RGBDGaze: Gaze Tracking on Smartphones with RGB and Depth Data

Riku Arakawa
Carnegie Mellon University
Pittsburgh, USA
rarakawa@cs.cmu.edu

Mayank Goel
Carnegie Mellon University
Pittsburgh, USA
mayankgoel@cmu.edu

Chris Harrison
Carnegie Mellon University
Pittsburgh, USA
chris.harrison@cs.cmu.edu

Karan Ahuja
Carnegie Mellon University
Pittsburgh, USA
kahuja@cs.cmu.edu

ABSTRACT

Tracking a user's gaze on smartphones offers the potential for accessible and powerful multimodal interactions. However, phones are used in a myriad of contexts and state-of-the-art gaze models that use only the front-facing RGB cameras are too coarse and do not adapt adequately to changes in context. While prior research has showcased the efficacy of depth maps for gaze tracking, they have been limited to desktop-grade depth cameras, which are more capable than the types seen in smartphones, that must be thin and low-powered. In this paper, we present a gaze tracking system that makes use of today's smartphone depth camera technology to adapt to the changes in distance and orientation relative to the user's face. Unlike prior efforts that used depth sensors, we do not constrain the users to maintain a fixed head position. Our approach works across different use contexts in unconstrained mobile settings. The results show that our multimodal ML model has a mean gaze error of 1.89 cm; a 16.3% improvement over using RGB data alone (2.26 cm error). Our system and dataset offer the first benchmark of gaze tracking on smartphones using RGB+Depth data under different use contexts.

CCS CONCEPTS

• **Human-centered computing** → **Mobile devices**; • **Computing methodologies** → *Computer vision.*

KEYWORDS

gaze tracking, eye tracking, mobiles, deep learning

ACM Reference Format:
Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. 2022. RGBDGaze: Gaze Tracking on Smartphones with RGB and Depth Data. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI '22), November 7–11, 2022, Bengaluru, India.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3536221.3556568

1 INTRODUCTION

Computer interfaces with the ability to track a user's on-screen gaze location offer the potential for more accessible and powerful multimodal interactions [10, 23, 24, 32, 50], perhaps one day even supplanting the venerable cursor. While useful for desktop computing, gaze also promises to be a powerful way to interact with phones, especially given the need to adapt to a variety of usage contexts (*e.g.*, inability to use the touchscreen with encumbered hands). Specialized gaze tracking hardware – either worn [20] or placed in the environment [41] – can track gaze with very high resolution *i.e.,* 1.1 mm (0.45°) but, the need for specialized equipment is a significant barrier for consumer adoption. When relying on existing onboard hardware, research has primarily focused on user-facing RGB cameras. Unfortunately, gaze models utilizing this RGB data are too coarse for interactions with many user interface widgets, which are generally small on mobile devices. To help close this gap, researchers have started assessing the value of depth cameras to improve performance [30, 34, 55], but all research to date has focused on desktop-grade depth cameras (*e.g.*, Microsoft Kinect V2 [54] , Intel Real Sense [22]). These sensors are much more capable than the depth cameras seen in smartphones, which must be very thin and comparatively lower powered. Furthermore, much of this prior RGB+Depth (RGBD henceforth) gaze research had users maintain their head position in a highly constrained way (*e.g.*, chin rest [37, 39]). This rigid requirement is at odds with the usual way a typical user interacts with a phone while walking, riding public transport, carrying handbags, *etc.* Thus, it is important to build a gaze tracker that adapts to a user's changing context, uses existing hardware, and provides usable resolution.

