other regions is negligible.



**Figure 5: (a) shows the experiment setup, where a ring-shaped PCB hosting 8 photodiodes is attached to a VR lens. (b) plots the reflected light intensity at each photodiode when pupil is in the center.**

light is measured as 100 lux at the lens. We first instruct the participant to stare at a center dot on the screen for 3 seconds while collecting data from photodiodes. We average each photodiode's readings and plot the result in Figure 5(b). Here the darkness of a bar indicates the light intensity perceived at this location. We make two observations. First, despite being 90%+ weaker than incoming screen light, reflected screen light can be reliably measured by photodiodes. It is far above the minimal intensity (0.01 lux) that the photodiode can sense. Also, sensor readings are very stable, with the standard deviation of 0.02 lux. Second, the reflected light is nonuniform across photodiodes, even under uniform screen light. We hypothesize that it is due to the asymmetry of the eye structure and its surrounding area, where reflectivity differs across directions.

We then instruct the participant to stare at four dots (top, bottom, left, and right) on the screen, each for 3 seconds. To examine whether photodiode can sense pupil's light-absorption effect, we examine the change of reflected light intensity observed by each photodiode, by subtracting its reading under a center pupil (Figure 5(b)). As shown in Figure 6, photodiodes in pupil's direction perceive larger decrease in light intensity (i.e., darker bars) as pupil absorbs more light, while photodiodes in the opposite direction perceive stronger reflected light as pupil's light absorption has less impact. The result demonstrates that photodiodes can capture the subtle reflected light change caused by pupil's light absorption.



**Figure 6: Impact of pupil's light absorption on the reflected light change perceived by each photodiode, in comparison to that under a center pupil in Figure 5(b).**

Next we vary the brightness of the white screen and repeat the above experiment, seeking to understand the minimal level of screen light that allows photodiodes to sense reflected light reliably. From our experiments, we observe that if the screen light is above 10 lux after passing the VR lens, then the reflected screen light is above sensor's noise range (0.05 lux) and thus can be reliably measured. We further examine whether the 10-lux threshold can be met by existing VR screen content. We test 20 popular VR applications supported by either mobile or tethered HMDs (e.g., the Play Store and SteamVR platforms). We place a photodiode (OPT3001) at the back of the VR lens facing the VR display to measure the screen



light intensity after passing the lens. The experiment is conducted in a dark room so that screen light is the only light source. We see that 99% of screen content leads to screen light above 15 lux, demonstrating that screen light of most VR content is sufficiently bright to be reused for detecting gaze direction.

**Q2: Would ambient light interfere with the sensing of reflected screen light?** Another natural question is whether ambient light can leak in the VR headset and interfere with the sensing of weak reflected screen light. To examine the impact of ambient light, we place a light sensor on each VR lens and turn off the display within a VR headset. We then ask a participant to wear the headset under normal office lighting (400 lux) and record light sensor readings. We repeat this experiment using five popular VR headsets in the market, including headsets of both mobile and tethered HMDs. Table 1 shows the mean and standard deviation of sensor reading for each headset. We see that for all headsets except Cardboard, light intensity within the headset is zero, demonstrating that the foam padding and head strap of these headsets effectively block ambient light from entering headsets and screen light from the display is the only light source for those headsets. Cardboard, on the other hand, has neither foam padding nor head strap, letting a slight amount of ambient light rays enter the headset. However, the intensity of the leaking ambient light is only 0.2 lux, similar to sensor’s noise level, and thus its impact is negligible.

VR Headset		Vive	FOVE	Daydream	DESTEK	Cardboard
Sensor Data	Mean	0 lux	0 lux	0 lux	0 lux	0.2 lux
	Std	0 lux	0 lux	0 lux	0 lux	0.1 lux

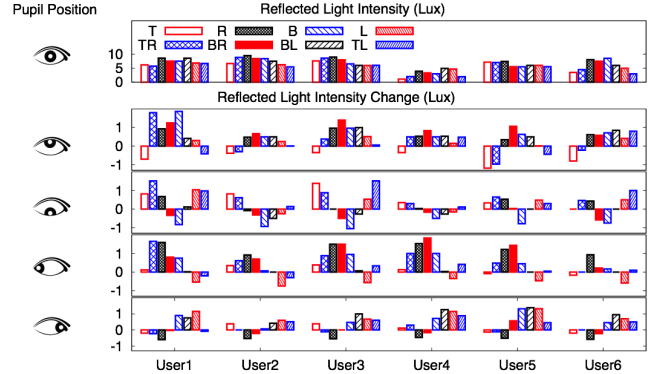
**Table 1: The impact of ambient light on photodiodes on VR lenses. The screen is off and a user wears a VR headset under office lighting.**

**Q3: Is it a universal phenomenon across users?** Finally, we repeat experiments with different users, aiming to understand whether pupil’s light-absorption effect can be observed across users with different eye characteristics. We invite six participants (Table 2), with ages ranging from 18 to 33 years old. Their eyes vary in color (green, black, blue), size, and length of eyelashes. Two participants wear contact lenses. Their skin color also differs (white, yellow, black). We ask each participant to wear our VR headset (Figure 5(a)) and repeat the prior experiment in Figure 5(b). In Figure 7 (top), we plot the reflected light intensity at each photodiode when a participant stares at screen center. We observe that the absolute reflected light intensity differs across users, indicating that eye characteristics affect light reflection. In particular, for the participant with dark skin (User 4), the reflected light is lower than other users with lighter skin colors. It is likely because some light rays are reflected by the skin around the eye and dark skin has lower reflectivity.

We then examine the change in reflected light when pupil moves to other direction. For each other gaze direction, Figure 7 (bottom) plots reflected light observed by each photodiode, minus its reading under a center pupil. We observe that despite the differences in absolute light change, pupil’s light absorption effect is consistently observed across users. For instance, when the pupil moves to the top, the top photodiode observes a decrease in reflected light intensity for all users; when the pupil is on the left, the left photodiode similarly observes a drop in reflected light intensity across all users. The results are encouraging, indicating pupil’s light absorption

User ID	1	2	3	4	5	6
Eye Color	Blue	Black	Black	Black	Green	Black
Skin Color	White	Yellow	Yellow	Black	White	White
w/ Contact Lenses	No	Yes	No	No	No	Yes

**Table 2: Participant information.**



**Figure 7: Examining pupil’s light-absorption effect across users. The top figure plots reflected light intensity at 8 photodiodes (Figure 5(a)) when pupil is in the center. The other four figures plot changes in reflected light when pupil moves to four directions. We observe that sensors in user’s gaze direction perceive larger decline in reflected light intensity. The exact amount of change varies across users.**

effect can be consistently observed across users. On the other hand, the results also indicate challenges in dealing with diverse eye and skin characteristics, which we will elaborate on later (§ 2.3).

In summary, our experiment results confirm the feasibility of using low-cost photodiodes around VR lens to sense pupil’s light-absorption effect. Under static screen light, this effect results into a spatial pattern of changes in photodiode readings correlated to pupil movement and can be exploited to infer gaze direction.

## 2.3 Challenges

To exploit pupil’s light absorption property for VR gaze tracking, we face several challenges. The first challenge comes from screen light dynamics. Our prior experiments assume uniform, static screen light to illustrate the concept. However, actual VR screen content is colorful and dynamic, which emits screen light varying both spatially and temporally. As a result, the reflected light perceived by photodiodes also inherently varies spatially and temporally, even under a fixed pupil position. The spatial variation makes it hard to estimate the reflected light when pupil is in the center, given that the asymmetric eye structure also affects reflected light intensity in a nonuniform manner (Figure 5(b)). The temporal variation of screen light makes it infeasible to leverage prior observations for inferring nonuniform reflected light. Therefore, the spatial pattern of reflected light changes is no longer an effective indicator of pupil position. Furthermore, reflected light intensity varies from 0.5 lux to 5.5 lux even without any pupil movement, while the reflected light change caused by pupil movement is within 2 lux (Figure 6, Figure 7). Thus, the challenge lies in extracting the reflected light change related to pupil movement given screen light dynamics.

The second challenge is to derive a gaze vector based on the light absorption effect while handling diverse eye and skin characteristics across users. A latest study [20] has revealed that user diversity

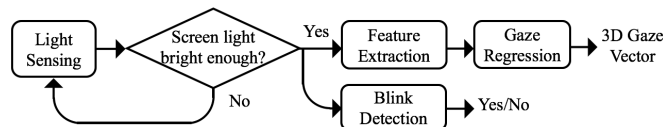


Figure 8: LiGaze system flow.

contributes the most to eye tracking errors. Similarly in our context, user diversity makes it hard to predict the reflected light under a center pupil, as shown in Figure 7 (top). Even if we can derive the change in reflected light with respect to that under a center pupil, its relation to the actual gaze vector can slightly vary across users. Simply seeking the photodiode with the largest decrease in reflected light leads to a very coarse estimate of gaze direction, with error  $\approx 360^\circ/N$ , where  $N$  is the number of photodiodes.