This paper presents a gaze tracker that uses an off-the-shelf phone's front-facing RGB and depth camera. We collected data from and implemented our system in recent Apple iPhones (X and above), which feature a 1080p user-facing camera and Apple's structured light TrueDepth camera (similar to the technology used in the Kinect V1 [54] and earlier PrimeSense models). Our mobile RGBD dataset of 50 participants (which we make freely available at https://github.com/FIGLAB/RGBDGaze) is the first of its kind, offering RGBD data paired with user gaze location across a variety of use contexts. We implemented a CNN model based on a spatial weights structure to efficiently fuse the RGB and depth modalities. Our model achieves 1.89 cm on-screen euclidean error on our dataset in a leave-one-participant-out evaluation, showing a significant improvement over existing gaze-tracking methods in mobile settings. This result reaffirms the utility of fusing RGB and depth data, and offers the first benchmark for smartphone-based RGBD gaze tracking while a user is not simply sitting.

2 RELATED WORK

Gaze estimation is a well-studied field in computer science. For a full survey, refer to [4, 12]. Many approaches have been explored ranging from head-mounted devices [2, 27, 38, 42, 43, 46] to external

ICMI '22, November 7–11, 2022, Bengaluru, India

ACM ISBN 978-1-4503-9390-4/22/11.
https://doi.org/10.1145/3536221.3556568

Table 1: Comparison of our system with prior gaze tracking work. Grey-colored rows denote systems benchmarked using our dataset (detailed in Section 5.3). Unconstrained studies are those where the distance and/or angle between the capturing device and user was not static.

System	Capture Modality		Mobile	Unconstrained	Calibration	Mean Gaze Error
	RGB	Depth	Device	Study	–Free	
Columbia Gaze [37]	✓					-
UT MultiView [39]	✓				✓	6.5°
ETH-XGaze [51]	✓				✓	4.7°
MPII Gaze [52]	✓			✓	✓	6.3°
RT-GENE [14]	✓			✓	✓	7.7°
Gaze360 [21]	✓			✓	✓	13.5°
Wang and Ji [45]	✓	✓		✓		4.0°
Zhou et al. [55]	✓	✓		✓		1.99°
EyeDiap [34]	✓	✓		✓	✓	8.1°
ShanghaiTechGaze+ [30]	✓	✓		✓	✓	3.87 cm
EyeTab [47]	✓		✓		✓	2.58 cm
Valliappan et al. [44]	✓		✓	✓		0.46 cm
EyeMU [24]	✓		✓	✓		1.7 cm
iTracker [25]	✓		✓	✓		1.34 cm
iMon [18]	✓		✓	✓	✓	1.57 cm
TabletGaze [17]	✓		✓	✓	✓	3.17 cm
iTracker [25]	✓		✓	✓	✓	2.77 cm
Apple ARKit [7]	✓		✓	✓	✓	6.38 cm
Our System	**✓**	**✓**	**✓**	**✓**	**✓**	**1.89 cm**

sensors such as cameras in the environment [1, 33, 41]. In particular, recent advancement in machine and deep learning has significantly improved the accuracy of the image-based gaze tracking systems [44, 52]. In this section, we focus on gaze methods that most closely relate to our efforts, and in particular, works that present a gaze dataset or run on commodity smartphones.

2.1 Gaze Tracking Across Capture Modalities

Prior works have collected various datasets for training and evaluating the data-driven models, which are summarized in Table 1. Early works [37, 39] included the use of a chin rest to constrain the head movements of the participants. Following works tackled more naturalistic movement scenarios such as users interacting with their laptops [52] and even more challenging free-viewing tasks [14, 21]. With the advent of sophisticated cameras on mobile devices, researchers have gathered datasets tailored for these specific form factors. For instance, GazeCapture [25] collected data on smartphones employing a pool of crowd workers. Similarly, TabletGaze [17] collected data on a tablet for four different types of use contexts: standing, sitting, lying, and slouching.

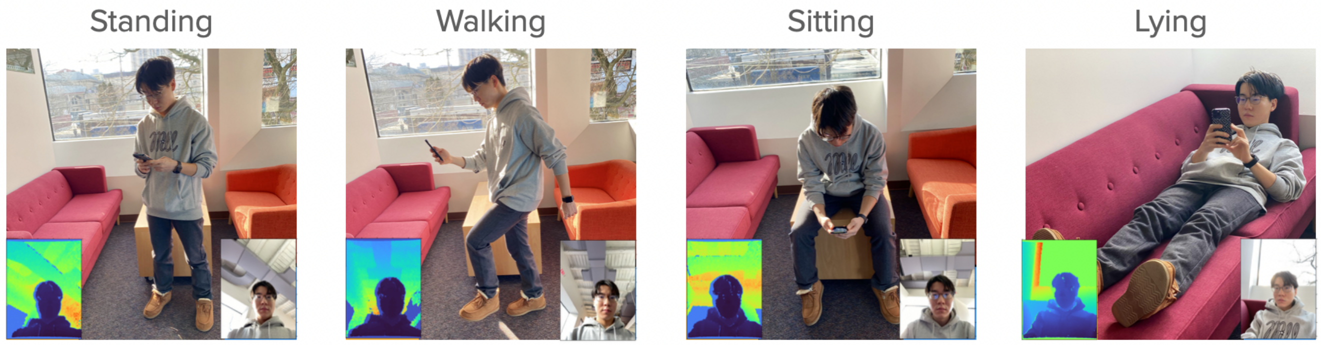
While most of these works assume RGB images as inputs for their models, several works have shown that the addition of depth channel can improve the accuracy. Originally, the depth information is used to obtain an accurate head pose or iris position, which is then utilized to estimate gaze in a model-based manner [9, 15, 19, 29, 40, 48, 55]. Along these lines, [30, 34] make use of an external Microsoft Kinect Depth sensor [54] or Intel RealSense RGBD sensor [22] to collect RGB and depth data from their participants as they view a 3D gaze target. Lian et al. [30] showed that the addition of the depth channel decreased the error by roughly 18% (from 4.67 cm to 3.87 cm error) on the EyeDiap [34] dataset. Similarly, Xiong et al. [48] showed 34% decrease in error by introducing depth to RGB (from 3.2° to 2.1°). However, to the best of our knowledge, there has been no prior work that has made use of depth cameras found in smartphones, which are low-powered and much less precise compared to their desktop-grade counterparts. Furthermore, the smartphone form-factor is inherently more unconstrained due to the mobile nature of the user and device.

2.2 Gaze Tracking on Mobile Devices

With the proliferation of mobile devices, their increasing screen sizes, and advances in camera capture technology, gaze estimation on smartphones has received significant attention. They can be broadly categorized into systems that require per-user calibration or are calibration-free.

Systems with per-user calibration tend to be more accurate as they have access to more personalized environmental and user data. Approaches such as [24], [25] and [44] achieve an error of 1.7 cm, 1.34 cm and 0.46 cm respectively using RGB cameras. Calibration-free techniques are more generalizable and can be used out of the box. These include RGB systems such as EyeTab [47] and TabletGaze [17] with a tracking error of 2.58 cm and 3.17 cm respectively. The demand for gaze tracking in smartphones has led to eye tracking API's for developers, such as the ARFaceAnchor module of Apple ARKit 4 [7].

Recently, Huynh *et al.* [18] built a new ML model trained using the GazeCapture dataset. While this model outperforms our system in terms of mean gaze error, the Gaze Capture dataset is limited


Standing

Walking

Sitting

Lying

Figure 1: Examples of the four use contexts we captured during data collection (standing, walking, sitting, and lying). Left-bottom insets are depth images rendered in the Turbo colormap. Right-bottom insets are views from the user-facing camera.

to older iPhones (iPhone 6S and below) with smaller screens and does not include data while the user is mobile. In contrast, we record data across four use contexts and on larger recent iPhones (iPhone X and above). Thus, a direct comparison of performance between the two models is difficult. The primary demonstration of our approach is the value of depth information while tracking gaze in a mobile context. While researchers have looked at depth from the perspective of larger and stationary computers, we posit that depth is more valuable when the user has greater physical freedom. No prior work has explored the fusion of RGB and depth imagery captured using mobile device hardware (Table 1). User-facing depth sensors are becoming increasingly common (*e.g.*, Apple iPhone 12, Google Pixel 4, Samsung Galaxy S10 Plus), opening the opportunity to significantly improve mobile gaze tracking accuracy. Our research ties these multimodal sources together to provide a robust and state-of-the-art smartphone-based gaze tracking system.