Additionally, blink is another factor that can interfere with the sensing of reflected light intensity. Finally, the gaze tracking algorithm needs to entail low computational overhead so that the system can infer gaze direction in real time with low power.

### 3 LIGAZE OVERVIEW

At high level, LiGaze addresses above challenges as follows. To deal with nonuniform, time-varying screen light, LiGaze uses an additional set of photodiodes facing the display to sense incoming screen light in different directions. Based on the sensed screen light, LiGaze estimates the reflected screen light assuming pupil is in the center, and extracts features related to pupil’s light-absorbing effect. These features are used to infer gaze vectors in real time using supervised learning. To deal with user diversity, LiGaze leverages a quick calibration to customize the model for estimating reflected light and parameters in the trained learning model. LiGaze also runs a parallel process that detects the blink event by examining photodiode data over a time window. Next, we elaborate on the sensor platform design and system flow.

**Sensor Platform.** LiGaze design relies on a unique dual-side light-sensing unit, where photodiodes are embedded in both sides in a back-to-back manner (Figure 11(a)). The sensing unit is realized as a thin, ring-shaped (annulus) PCB and easily attached to the back of each VR lens. Photodiodes on one side of the PCB face the VR display and sense incoming screen light in different directions, while photodiodes on the other side of the PCB face user’s eyes and sense screen light reflected by the eyeball. Because of the small form factor of the photodiode, the panel can be made in a very narrow width (e.g., 2 mm) to ensure that it does not obstruct user’s view. We will present a user study in § 6.5 to examine the impact of our sensing panel on user’s perception.

**System Flow.** LiGaze starts with a short (3-minute) calibration<sup>3</sup> to calibrate system parameters. Figure 8 shows the system flow after calibration. LiGaze first checks whether the sensed incoming screen light is sufficiently strong (i.e., above 10 lux in our experiments) for later gaze tracking. If so, LiGaze starts two parallel processes, one for inferring the 3D gaze vector, and the other for detecting blink. Specifically, the gaze tracking process consists of three steps: estimating the reflected light under a center pupil, extracting features on the reflected light change associated with pupil position,

<sup>3</sup>A short calibration is common for all existing eye trackers and eye-tracking designs.

and running a boosted tree regression algorithm to estimate a 3D gaze vector. The blink detection process examines the reflected light changes and leverages the spatial correlation among changes perceived by photodiodes to detect blink. Since a blink typically lasts 300 to 400 milliseconds [1], LiGaze runs the blink detection algorithm every 0.3 seconds. Next we describe the detailed design of gaze tracking and blink detection.

## 4 LIGAZE DESIGN

Based on the real-time photodiode data, LiGaze infers 3D gaze vectors represented as normalized 3D vectors. Next we first introduce our estimation of reflected screen light under a center pupil. We then describe our learning algorithm for gaze inference and the mechanism for blink detection.

### 4.1 Estimating Reflected Screen Light

Given  $N$  photodiodes on each side of the annulus sensing panel, we seek to model the relationship between screen light sensed by  $N$  front photodiodes (facing the screen) and the reflected screen light sensed by  $N$  back photodiodes (facing the eye), given an fixed pupil location (center). This relationship characterizes the impact of the reflections occurred after screen light rays strike the eyeball.

We start our exploration with experiments. We attach the annulus PCB in Figure 11(a) to a wearable VR headset, instruct a user to wear the headset, and ask the user to stare at screen center. We display 257 images on the screen to generate nonuniform, time-varying screen light and collect data from all photodiodes. For every pair of front and back photodiodes, we compute the Pearson correlation coefficient [15], which is the standard measure of the linear correlation between two variables, with  $r$  ( $-1 \leq r \leq 1$ ) indicating the strength and direction of the correlation, and  $p$  indicating the significance of the finding.

Table 3 lists the correlation coefficients ( $r$ -values,  $p < 0.01$ ) for all pairs. Interestingly, a strong linear correlation commonly exists between back photodiode in direction  $d$  and front photodiode in direction  $d'$  that is diagonally opposite to  $d$ . Front photodiodes in directions other than  $d'$  are also correlated with back sensor in direction  $d$ , and the correlation becomes weaker for front photodiodes further away from direction  $d'$ . Take the top-right (TR) back photodiode as an example, its perceived reflected light has the strongest linear correlation ( $r = 0.91$ ) with that at the bottom-left (BL) front photodiode. For front photodiodes further away from BL, the correlation gradually decreases. This observation suggests that reflected light intensity is dominated by specular reflections (possibly due to eyeball’s glossy surface), which direct each incoming screen light ray to a coplanar direction diagonally opposite to its incident direction. Diffuse reflections, on the other hand, scatter a light ray in a broad range of directions. Although they also contribute to the reflected light intensity perceived by a back photodiode, diffused light rays are much weaker and thus play a much less significant role.

Given the linearity of light transport [50] (i.e., contributions of individual light rays add up at a receiver location), we consider to model the relationship between back and front photodiodes’ readings as a simple linear relation. Similar linear models have also been applied in prior works [18, 50, 59] in computer graphics and vision. Specifically, let  $\mathcal{I}$  be the vector of  $N$  front photodiodes’

		Front sensor (facing display)							
		T	TR	R	BR	B	BL	L	TL
Back sensor (facing eye)	T	0.48	0.75	0.87	0.87	0.96	0.81	0.59	0.54
	TR	0.69	0.55	0.76	0.76	0.79	0.91	0.88	0.78
	R	0.66	0.64	0.55	0.67	0.79	0.83	0.93	0.89
	BR	0.86	0.79	0.62	0.51	0.64	0.74	0.76	0.84
	B	0.97	0.83	0.68	0.55	0.50	0.51	0.78	0.79
	BL	0.84	0.90	0.89	0.86	0.85	0.61	0.60	0.58
	L	0.80	0.91	0.92	0.78	0.65	0.49	0.52	0.58
	TL	0.66	0.88	0.89	0.91	0.86	0.78	0.52	0.47

**Table 3: Pearson correlation coefficient ( $r$ -value) matrix ( $p < 0.01$ ) between the reflected light and display light intensity around the eye.  $T, R, B$ , and  $L$  denote top, right, bottom, and left direction, respectively.**

readings. Assuming a center pupil, the estimated back photodiodes' readings, denoted as vector  $\tilde{\mathcal{R}}$ , can be written as:

$$\tilde{\mathcal{R}} = \mathbb{W} \cdot \mathcal{I}, \quad (1)$$

where  $\mathbb{W}$  is a  $N \times N$  weight matrix:  $\mathbb{W} = \{w_{ij} | i, j \in [1, N]\}$ .  $w_{ij}$  indicates the contribution of incoming light intensity  $I_j$  perceived by front photodiode  $j$  to the estimated reflected light  $\tilde{R}_i$  perceived by back photodiode  $i$ .

**Calibration.** The weight matrix  $\mathbb{W}$  can differ across users based on their eye and skin characteristics (Figure 7). Subtle offset of the headset to the eyes can also affect screen light's reflections and thus  $\mathbb{W}$ . To deal with user diversity and headset offsets, we determine  $\mathbb{W}$  in a calibration phase when a user first puts on the VR headset. During the calibration, the user watches a short video while staring at the screen center. In the meantime, the readings of all front and back photodiodes are then used to calculate  $\mathbb{W}$  using linear regression. Specifically, let  $\mathcal{R}$  be the vector of measured back photodiodes' readings, we are seeking  $\mathbb{W}$  that leads to the estimated vector best matching the measurement  $\mathcal{R}$ :

$$\mathbb{W} = \underset{\mathbb{W}}{\operatorname{argmin}} \|\mathcal{R} - \tilde{\mathcal{R}}(\mathbb{W}')\|^2. \quad (2)$$

With the calibrated  $\mathbb{W}$ , upon real-time readings  $\mathcal{I}$  from front photodiodes, we can then estimate on the fly reflected light at each back photodiode using Eq. (1), assuming pupil is in the center.

To shorten the calibration, we judiciously design the video content so that it best represents the variety of brightness distribution of the screen using a limited number of frames. Specifically, we divide each video frame into  $10 \times 10$  grids and randomly change the gray-scale color of each grid based on a quasi-random sampling strategy in the color space [65]. Quasi-random sampling can generate samples in high-dimensional space more uniformly than the traditional pseudo-random sampling [45]. This strategy allows us to use a  $2\text{-min}^4$  video to estimate  $\mathbb{W}$  accurately.

## 4.2 Gaze Inference

The next step of LiGaze is to extract features related to pupil position and infer a 3D gaze vector. Given incoming screen light, our model in § 4.1 estimates the reflected light that each back photodiode would perceive if pupil is in the center. Thus, the difference between this estimate and the measured reflected light is caused by the offset

between the actual pupil position and the center. Specifically, we consider normalized change, since the actual amount of change in reflected light can depend on the level of reflected light intensity. Hence, let  $\mathcal{R} = \{R_1, \dots, R_N\}$  denote the vector of measured reflected light at  $N$  back photodiodes, we compute  $f_i = R_i/\tilde{R}_i$  for each photodiode, and use the resulting set  $\mathcal{F} = \{f_1, \dots, f_N\}$  as the feature vector for later gaze inference.