3 RGB+DEPTH DATASET & COLLECTION

We collected a first-of-its-kind dataset of RGBD on mobile devices. For data collection, we created an iOS application capable of recording and uploading gaze tracking data. Our application runs on Apple iPhone X and above, as they all feature a high resolution front-facing RGB camera and a TrueDepth camera (640 × 480 depth map interpolated from a 170 × 170 IR dot pattern). For our data collection, we recruited 50 participants (mean age 25 years, 34 male, 16 female). Fourteen of them wore glasses during the data collection. Twenty of the participants were recruited through in-class recruitment and the remainder of the 30 were recruited using an online sign-up form posted on various social media sites. The app was delivered via TestFlight.

The custom iOS application asked participants to look at a target (red dot) that was moving on the screen. While the user gazed at the target, synchronized RGB and depth imagery was logged at approximately 8 Hz, along with the ARKit gaze prediction (which we capture as a state-of-the-art commercial benchmark). The speed of the dot movement was varied to add diversity. The data collection was paused when the face was not detected using the Apple Vision Framework [6]. In a similar way to GazeCapture [25], we recorded device motion (9-axis IMU) sensor data synchronized to image data. While we did not use this sensor data in our study, this could be a useful resource in future work.

While using the app, the target dot was animated to cover various locations on the screen (Figure 2(a)). Specifically (and similar to [17, 25, 49]), we first pre-determined 5 × 7 = 35 fixed locations (Figure 2(b)), and then the dot repeatedly moved linearly (vertically, horizontally, and diagonally) from one location to another in a random fashion such that it covered each location four times. In addition, we implemented an "undo" functionality that lets the participant jump back to the previous gaze location (in case they were not paying attention or did not follow the target). The "undo" button shown in Figure 2(c) was hidden while the target was moving unless the participant tapped anywhere on the screen to stop its animation.

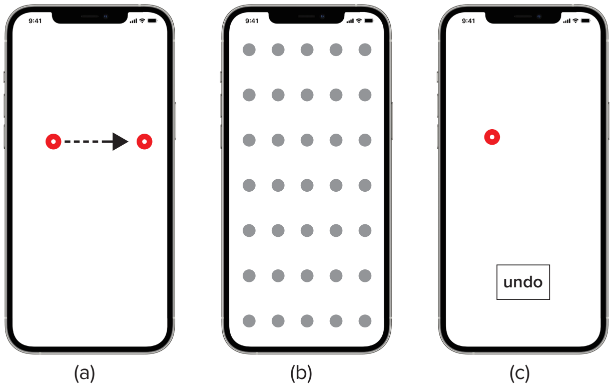


Figure 2: Our application for gaze data collection. (a) The target (red dot) moves around the screen. (b) The screen is divided by 35 fixed locations (illustrated here, but hidden from users) and the target moves from one to another. (c) Participants are able to stop by tapping the screen and "undo" a trial by pressing the button.