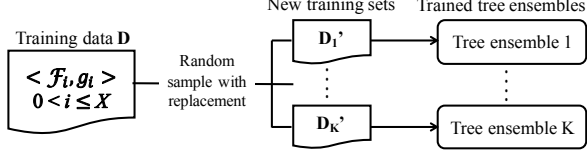
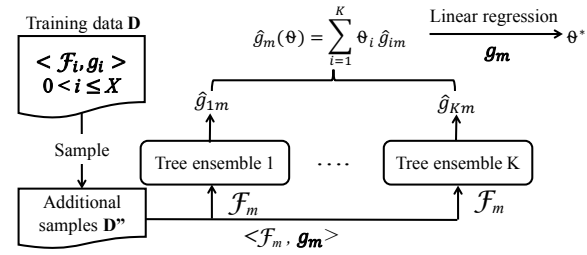
We infer gaze using supervised learning. In particular, we choose boosted trees (or tree ensemble) [12, 21], which represent the relationship between features and prediction values as a set of regression trees. Here each tree is similar to a decision tree but differs in that it associates a weight to each leaf (i.e., decision). By summing the predictions of multiple trees, boosted trees improve the reliability and accuracy of the final prediction. Another important benefit is its lightweight computation. Once the tree ensemble is trained, online regression involves only comparison operations. Thus it is fast and entails low power, which is critical for real-time execution on micro-controllers. We have also explored other options such as Support Vector Machine (SVM) [58] and its variants (SVC1V1 [27], SVOREX [14], REDSVM [38]), and feed-forward neural network [9]. However, these algorithms entail either higher latencies or larger memory footprints. For example, neural network involves floating-point addition, multiplication, and exponential operations; SVM and its variants require storing a large number of support vectors (e.g., 10K vectors for a training set with 200K samples).

**Offline Training.** We train boosted trees offline using a data set consisting of computed feature vectors and 3D gaze vectors from existing eye trackers. Let  $D$  denote the training set with size  $X$ , where  $D$  consists of  $X$  pairs of feature vector and actual 3D gaze vector  $g$ , i.e.,  $D = \{\langle \mathcal{F}_i, g_i \rangle | 0 < i \leq X\}$ . To improve the stability and accuracy of our inference, we combine the power of multiple tree ensembles by applying bootstrap aggregating (i.e., bagging) [10]. As illustrated in Figure 9, it has two steps. *First*,  $K$  tree ensembles are trained using  $K$  new training sets generated from the original training set  $D$ . Each new training set  $D'_k$  (with size  $X'$ ) is created by uniformly sampling  $D$  with replacement. Each new training set is then used to train a tree ensemble separately. *Second*, the resulting  $K$  tree ensembles are combined linearly. Additional training samples are used to seek the optimal linear combination using linear regression. Specifically, for each training sample  $\langle \mathcal{F}, g \rangle$ , we feed it to each tree ensemble and obtain the prediction value. Let  $\hat{g}_j$  be the prediction from the  $j$ -th tree ensemble, then the final prediction  $\hat{g}$  is derived as  $\hat{g}(\theta) = \sum_{j=1}^K \theta_j \cdot \hat{g}_j$ , where  $\theta_j$  is the weight of the  $j$ -th tree ensemble. We then run linear regression to determine the optimal  $\theta^*$ , such that  $\theta^* = \operatorname{argmin}_{\theta} \|\hat{g}(\theta) - g\|$ . By combining multiple tree ensembles, bootstrap aggregating seeks to learn underlying correlations that could differ across user groups.

**Online Inference.** With the trained tree ensembles, LiGaze's online inference operates with only photodiode data coming on the fly. Since individual user can have his/her own features and differ from users in the training data, to better address user diversity, for each user during online testing, we calibrate  $\theta^*$  through a short (1-minute in our experiments) calibration phase (after the phase of calibrating  $\mathbb{W}$ ). It is a simple game where the user is prompted to stare at various dots with known locations spread in a virtual

<sup>4</sup>Our experiments show that after 2 minutes, user's gaze is more likely to start drifting away from the screen center.



Step 1: train  $K$  tree ensemblesStep 2: seek optimal linear combination of  $K$  tree ensembles

**Figure 9: Applying bootstrap aggregating over boosted trees, where  $K$  tree ensembles are trained using training sets generated by random sampling with replacements, and their optimal linear combination is derived by linear regression with additional training samples.**

scene. We then feed the front and back photodiodes' readings to the  $K$  trained tree ensembles to obtain predicted  $\hat{g}$ . Assuming the dots are the actual locations the user stares at, we calculate gaze vectors  $g$  based on actual dot locations and treat them as ground truth. Using linear regression, we compute the optimal adjustment  $\epsilon$  over existing  $\theta^*$ , so that  $(\theta^* + \epsilon)$  best matches this user's ground truth. Specifically,  $\epsilon$  is calculated as:

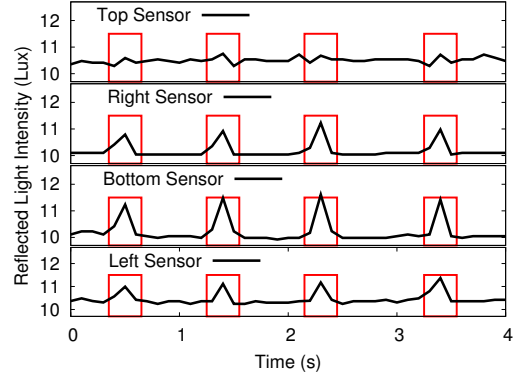
$$\epsilon = \underset{\epsilon'}{\operatorname{argmin}} (||\hat{g}(\theta^* + \epsilon') - g||^2 + \lambda ||\epsilon'||^2), \quad (3)$$

where  $\lambda$  is a regularization term to constrain the amount of adjustment  $\epsilon$ . We determine  $\lambda$  using cross-validation among users in the training data set. We consider slight adjustment over  $\theta^*$ , rather than calculating a new  $\theta^*$ , because of the small amount of data from the short calibration phase. Overall, this short calibration allows the system to slightly adjust  $\theta^*$  to best fit a specific user.

After the quick calibration, upon each set of back and front photodiode readings coming on the fly, we infer the 3D gaze vector as follows: 1) with front sensors' readings  $\mathcal{I} = \{I_1, \dots, I_N\}$ , we estimate  $\tilde{R}_i$ , the reflected light under a center pupil for each back photodiode  $i$  based on Eq. (1); 2) based on back sensors' readings  $\mathcal{R} = \{R_1, \dots, R_N\}$ , we derive the feature vector  $\mathcal{F} = \{f_i\}$ , where  $f_i = R_i / \tilde{R}_i$ ; 3) we input  $\mathcal{F}$  to each of the  $K$  trained tree ensemble and obtain prediction  $\hat{g}_j$  from tree ensemble  $j$ ; 4) finally the gaze vector is inferred as  $\hat{g}(\theta^* + \epsilon) = \sum_{j=1}^K (\theta_j^* + \epsilon_j) \cdot \hat{g}_j$ .

### 4.3 Blink Detection

LiGaze detects a blink event based on a simple observation: when a blink occurs, the reflected light intensity around the eye experiences similar pulse changes in most directions. Figure 10 shows the time series of reflected light intensity perceived by four photodiodes around the eye. We observe that reflected light intensity in three directions exhibit similar pulse changes during a blink. When the eye is closed, the pupil is covered by the eyelid, which reflects light rays better than pupil. When the eye is opened, pupil continues



**Figure 10: Reflected light intensity changes when user blinks the eye. Each red box represents a blinking instance**

absorbing incoming light rays, which decreases the reflected light intensity around the eye. Since the eyelash can block some light rays between the eyeball and photodiodes, this pulse change is not significant at the top photodiode. However, a blink still leads to noticeable patterns in most photodiodes. Thus, by counting the number of photodiodes concurrently perceiving pulses of reflected light intensity, we can detect the occurrence of blink.

To detect the reflected light pulses around the eye, we first extract gaze features to remove the impact of screen content change, using the estimate of reflected screen light (§ 4.1). We then compute the first-order derivatives of time series of gaze features at each photodiode. A pulse of light intensity can be detected when two adjacent first-order derivatives are positive and negative, respectively. When a sufficient number (half of the photodiodes in our implementation) of photodiodes perceive pulses in their gaze features, LiGaze outputs a blink event. To further reduce the false positive due to subtle screen content change (e.g., cut scene in movie), LiGaze bypasses the blink detection when half of the front photodiodes concurrently detect a pulse in incoming screen light.

## 5 LIGAZE PROTOTYPE

We build a LiGaze prototype using off-the-shelf hardware and attach it to an existing VR headset (FOVE). The prototype contains three main components:

**Light-Sensing Unit.** We design and fabricate a thin (0.8 mm), ring-shaped PCB (Figure 11(a)) that hosts 16 photodiodes (OPT3001) on each side and their associated circuits (e.g., resistors, capacitors). The photodiode is 1.9 mm  $\times$  1.25 mm in size, so the PCB ring is narrow in width (2 mm), avoiding to affect user's view. The ring diameter is 36 mm, customized to fit the lens of the FOVE headset.

We select OPT3001 for three reasons. First, it is ultra-low power, consuming only 6  $\mu$ W in the active state. Also, with a built-in ADC, it directly outputs digitized light intensity (in lux, at 10-Hz rate), thus removing the need to add external ADC that can be energy-consuming. Second, as an ambient light sensor responding only to visible light (400 nm – 700 nm), it can sense very weak light (as low as 0.05 lux) and provides a high resolution (0.01 lux). Thus, it is suitable for sensing weak reflected light in our scenario. Third, it has a  $\pm 45^\circ$  field of vision (FOV). With only 1.8 cm to the eye, this FoV is sufficient for covering the whole eye surface after aggregating the coverage of 16 photodiodes.

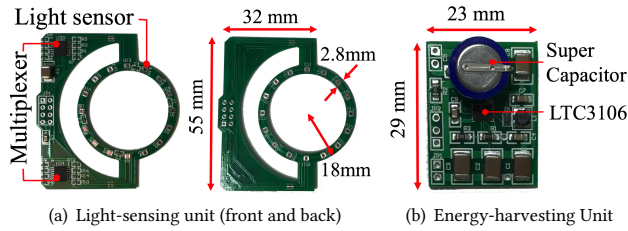


Figure 11: Fabricated PCBs in LiGaze prototype.

Photodiodes transmit raw data to a micro-controller through a hardware Inter-Integrated Circuit (I2C) interface at the clock of 400 kHz. Given that OPT3001 allows up to 4 serial bus addresses, one hardware I2C can only differentiate up to 4 photodiodes. To support fetching data from 32 (16 front and 16 back) photodiodes, we design a switch using two 4-channel analog multiplexers (74HC4052). The switch divides 32 photodiodes into 8 groups and lets each group transmit data to the micro-controller in turn. In our experiment, the switch consumes less than  $2 \mu\text{W}$ , and the hardware I2C interface is 1.5 times more energy-efficient than software simulation I2C interface for supporting the 32 sensors. The switch and its associated circuit are integrated in a side PCB board connecting to the PCB ring. This is different from the PCB used in our prior experiments (Figure 5(a)), which is a single-sided PCB ring hosting eight photodiodes. This new PCB design leads to a cleaner look with only 7 wires connecting to the micro-controller (Figure 1). The arc gap allows the rim of the VR lens to fit in.

**Micro-Controller.** The micro-controller periodically wakes up to receive data from photodiodes, computes the feature vector related to pupil position, and runs our gaze inference algorithm (§ 4.2) to derive the current 3D gaze vector. We choose MSP432 because of its ultra-low power according to the power benchmark ULP-Bench [5]. ULPBench is from the embedded micro-processor benchmark consortium (EEMBC) and is a standard way to compare power performance on micro-controllers. The gaze inference results can be either stored on the micro-controller or transmitted to other computing units through a USB cable.

**Energy-Harvesting Unit.** To demonstrate LiGaze’s ultra-low power, we add a credit-card sized solar cell (AM-1816) atop the headset, which harvests energy from indoor lighting to power the entire system (sensing and computation). AM-1816 is an amorphous silicon solar cell and is sensitive only to visible light (wavelengths from 400 nm to 700 nm). Therefore, it is ideal for harvesting indoor light. We use a buck-boost DC/DC converter (LTC3016) to maintain the output voltage at 4V, which maximizes the output power at varying light conditions (e.g., 200 lux - 800 lux).

The main challenge in designing this unit is to maintain an output power above the minimum required by the system. Since the solar cell is atop the headset, user’s head movement can cause drastic variations in light intensity perceived by the solar cell, resulting into drastic changes in output power ( $200 \mu\text{W}$  to  $1200 \mu\text{W}$ ). Such power variation can make the system out of power occasionally. To address this problem, we leverage a 15-F super capacitor to store extra harvested energy (Figure 11(b)). The capacitor can be fully charged after 12 hours under 600-lux lighting. Once fully charged, it can support our system for 18 hours without harvesting additional

# of Users	Eye Color			Skin Color		
	Black	Blue	Green	White	Yellow	Black
	18	8	4	15	9	6

Table 4: Participant information in evaluation study, where participants differ in eye and skin characteristics.

energy. It allows the system to run in a dark environment and to deal with sudden light changes due to head movement.

## 6 LIGAZE EXPERIMENTS

To evaluate LiGaze prototype, we recruit 30 participants (from 18 to 45 years old, 19 males and 11 females). The user information is summarized in Table 4. Three out of 30 participants wear contact lens. We first focus on gaze tracking performance across participants and dynamic VR content. We then measure LiGaze’s energy consumption and analyze the tradeoff between energy and performance. Finally, we consider several practical factors when using LiGaze, including head movement and user perception. We have made a demo video of LiGaze available at [3].

**Experimental Setup.** We add LiGaze to an existing VR headset (FOVE 0, \$600) equipped with an eye-tracking system, which provides 3D gaze vectors at 120 Hz with a claimed accuracy of  $1^\circ$  error. We run our system concurrently with FOVE and use FOVE’s output as ground truth to evaluate LiGaze. We choose FOVE because it is the first VR headset in the market (released in Feb. 2017) with a built-in eye tracker. FOVE uses near-infrared emitters and infrared cameras in the headset to track gaze. Since our light sensors only respond to visible light (400–700 nm) spectrum, FOVE’s near-infrared emitters have a negligible impact on LiGaze.

In our experiment, each participant wears the FOVE headset and watches various VR content. The resulting light intensity at VR lenses is measured as 34 lux on average, with the 5-th percentile at 20 lux and 95-th percentile at 75 lux (Figure 12). The experiment starts with two quick calibrations: a 2-min calibration described in § 4.1 to compute  $\mathbb{W}$  in Eq. (2), and a 1-min game to calibrate parameters in Eq. (3). After calibration, the participant watches 30-min VR content<sup>5</sup>, including drama, sport, scenery, and games. The VR content is randomly selected from a pool for each participant.

To evaluate LiGaze gaze tracking performance, we examine four metrics: 1) *accuracy*: the 3D angular difference between 3D gaze vectors inferred by LiGaze and FOVE; 2) *precision*: a measure of stability and reliability of gaze inferences, defined as the variance of the temporally-successive point of gazes [26]; 3) *latency*: the duration that an online gaze inference takes; and 4) *power consumption*: the overall power consumption of LiGaze.

### 6.1 Gaze Tracking Accuracy

To examine gaze tracking accuracy, we consider two scenarios: 1) *within-user accuracy*, where we use the same participant’s data to train and test the inference model (25% for training and 75% for testing); 2) *cross-user accuracy*, where we train the model using half of participants’ data and test it using the other participants’ data. Specifically, we divide participants into two groups based on

<sup>5</sup>Participants only watch VR content because no existing VR applications in FOVE support VR controllers yet.



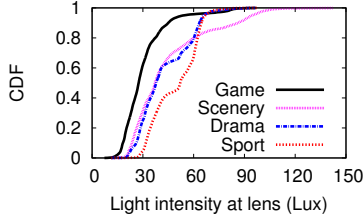
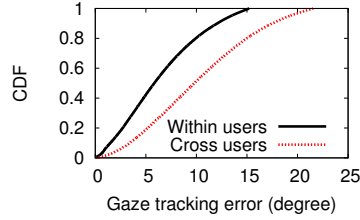
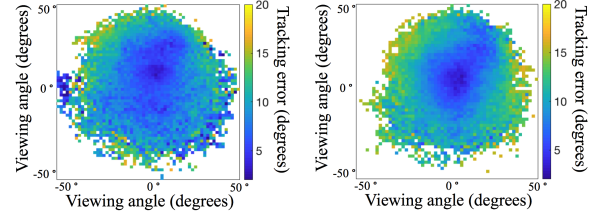


Figure 12: Light intensity at VR lens under different VR content types.



(a) CDF of gaze tracking error



(b) Error distribution (within user)

(c) Error distribution (cross user)

Figure 13: Overall gaze tracking accuracy in comparison to FOVE.

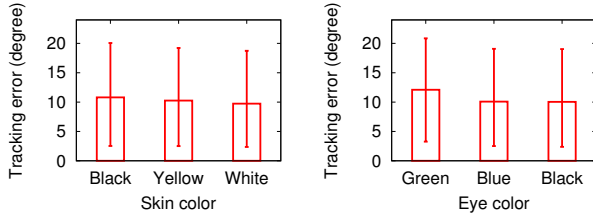


Figure 14: Impact of skin and eye color on LiGaze accuracy.