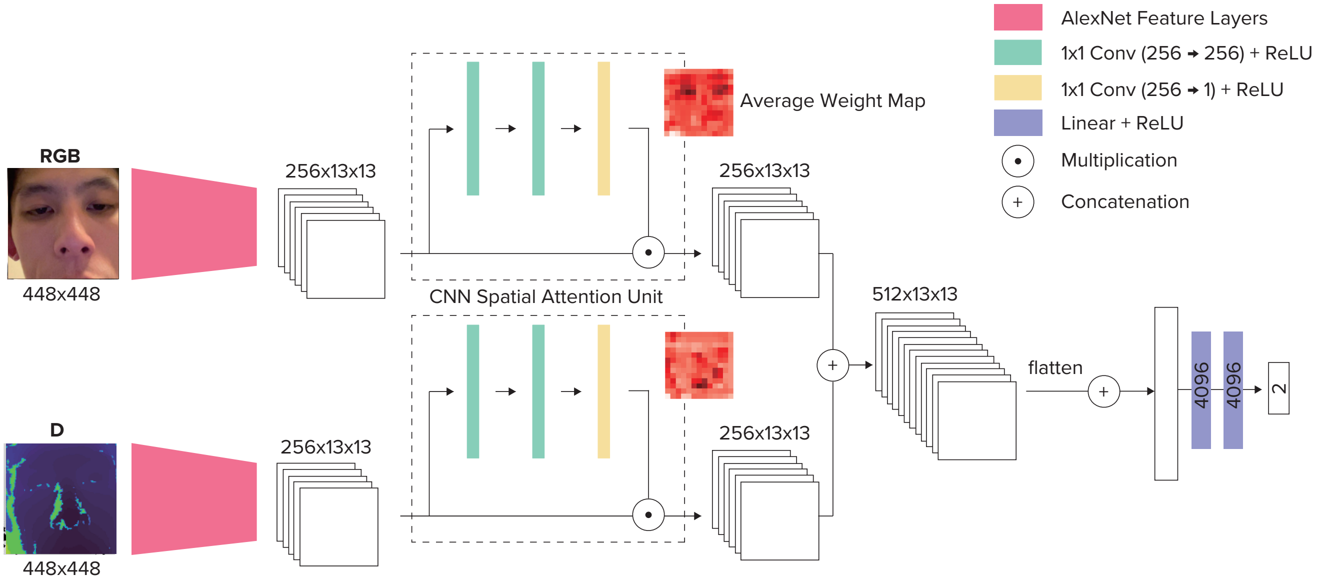


Figure 3: Overview of our multimodal deep learning architecture. Input is the RGB and depth image of the user's face, with the 2D gaze location on the screen as output. The spatial attention maps of the corresponding RGB and depth maps are visualized in red heatmaps (darker color intensity denotes higher attention value).

To ensure data reliability, we followed an approach similar to Krafka et al. [25]. First, during data recording, the app was kept in Airplane Mode to avoid any distractions via notifications. We further monitor user attention to ensure constant engagement with the application. This is done by introducing a color check mechanism during the task. Specifically, the color of the inner center of the moving dot changed to either white, green, or blue randomly during each animation (motion from one target gaze location to another). Users needed to perform a tapping action according to the color; tap nowhere if it was white, tap the right side of the screen if it was green, and tap the left side if it was blue. If they failed this color check, they were warned and the failed sequence was repeated.

The study consisted of four sessions spanning four distinct, yet common use contexts: standing, walking, sitting, and lying down (Figure 1). During each session, the participants were not given any specific instructions on how they should hold the device. We note that they routinely changed hands and arm positions during the sessions. Each collection session lasted for approximately four minutes, and the order of the sessions was randomized for each participant. The study took roughly 20 minutes and participants were compensated with $10 USD for their time. We also did not control for the environment, time of day or illumination during the data collection period. This led to high variability of data, critical in aiding the development of a robust, calibration-free gaze tracker.

We pruned the collected dataset by removing data points where the participants blinked. For this we employed an eye-aspect-ratio method [13]. Roughly 2% of our data consisted of blinks, which were dropped from analysis. In total, our final dataset consisted of 160,120 data points across 50 participants.