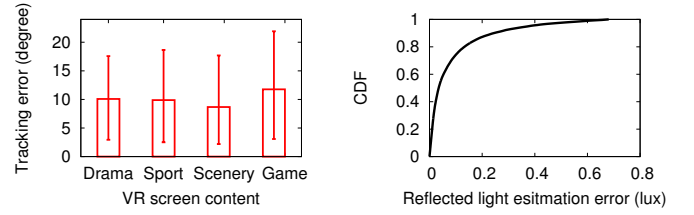


Figure 15: Impact of VR screen content type. Figure 16: Accuracy in estimating reflected light

their skin colors (light and dark). We train and test inference model separately for each group of participants. Note that within-user accuracy only serves as a reference to examine LiGaze’s performance if the same user’s training data is available (e.g., through click-event based collection of a user’s gaze data using methods similar to [28, 51]). LiGaze does not require to train each new user, rather, it builds an inference model offline using pre-collected training data.

**Overall Accuracy.** Figure 13(a) plots the CDFs of within-user and across-user accuracy, over 418,906 gaze inference instances. The mean error is  $6.3^\circ$  and  $10.1^\circ$  with 95-th percentile at  $13.4^\circ$  and  $19.2^\circ$ , for within- and cross-user accuracy, respectively. We further examine the spatial pattern of tracking accuracy and plot the error distribution in Figure 13(b) and 13(c). We observe that tracking error in the center region is smaller (within  $5^\circ$ ), while higher tracking errors commonly occur at the boundary of the eye’s field of view (FoV). It can be attributed to two factors. First, 40% of our training data contain samples with gazes in the center region. With fewer training samples in border regions, gaze inferences in those regions are less accurate. We plan to further enrich our training set to improve the performance. Second, a recent study [20] reports larger tracking errors in screen corners using two commercial IR-based eye trackers (Tobii EyeX and SMI REDn). Since FOVE uses a similar methodology (IR-based), it can suffer from similar problems, causing larger deviation between FOVE and LiGaze inferences. We plan to test higher-end eye tracker in the future to examine this problem.

**User Diversity.** Focusing on cross-user accuracy, we examine the impact of user diversity on gaze inference accuracy. Figure 14 plots the mean accuracy for participants with different skin and eye colors. We also include error bars covering 90% confidence intervals. We observe that the average accuracy is similar among different skin colors, as well as eye colors (blue and black). The accuracy of the green-eye participant is lower because only four participants have green eyes. However, the within-user accuracy for these participant is  $7.4^\circ$  on average, indicating that the performance can be greatly improved if there are other participants with green eyes in the training set. We conclude that it is necessary to have representative eye colors in the training set for robust inference.

**VR Content.** Next, we examine the impact of VR screen content on LiGaze’s gaze tracking accuracy. We partition VR screen content into four types: drama (quick scene changes), sports (stable background with moving foreground objects), scenery (small, gradual scene change), and games (drastic scene changes). Figure 15 plots the cross-user accuracy under each content type. We observe that the accuracy under games is slightly lower. The reason is twofold. First, we observe that to highlight target objects, game content is generally darker than other content types (Figure 12). The lower screen light intensity leads to smaller changes in reflected light at photodiodes and thus introduces higher errors in gaze inference. Second, users often rapidly move their gaze directions during games. Since it takes 3.6 ms to collect both reflected light and screen light intensity from all photodiodes, the gaze movement within the data collection data can affect the tracking performance. Due to hardware limitation, the I2C interface’s baud rate is limited by its full-speed mode (400 Kbps). We can shorten data collection duration by using the faster modes such as the fast mode (1 Mbps) or even high-speed mode (3 Mbps). We leave it as future work.

**Accuracy in Estimating Reflected Light.** Finally, we examine the accuracy of our linear model in § 4.1 in predicting reflected screen light under a center pupil, as it is the basis of LiGaze’s gaze inference. For each participant, the weight matrix  $\mathbb{W}$  is computed using the short (2-min) calibration data. We then compute estimation error as the difference between the estimated and measured reflected light intensity at back photodiodes. Figure 16 plots the CDF of reflected light estimation error for all 30 participants. Overall, the estimation is fairly accurate: the mean estimation error is 0.09 lux with 0.38 lux as the 95th percentile. Given that the reflected light change is between -1 lux to 2 lux (Figure 6), our model is sufficient for deriving features related to pupil position.

## 6.2 Gaze Tracking Precision and Latency

**Precision.** We also measure the precision of LiGaze’s gaze tracking, which reflects the reproducibility or stability of gaze inferences in spatial domain. It is calculated as the root mean square (RMS) of

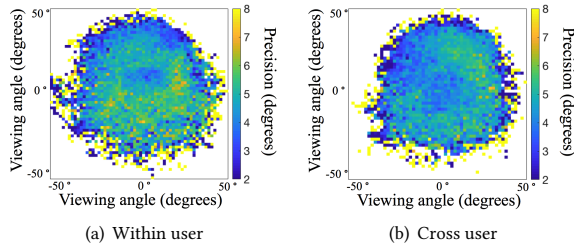


Figure 17: Precision distribution of LiGaze gaze tracking.

successive inference results under a fixed gaze [26]. We compute this metric using the gaze inference results during the 5-min calibration, where users stare at screen center. As shown in Figure 17, the mean precision is  $5.2^\circ$  and  $5.9^\circ$  using the within-user and cross-user model, respectively. Similar to the accuracy distribution, the precision is larger at the boundary of the eye’s FoV.

**Latency.** Table 5 shows the latency of LiGaze’s main steps. The sensing step includes photodiode sensing duration and the data communication between photodiode and MCU through the I2C board. The feature extraction is mainly on estimating reflected light, with tens of floating-point multiplications. The online gaze inference only includes hundreds of floating point addition and tens of floating point multiplications, thanks to the simplicity of the boosted trees regression. Blink detection entails computing the first-order derivatives of adjacent data points and thus the computation overhead is negligible. Overall, LiGaze generates an inferred gaze vector within 7.8 ms on average, holding the potential to support 128-Hz tracking if photodiodes support higher ADC sampling rates.

Steps	Light Sensing	Feature Extraction	Gaze Inference	Blink Detection	Total
Duration (ms)	3.6 ( $\pm 0.1$ )	1.5 ( $\pm 0.15$ )	2.6 ( $\pm 0.15$ )	0.1 ( $\pm 0.01$ )	7.8 ( $\pm 0.4$ )

Table 5: LiGaze latency.

### 6.3 Power Consumption

We measure LiGaze’s power consumption using Monsoon power monitor. Table 6 lists the power consumed by the main components of LiGaze. Overall the system (sensing and computation) consumes  $791 \mu\text{W}$  on average. The MCU consumes 75% of the total power at 10-Hz tracking rate, where it is active for only 78 ms every second (Table 5). 32 photodiodes consume  $226 \mu\text{W}$  because they not only sense the ambient light intensity, but also digitize the light intensity using built-in ADC. The data collection relies on the hardware I2C interface for the MCU to communicate with photodiodes. Its power consumption can be further lowered down by leveraging DMA. When the MCU does not infer gaze movement, it enters an ultra power mode (LPM3) with only a few lower-power clocks awake.

Next, we evaluate the efficacy of our energy-harvesting unit, seeking to examine whether the whole gaze tracking system can be completely powered by the energy harvested from indoor lighting. We measure the harvested power and the power loss using our energy-harvesting unit under three levels of indoor lighting. Table 7 lists the results and energy conversion ratios. We observe that once the ambient light is above 400 lux, LiGaze can be powered without additional power. The additional harvested energy in the brighter environments can be stored in the super capacitor so that the system can continue tracking gaze movement under temporally

dark scenarios caused by user movement and ambient light change. The energy conversion ratio for the energy harvest component remains 86–89% regardless of the ambient light level.

Finally, to understand the tradeoffs between energy and tracking accuracy, we compare LiGaze to three existing low-power gaze tracking systems: iGaze [74], iShadow [40], and CIDER [41]. Since all these systems run in different tracking rates, we examine the energy (in  $\mu\text{J}$ ) consumed per gaze inference. For LiGaze, we also down-sample photodiodes to examine its performance and energy with fewer photodiodes (e.g., 8, 16). Figure 18 plots the comparison, where the black line indicates the potential trend of LiGaze in striking the balance between energy and tracking accuracy. We observe that LiGaze reduces the energy per inference by multiple orders of magnitude, with sacrifices in tracking accuracy. However, with more photodiodes, LiGaze can capture more feature dimensions in boosted trees, which can potentially improve the tracking accuracy. The trend (black line) indicates that with more photodiodes, LiGaze can achieve accuracy comparable to CIDER and iShadow, yet consuming almost half the power. By using photodiodes to capture low-level light signals, LiGaze can adapt the number of photodiodes to specific application needs, making it flexible and efficient. Whereas other systems all rely on cameras capturing hundreds to millions of pixels, where many pixels are often redundant.