4 IMPLEMENTATION

4.1 Network Architecture

We developed a multimodal learning-based method for estimating a user's gaze on a smartphone. We first crop the user's face using the Apple Vision Framework [6]. The cropped face (448 × 448 pixels) in RGB and depth views serve as the input to our multi-input Convolutional Neural Network (CNN). The output is the predicted 2D gaze location in the screen coordinate frame of the smartphone. The overview of our CNN can be seen in Figure 3. For our image-based feature extractor, we make use of spatial attention [31, 53] neural networks. This helps assign different information weights to different regions of the facial image. For example, it will automatically assign higher weights to the eyes or rather different parts of the eyes would be weighted differently based on their information entropy in the CNN. Prior work has found such approaches [53] to be more accurate and computationally less intensive than those that crop different regions and feed them individually to a model [25]. We warm-start our RGB and depth convolutional feature extractors (Figure 3 pink color) with AlexNet [26] weights. The embedding is then passed to three fully-connected layers with the ReLU activation, outputting the final two-dimensional gaze value.

4.2 Training Protocol

The model is implemented with PyTorch 1.9.1 [35]. The RGB part of the model is first trained with the GazeCapture dataset while the depth part is initialized without any pretraining. We use a batch size of 16 and update the model weights using the SGD optimizer [8] with the initial learning rate 0.0005, the momentum 0.9, the weight decay 0.0001. The learning rate is decayed by 0.1 for every

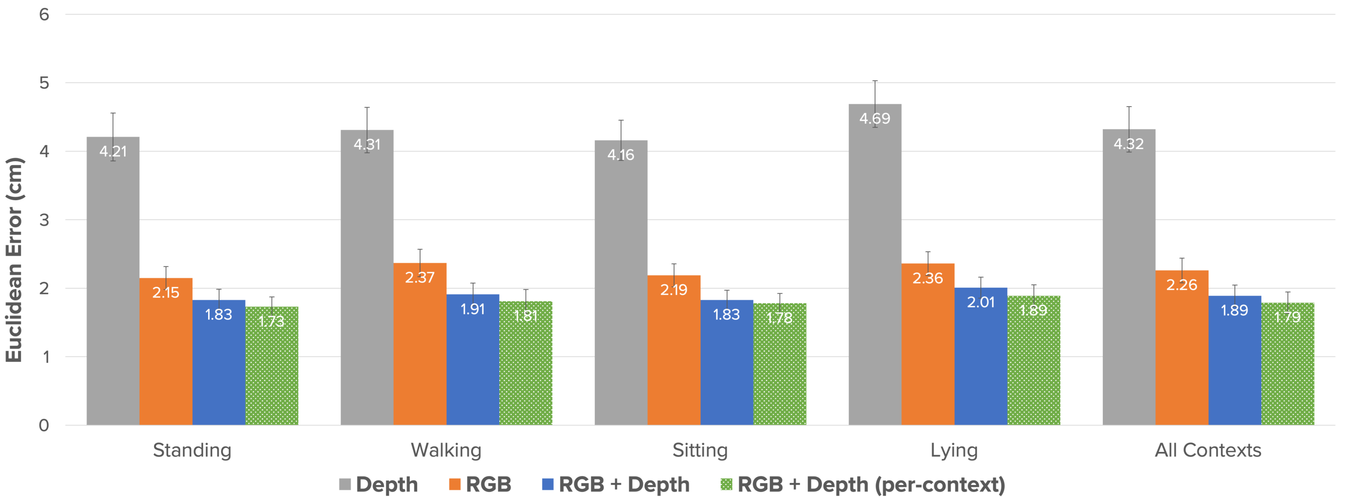


Figure 4: Overall accuracy of our approach across different data input modalities (RGB and depth) and use contexts (sitting, standing, walking and lying). The error bars are standard error.

five epochs. We use the mean squared error between the ground truth and predicted gaze points as a loss function. We train our model up to 20 epochs on an NVIDIA GeForce GTX 1080 Ti GPU, and it takes approximately 12 hours to train one fold in leave-one-participant-out procedure.

4.3 On-Device Model

We deployed our model in a real-time iOS application. We first converted our trained PyTorch model to Core ML using Apple's CoreMLToolkit [5]. A pair of synchronized RGB and depth frames are then pre-processed, which includes finding the face crop using Apple's Vision Framework [6] and normalizing the data between -1 and 1. This data is then sent to our CNN for prediction. On the iPhone 12 Pro Max, our model runs at 7 fps with an average latency of 121.3 ms (SD = 9.2 ms) from captured photo to gaze prediction. Using RGB alone, our system has a latency of 85.3 ms (SD = 7.4 ms) and runs at 10 fps. The tracking can run continuously for around 3.5 hours on the iPhone 12 Pro (battery capacity of 10.8 Wh). The model outputs a 2D gaze prediction, which is plotted on the screen. As a comparison point, Apple's Animoji feature, which digitizes people's faces and tracks their eyes, runs with a latency of around ~110 ms on an iPhone 12 Pro [3]. Similar to data collection, the predictions are paused when the face is not detected by the Apple Vision Framework [6]. Please refer to the Video Figure for a real-time demo.

5 RESULTS AND DISCUSSION

In this section, we evaluate the efficacy of our multimodal model on our RGBD dataset. We first compute the performance metrics of our system across different input data modalities and use contexts, and then compare our system to RGB-based, state-of-the-art gaze tracking methods.

5.1 Overall Accuracy

To evaluate the efficacy of our model, we follow a leave-one-participant-out protocol. The model is calibration-free, as no per-participant data is shared between the splits. Overall, our model achieves a euclidean gaze error of 1.89 cm (SD = 1.09 cm) when using RGBD data (Figure 4 far-right chart).

Upon inspection, we find that our RGBD model has a very high error when the eyes are partially closed or occluded (by the user's hair, glasses frame, hands, *etc.*). In these cases, it is impossible to resolve a full view of both eyes and the model falls back on head pose estimation for gaze. Other cases of error include poor lighting conditions, strong ocular reflections and motion blur. Figure 5 showcases these sample error cases of the model.

To test the performance impact of each data modality, we trained the following model variants: depth-only, RGB-only, and RGBD. Figure 4 summarizes this result. Our RGB-only model has a euclidean error of 2.26 cm (SD = 1.27 cm), which falls to 1.89 cm (SD = 1.09 cm) with the addition of depth data. Anecdotally, upon visualizing the spatial attention maps of our model (Figure 3), we find that the RGB attention model assigns a higher weight to the two eyes, thus focusing on eye gaze; while the depth model assigns a higher weight to the central region and edge of the face, thereby focusing on head pose.

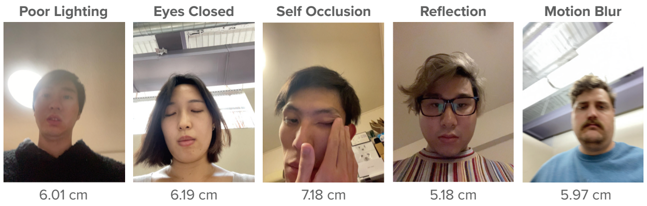


Figure 5: Example images where our RGBD Gaze model had high error. Euclidean error is noted below each image.

5.2 Effect of Use Context on Accuracy

Different use contexts result in different body postures [3] and face visibility [17], and also introduce artifacts such as motion blur. When we test our model across different use contexts (see Figure 4, blue bars) using the leave-one-participant-out protocol, we find that lying has the highest error (2.01 cm, SD = 1.18 cm) while sitting and standing have the lowest error (1.83 cm, SD = 1.10 cm and 1.83 cm, SD = 1.01 cm). This is in line with prior work [17] and can be attributed to the least facial visibility for the lying down context. Compared to standing and sitting, walking has a relatively lower accuracy, that is, 1.91 cm (SD = 1.08 cm). This can be attributed to motion blur and head motion caused due to the movement of the smartphone in the walking scenario.