### 6.4 Blink Detection

To examine LiGaze’s accuracy in blink detection, we compute the precision and recall for all participants, where precision is the percentage of correctly identified blink events among all blink events identified by LiGaze, while recall is the percentage of actual blink events that are identified by LiGaze. Table 8 summarizes the results, where 6511 blink instances were collected from 30 participants during the study. The average interval between adjacent blinks is 9.7 seconds across participants, and each blink lasts 0.3 – 0.4 seconds. Because of photodiodes’ limited ADC sampling rates (10 Hz), LiGaze can only capture up to 4 data points during a blink and it may miss the data point when a user just closes the eyes. We also observe that the blink detection is more accurate for users with lighter skin, and the recall for users with black skin is low. This is because our blink detection algorithm assumes eyelid reflects light well, whereas the reflectivity of black skin is low, leading to more missed blink events. In the future, we plan to test photodiodes with higher ADC sampling rates, which can allow LiGaze to capture more data points within a blink instance and extract more detailed temporal features to improve detection accuracy.

### 6.5 Practical Considerations

**Head Movement.** To examine the impact of head movement on LiGaze, we ask a participant to wear the VR headset and watch the same VR content for two rounds. In the first round, the participant keeps the head still, while in the second time, the participant freely turns the head. As we examine the gaze tracking accuracy in these two rounds, we observe that the head movement has negligible impact on the average tracking accuracy, where the difference is within  $0.8^\circ$ . It is because unlike other gaze tracking scenarios, in VR, the headset keeps fixed relative positions among the eyes, photodiodes, and the VR display, thanks to the head strap. Thus,

Components	Mean Power ( $\mu\text{W}$ )
32 photodiodes	226 ( $\pm 20$ )
Data collection (MCU)	207 ( $\pm 15$ )
Feature extraction (MCU)	120 ( $\pm 12$ )
Gaze inference (MCU)	193 ( $\pm 10$ )
Blink detection (MCU)	5 ( $\pm 0.5$ )
Standby (MCU)	40 ( $\pm 10$ )
Total	791 ( $\pm 67$ )

Table 6: LiGaze energy consumption.

Ambient light (Lux)	Harvested power ( $\mu\text{W}$ )	Power loss ( $\mu\text{W}$ )	Conversion ratio (%)
600	1091	163	87
500	924	114	89
400	788	107	88
300	589	96	86

Table 7: The performance of LiGaze energy-harvesting unit.

	Skin color		
	White	Yellow	Black
Precision	0.86	0.92	0.67
Recall	0.81	0.80	0.76
Overall Precision	0.83		
Overall Recall	0.80		

Table 8: LiGaze blink detection performance.

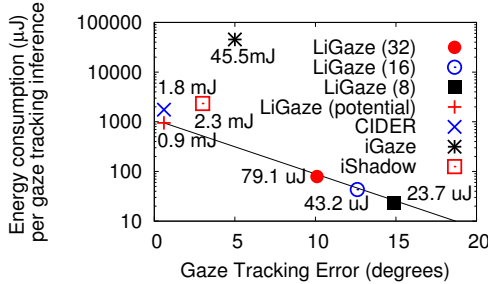


Figure 18: Tradeoffs between energy and tracking accuracy for LiGaze and existing low-power gaze tracking systems. We down-sample photodiodes in LiGaze to evaluate the trend (the black line) in balancing energy consumption and tracking accuracy.

the light propagation properties within the headset remains the same regardless of the user head movement, making the system robust against head movement.

**User Perception.** Since LiGaze adds an annulus PCB on top of each VR lens, we seek to understand whether the PCB affects user’s viewing of VR content. In the experiment, we ask sixteen participants to wear the headset with and without our PCB board for 10 minutes. We then ask them to rate their perception of VR content from 1 to 5, where 1 means the PCB board significantly affects the viewing of VR content, and 5 means the user cannot feel the PCB board at all while watching the VR content. All users in the study rate LiGaze as 5, demonstrating that LiGaze does not affect user’s VR experiences.

**Supported Applications.** We also examine the implications of LiGaze’s gaze tracking accuracy to end users in practical applications. We consider using LiGaze to differentiate a few regions in the screen, which can enable simple user input (e.g., selecting/clicking buttons, typing letters or numbers) using only glances. We test three settings, where we split the eye’s FoV into  $2 \times 2$ ,  $3 \times 3$ , and  $4 \times 4$  grids, respectively. In each setting, we select the testing data with actual gaze directions (based on FOVE) within a circular center ( $3^\circ$  range<sup>6</sup>) of each grid. We then check whether the corresponding gaze directions inferred by LiGaze are also within the same grid’s circular center. We observe that LiGaze accurately differentiates different regions, achieving 100%, 99.5%, and 91.7% accuracy under 4, 9, and 16 regions, respectively. It demonstrates that LiGaze can be utilized in a wide range of interaction applications such as dialing phone numbers, navigating VR world, selecting menu items, browsing photo gallery, and controlling the depth of field effect rendering in VR world. We further implement a few demo applications as a proof of concept based on the above scenarios. We conduct a user

<sup>6</sup> $3^\circ$  is the typical range of gaze drift based on our data collected using FOVE.

study with 10 users to gather user feedback on using LiGaze for accomplishing these tasks. Users are asked to rate their satisfaction on both accuracy and latency of LiGaze from 1 to 5, where 5 means high satisfaction. On average, users have responded with 4.2 and 4 on accuracy and latency, respectively. It indicates that LiGaze’s performance is sufficient for these basic interaction applications.

## 7 DISCUSSIONS AND FUTURE WORK

We discuss the insights and lessons we learned through the study, current limitations, and open questions for future research.

**Data Quality of Ground Truth.** Our extensive experiments with FOVE have revealed issues of its performance stability. Participants have occasionally reported noticeable inference errors of FOVE towards the end of the study. FOVE’s practical performance deviates from the numbers (e.g.,  $1^\circ$  mean error) claimed by the manufacturer. This echoes the findings in a latest study [20], which reports similar results on commercial eye trackers’ performance in the wild. Additionally, we observe that some participants exhibit fatigue in the end of the study, resulting into tears in the eye or long closing of eyelid. Both factors affect the quality of ground truth data, the training of our gaze inference algorithm, and evaluation results. In the future, we plan to test other commercial eye trackers and examine whether they provide better-quality tracking results as ground truth. We will also refine the design of our study to reduce the likelihood of user fatigue. Potential solutions include partitioning a study into multiple short (e.g., 15-minute) sessions, and adding more interactive VR games to make the study more entertaining.

**Improving Tracking Accuracy.** LiGaze’s current tracking accuracy is constrained by two factors. First, the training data set is still relatively small. Moving forward, we will recruit more diverse users (especially those with green or hazel eyes) to train our boosted-tree model and improve tracking accuracy among different user groups. Second, limited by the photodiode’s built-in ADC, our current prototype samples sensor data at 10 Hz. This low sampling rate causes the system missing very rapid eye movements (e.g., saccades lasting only tens of milliseconds). Also, it affects the accuracy of blink detection, as some fast eye movements (e.g., closing eye lid) during a blink are occasionally missed, which triggers more false negatives. This problem, however, is not a fundamental limit of our approach. It can be solved by using photodiodes with ultra low-power ADCs supporting higher sampling rates. For example, ADS7042 from Texas Instrument consumes less than  $1 \mu\text{W}$  at 1 KSPS. Higher sensor sampling rate allows LiGaze to capture more subtle gaze movement and extract finer-grained time-series changes to improve tracking accuracy.



**Prior Knowledge of VR Content.** As a stand-alone system, LiGaze currently assumes no knowledge and control of screen content. However, such knowledge can be available if our system is integrated into the VR system, where the gaze inference algorithm and VR content rendering are both running at the HMD or server. In this scenario, we can remove front photodiodes (for sensing incoming screen light), and instead directly model the relationship between an image frame and the reflected light perceived by back photodiodes. We can then leverage this relationship to identify features related to pupil movement. This approach can lead to higher tracking accuracy, as front photodiodes only sample incoming screen light while an image frame allows us to estimate a full spatial distribution of screen light. We plan it as a future exploration.

**Scenarios Beyond VR.** The principle of our approach can be extended to broader application scenarios, such as augmented reality (AR) headsets or even general eyeglasses. The key challenge lies in the diversity of ambient light sources, which leads to light rays coming in more diverse directions when striking the eyeball. To sense ambient light in diverse directions, we will explore optimizations of photodiode placement along the eyeglass rim. For AR headset with front-facing cameras, we can leverage cameras to sense high-resolution information about incoming light. Photodiodes can complement cameras by sensing light outside camera's view. Fusing the information can better sense incoming ambient light and identify changes related to pupil movement.

## 8 RELATED WORK

We categorize existing gaze/eye tracking into two types and overview representative works in each category. We refer readers to surveys [24, 43] for more comprehensive summaries.

**Camera/Vision Based Methods.** Most gaze tracking systems rely on cameras. A common feature is limbus, the boundary between sclera and iris. The color contrast of these two regions makes it a feature easy to track horizontally, but not vertically. [32] used edge detection to segment limbus. Identifying pupil is hard because of the lower color contrast between pupil and iris.

To enhance the color contrast between pupil and iris, existing methods often use an IR light (880 nm) to illuminate eyes. By placing the camera either near or off light source's optical axis, camera sees a bright or dark pupil, which can be used to locate pupil [19, 42]. Furthermore, [49, 75] applied ellipse fitting to track iris and compute pupil center. [7] used artificial neural network to study implicit features for gaze tracking. Additionally, pupil-corneal reflection property have been used by examining the Purkinje images [17, 35], which are reflections created at different layers of the eye.

In addition to IR-based approaches, prior efforts examined using camera images under visible light to track gaze. They have studied gray-scale images [33], webcam images [28, 51, 56, 60], and depth images from Kinect [71]. The tracking methods include analyzing limbus features [34, 36, 37], inferring flow-field information [33], identifying iris center [6, 53, 62, 67], and tracking gaze locking [64].

Prior works have also studied gaze tracking when cameras are remote [29, 68, 73]. They focused on improving the tracking robustness under lighting condition changes [76], head motion [8, 25, 46–48, 77], less per-user calibration [61, 72], long range [13], and various facial features (e.g., eye corners, face color) [16, 44, 57, 63].

With the rise of wearable/mobile devices, recent research started to expand eye tracking to these low-power devices. Commercial eye trackers (e.g., Tobii, SMI), however, remain very expensive and consume high power (with battery packs). [30, 39] studied lower-cost alternatives with consumer cameras and open-source computer vision software. [74] added eye tracking to glasses to infer user attention and guide the design of networking protocols. Its eye tracking achieved  $5^\circ$  mean error with 2.5-W power. [40, 41] examined the performance tradeoff and proposed low-power designs by sampling image pixels. The latest prototype achieves  $1^\circ$  accuracy while consuming 7 mW at 10-Hz tracking rate. A recent work [22] also studied eye tracking for mobile VR (Google Cardboard). Reusing smartphone display and camera, they exploited the corneal reflection of screen to estimate user's gaze.

Comparing to all above works, our work differs in that it removes the need of cameras and uses only a few photodiodes, further lowering the power consumption. [4, 54] also studied camera-free methods but with a constant IR light. In contrast, our method is completely passive and deals with screen light dynamics. [66] inferred gaze by adding a photodiode to each LCD pixel to detect light reflected by eyes. It requires modifying the LCD display to embed the dense photodiode array, whereas our method does not require modifying the VR hardware.

**Non-Vision Based Methods.** Other methods use magnetic field or electric skin potential to estimate gaze. In [55], Robinson used a contact lens embedded with a small coil to track user's gaze. By measuring the voltage induced in the coil by an external magnetic field, it estimated user gaze. A recent study [70] designed a wearable scleral coil tracking system for VR, where scleral search coils are mounted on a VR headset to create a magnetic field. Electrooculogram (EOG) places skin electrodes around the eye to measure differences in skin potential caused by eye movement [31]. In [11], Bulling et. al built wearable EOG goggles to detect blink, saccade, and eye movement in four directions. Overall, these methods are very accurate ( $< 1^\circ$ ) and yet the most intrusive. Our work differs in that our system is much less intrusive and entails a much simpler set up with only a few small photodiodes on VR lens boundary.

## 9 CONCLUSION

We demonstrated the feasibility of using only a few low-cost photodiodes to sense reflected screen light for inferring user's gaze in VR. We designed and fabricated a dual-sided light-sensing unit hosting photodiodes to sense incoming and reflected screen light. Exploiting pupil's light absorption property, we designed lightweight learning algorithm to extract features related to pupil movement and infer gaze on the fly. We validated our design using prototype experiments. Moving forward, we plan to expand our training data with larger-scale user studies and improve system performance for diverse users.

## ACKNOWLEDGMENT

We sincerely thank the reviewers for their insightful comments and Emmanuel Akosah '19 for helping in organizing the user study and developing demo applications. This work is in part supported by National Science Foundation under CNS-1552924.

## REFERENCES

- [1] 2008. Harvard Database of Useful Biological Numbers. <http://bionumbers.hms.harvard.edu/bionumber.aspx?s=y&id=100706&ver=0>. (2008).
- [2] 2016. Virtual & Augmented Reality: The Next Big Computing Platform? <http://www.goldmansachs.com/our-thinking/pages/virtual-and-augmented-reality-report.html>. (2016).
- [3] 2017. LiGaze demo video. <http://dartnets.cs.dartmouth.edu/ligaze>. (2017).
- [4] 2017. Optical Eye Tracking System. [http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2010/yh428\\_aoo34/eyetracking/index.html](http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2010/yh428_aoo34/eyetracking/index.html). (2017).
- [5] 2017. ULPBench. <http://www.eembc.org/ulpbench/>. (2017).
- [6] S. J. Baek, K. A. Choi, C. Ma, Y. H. Kim, and S. J. Ko. 2013. Eyeball model-based iris center localization for visible image-based eye-gaze tracking systems. *IEEE Transactions on Consumer Electronics* 59, 2 (May 2013), 415–421.
- [7] Shumeet Baluja and Dean Pomerleau. 1994. *Non-Intrusive Gaze Tracking Using Artificial Neural Networks*. Technical Report.
- [8] D. Beymer and M. Flickner. 2003. Eye gaze tracking using an active stereo head. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- [9] Christopher M Bishop. 1995. *Neural networks for pattern recognition*. Oxford university press.
- [10] Leo Breiman. 1996. Bagging predictors. *Machine Learning* 24, 2 (1996), 123–140.
- [11] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. 2009. Wearable EOG Goggles: Eye-based Interaction in Everyday Environments. In *CHI Extended Abstracts on Human Factors in Computing Systems*.
- [12] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.
- [13] D. C. Cho, W. S. Yap, H. Lee, I. Lee, and W. Y. Kim. 2012. Long range eye gaze tracking system for a large screen. *IEEE Transactions on Consumer Electronics* 58, 4 (November 2012), 1119–1128.
- [14] Wei Chu and S Sathya Keerthi. 2007. Support vector ordinal regression. *Neural computation* 19, 3 (2007), 792–815.
- [15] Jack Cohen. 1988. *Statistical power analysis for the behavioral sciences*. Routledge.
- [16] P. M. Corcoran, F. Nanu, S. Petrescu, and P. Bigioi. 2012. Real-time eye gaze tracking for gaming design and consumer electronics systems. *IEEE Transactions on Consumer Electronics* 58, 2 (May 2012), 347–355.
- [17] T. N. Cornsweet and H. D. Crane. 1973. Accurate two-dimensional eye tracker using first and fourth Purkinje images. *Journal of the Optical Society of America* 63, 8 (Aug 1973), 921–928.
- [18] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the Reflectance Field of a Human Face. In *Proc. of SIGGRAPH*.
- [19] Y. Ebisawa and S. i. Satoh. 1993. Effectiveness of pupil area detection technique using two light sources and image difference method. In *Proc. of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 1268–1269.
- [20] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish S. Kulkarini, Shaun Kane, and Meredith Ringel Morris. 2017. Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design. In *Proc. of CHI*.
- [21] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.
- [22] Scott W. Greenwald, Luke Loreti, Markus Funk, Ronen Zilberman, and Pattie Maes. 2016. Eye Gaze Tracking with Google Cardboard Using Purkinje Images. In *Proc. of the ACM Conference on Virtual Reality Software and Technology*.
- [23] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D Graphics. *ACM Trans. Graph.* 31, 6 (Nov. 2012).
- [24] D. W. Hansen and Q. 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 (March 2010), 478–500.
- [25] Craig Hennessey, Borna Nouredin, and Peter Lawrence. 2006. A Single Camera Eye-gaze Tracking System with Free Head Motion. In *Proc. of the Symposium on Eye Tracking Research & Applications*.
- [26] Kenneth Holmqvist, Marcus Nyström, and Fiona Mulvey. 2012. Eye Tracker Data Quality: What It is and How to Measure It. In *Proc. of the Symposium on Eye Tracking Research and Applications*.
- [27] Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks* 13, 2 (2002), 415–425.
- [28] Michael Xuelin Huang, Tiffany C.K. Kwok, Grace Ngai, Stephen C.F. Chan, and Hong Va Leong. 2016. Building a Personalized, Auto-Calibrating Eye Tracker from User Interactions. In *Proc. of CHI*.
- [29] Qiang Ji and Xiaojie Yang. 2002. Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance. *Real-Time Imaging* 8, 5 (2002), 357–377.
- [30] Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In *Proc. of UbiComp Adjunct*.
- [31] A. E. Kaufman, A. Bandopadhyay, and B. D. Shaviv. 1993. An eye tracking computer user interface. In *Proc. of 1993 IEEE Research Properties in Virtual Reality Symposium*. 120–121.
- [32] K. Kim and R. Ramakrishna. 1999. Vision based eye gaze tracking for human computerinterface. In *Proc. of the IEEE International Conf. on Systems, Man and Cybernetics*.
- [33] R. Kothari and J. L. Mitchell. 1996. Detection of eye locations in unconstrained visual images. In *Proc. of 3rd IEEE International Conference on Image Processing*, Vol. 3. 519–522.
- [34] Bartosz Kunka and Bozena Kostek. 2009. Non-intrusive infrared-free eye tracking method. In *Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings (SPA), 2009*. IEEE, 105–109.
- [35] Ji Woo Lee, Chul Woo Cho, Kwang Yong Shin, Eui Chul Lee, and Kang Ryoung Park. 2012. 3D gaze tracking method using Purkinje images on eye optical model and pupil. *Optics and Lasers in Engineering* 50, 5 (2012), 736–751.
- [36] Dongheng Li, Jason Babcock, and Derrick J. Parkhurst. 2006. openEyes: A Low-cost Head-mounted Eye-tracking Solution. In *Proc. of the Symposium on Eye Tracking Research & Applications*.
- [37] Dongheng Li and Derrick Parkhurst. 2006. Open-Source Software for Real-Time Visible-Spectrum Eye Tracking. (2006).
- [38] Hsuan-Tien Lin and Ling Li. 2012. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation* 24, 5 (2012), 1329–1367.
- [39] Radosław Mantiuk, Michał Kowalik, Adam Nowosielski, and Bartosz Bazyluk. 2012. Do-it-yourself eye tracker: Low-cost pupil-based eye tracker for computer graphics applications. In *International Conference on Multimedia Modeling*. Springer, 115–125.
- [40] Addison Mayberry, Pan Hu, Benjamin Marlin, Christopher Salthouse, and Deepak Ganesan. 2014. iShadow: Design of a Wearable, Real-time Mobile Gaze Tracker. In *Proc. of MobiSys*.
- [41] Addison Mayberry, Yamin Tun, Pan Hu, Duncan Smith-Freedman, Deepak Ganesan, Benjamin M. Marlin, and Christopher Salthouse. 2015. CIDER: Enabling Robustness-Power Tradeoffs on a Computational Eyeglass. In *Proc. of MobiCom*.
- [42] C.H Morimoto, D Koons, A Amir, and M Flickner. 2000. Pupil detection and tracking using multiple light sources. *Image and Vision Computing* 18, 4 (2000), 331–335.
- [43] Carlos H. Morimoto and Marcio R. M. Mimica. 2005. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding* 98, 1 (April 2005), 4–24.
- [44] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky. 2000. Real-time stereo tracking for head pose and gaze estimation. In *Proc. of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*.
- [45] Harald Niederreiter. 1988. Low-discrepancy and low-dispersion sequences. *Journal of number theory* 30, 1 (1988), 51–70.
- [46] Basilio Noris, Jean-Baptiste Keller, and Aude Billard. 2011. A wearable gaze tracking system for children in unconstrained environments. *Computer Vision and Image Understanding* 115, 4 (2011), 476–486.
- [47] B. Nouredin, P.D. Lawrence, and C.F. Man. 2005. A non-contact device for tracking gaze in a human computer interface. *Computer Vision and Image Understanding* 98, 1 (2005), 52–82.
- [48] Takehiko Ohno and Naoki Mukawa. 2004. A Free-head, Simple Calibration, Gaze Tracking System That Enables Gaze-based Interaction. In *Proc. of the Symposium on Eye Tracking Research & Applications*.
- [49] Takehiko Ohno, Naoki Mukawa, and Atsushi Yoshikawa. 2002. FreeGaze: A Gaze Tracking System for Everyday Gaze Interaction. In *Proc. of the Symposium on Eye Tracking Research & Applications*.
- [50] Matthew O’A’Toole. 2016. *Optical Linear Algebra for Computational Light Transport*. Ph.D. Dissertation. University of Toronto.
- [51] Alexandra Papoutsaki, James Laskey, and Jeff Huang. 2017. SearchGazer: Webcam Eye Tracking for Remote Studies of Web Search. In *Proc. of the Conference on Conference Human Information Interaction and Retrieval*.
- [52] 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 Trans. Graph.* 35, 6 (Nov. 2016).
- [53] B. R. Pires, M. Devyver, A. Tsukada, and T. Kanade. 2013. Unwrapping the eye for visible-spectrum gaze tracking on wearable devices. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*.
- [54] J. P. H. Reulen, J. T. Marcus, D. Koops, F. R. de Vries, G. Tiesinga, K. Boshuizen, and J. E. Bos. 1988. Precise recording of eye movement: the IRIS technique Part 1. *Medical and Biological Engineering and Computing* 26, 1 (January 1988), 20–26.
- [55] D. A. Robinson. 1963. A Method of Measuring Eye Movement Using a Scleral Search Coil in a Magnetic Field. *IEEE Transactions on Bio-medical Electronics* 10, 4 (Oct 1963), 137–145.
- [56] Javier San Agustín, Henrik Skovsgaard, Emilie Mollenbach, Maria Barret, Martin Tall, Dan Witzner Hansen, and John Paulin Hansen. 2010. Evaluation of a Low-cost Open-source Gaze Tracker. In *Proc. of the Symposium on Eye-Tracking Research & Applications*.
- [57] Bernt Schiele and Alex Waibel. 1995. Gaze Tracking Based on Face-Color. In *1995 International Workshop on Automatic Face- and Gesture-Recognition*.

- [58] Bernhard Scholkopf and Alexander J Smola. 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.
- [59] Pradeep Sen, Billy Chen, Gaurav Garg, Stephen R. Marschner, Mark Horowitz, Marc Levoy, and Hendrik P. A. Lensch. 2005. Dual Photography. In *Proc. of SIGGRAPH*.
- [60] Laura Sesma, Arantxa Villanueva, and Rafael Cabeza. 2012. Evaluation of Pupil Center-eye Corner Vector for Gaze Estimation Using a Web Cam. In *Proc. of the Symposium on Eye Tracking Research and Applications*.
- [61] Sheng-Wen Shih and Jin Liu. 2004. A Novel Approach to 3-D Gaze Tracking Using Stereo Cameras. *Trans. Sys. Man Cyber. Part B* 34, 1 (February 2004), 234–245.
- [62] J. Sigut and S. A. Sidha. 2011. Iris Center Corneal Reflection Method for Gaze Tracking Using Visible Light. *IEEE Transactions on Biomedical Engineering* 58, 2 (Feb 2011), 411–419.
- [63] Andreas Sippl, Clemens Holzmann, Doris Zachhuber, and Alois Ferscha. 2010. *Real-Time Gaze Tracking for Public Displays*. Springer Berlin Heidelberg, Berlin, Heidelberg, 167–176.
- [64] Brian A. Smith, Qi Yin, Steven K. Feiner, and Shree K. Nayar. 2013. Gaze Locking: Passive Eye Contact Detection for Human-object Interaction. In *Proc. of UIST*.
- [65] Il'ya Meerovich Sobol'. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7, 4 (1967), 784–802.
- [66] Mark B Spitzer and Jeffrey Jacobsen. 1994. Eye tracking system having an array of photodetectors aligned respectively with an array of pixels. (July 19 1994). US Patent 5,331,149.
- [67] A. Tsukada, M. Shino, M. Devyver, and T. Kanade. 2011. Illumination-free gaze estimation method for first-person vision wearable device. In *IEEE International Conference on Computer Vision Workshops*. 2084–2091.
- [68] Nestor Voronka and Charles J Jacobus. 2001. Low-cost non-imaging eye tracker system for computer control. (Oct. 9 2001). US Patent 6,299,308.
- [69] G. Waldman. 2002. *Introduction to Light: The Physics of Light, Vision, and Color*. Dover Publications.
- [70] 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 *Proc. of ISWC*.
- [71] Xuehan Xiong, Zicheng Liu, Qin Cai, and Zhengyou Zhang. 2014. Eye Gaze Tracking Using an RGBD Camera: A Comparison with a RGB Solution. In *Proc. of UbiComp Adjunct*.
- [72] Hirotake Yamazoe, Akira Utsumi, Tomoko Yonezawa, and Shinji Abe. 2008. Remote Gaze Estimation with a Single Camera Based on Facial-feature Tracking Without Special Calibration Actions. In *Proc. of the Symposium on Eye Tracking Research & Applications*.
- [73] Dong Hyun Yoo, Jae Heon Kim, Bang Rae Lee, and Myoung Jin Chung. 2002. Non-contact eye gaze tracking system by mapping of corneal reflections. In *Proc. of Fifth IEEE International Conference on Automatic Face Gesture Recognition*.
- [74] Lan Zhang, Xiang-Yang Li, Wenchao Huang, Kebin Liu, Shuwei Zong, Xuesi Jian, Puchun Feng, Taeho Jung, and Yunhao Liu. 2014. It Starts with iGaze: Visual Attention Driven Networking with Smart Glasses. In *Proc. of MobiCom*.
- [75] Jie Zhu and Jie Yang. 2002. Subpixel Eye Gaze Tracking. In *Proc. of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*.
- [76] Zhiwei Zhu, Kikuo Fujimura, and Qiang Ji. 2002. Real-time Eye Detection and Tracking Under Various Light Conditions. In *Proc. of the Symposium on Eye Tracking Research & Applications*.
- [77] Z. Zhu and Q. Ji. 2007. Novel Eye Gaze Tracking Techniques Under Natural Head Movement. *IEEE Transactions on Biomedical Engineering* 54, 12 (December 2007), 2246–2260.