Prior work has successfully demonstrated the detection of different coarse body poses/activities (*e.g.*, sitting, standing, running, walking, lying down) using smartphone motion data [3, 11, 28, 36]. To quantify the performance effect of each use context, we tested the performance of a per-context calibrated model. Here we validate the model on data from only that particular context. Overall, this reduces error by 5.3% (broken out by use context in Figure 4, green bars).

5.3 Comparison with Prior Work

A summary of comparative prior works can be found in Table 1. To the best of our knowledge, no prior work has made use of RGBD on smartphones for gaze estimation. We therefore benchmark our approach with two state-of-the-art RGB-based systems: Apple ARKit 4 ARFaceAnchor model [7] and iTracker [25]. Apple's ARKit provides an easy-to-integrate library for developers, and the iTracker model has been trained on a multitude of mobile devices (iPhone 4-6, iPad Pro, iPad Air). These two systems were run on the same data as our own model. ARKit and iTracker achieve a mean euclidean error of 6.38 cm and 2.77 cm, respectively. In contrast, our model, making use of RGBD, offered a much lower euclidean error of 1.89 cm.

To test the efficacy of our RGB spatial attention model, we benchmark it on the GazeCapture test dataset utilized by iTracker [25]. iTracker achieves an error of 2.04 cm (without any data augmentation) on their dataset. Our RGB-only model achieves a similar error of 2.03 cm on the dataset (vs. 2.26 cm on our test dataset). This dip can be attributed to the varying use contexts and challenging capture scenarios of our dataset as well as the larger screen size of the devices used. We also find that our model achieved better accuracies compared to tablet-based gaze estimation works such as TabletGaze [17] and EyeTab [47], which reported an error of 3.17 cm and 2.58 cm respectively. Note that these comparisons are provided as a reference, as all these methods were tested on different datasets.

6 OPEN SOURCE MODEL & DATA

To enable future research to build on our system and contribute to this domain, we have made our code, models, and dataset freely available at https://github.com/FIGLAB/RGBDGaze. Our synchronized RGB and depth dataset is the first of its kind for mobile devices. It also labels each segment by user activity context, along with synchronized 9-axis IMU data. We thank our participants for their permission to share this data.

7 LIMITATIONS & FUTURE WORK

While the accuracies of our system are promising, there are several key limitations that will need to be overcome before it is ready for commercial adoption. First is the accuracy of the system. Even with a calibration-free gaze error of under 2 cm, the accuracy falls short of the sub-millimeter accuracy afforded by dedicated eye trackers. In the future, this could be improved by collecting data across a wider array of mobile devices, scenes and users. The proliferation of depth cameras on smartphones and tablets (such as the Google Pixel 4 [16] and the Apple iPad Pro) could help with training a more generalizable gaze tracking model.

We also note that while the current contexts are encouraging and naturalistic, they can still be expanded. We can cover more situated contexts, for example, users interacting with the smartphone while driving, biking, or climbing stairs. Furthermore, rather than an experimenter conducting the study, we can increase the diversity of our dataset by crowd-sourcing the data collection (as done by Krafka et al. [25] and Xu et al. [49]). We believe that such a large-scale dataset consisting of RGBD modality can achieve high accuracy without per-user calibration, enabling practical gaze-powered mobile interactions.

8 CONCLUSION

Our work explores the feasibility of gaze tracking on smartphones using RGB+Depth (RGBD) data. We collected data from 50 participants across four use contexts and then trained a CNN model based on a spatial weights structure that can efficiently fuse our multimodal streams. Results demonstrate that our model offers improved accuracy, down to 1.89 cm euclidean error. While future work remains, this result suggests that RGB and depth information offers promise in enabling unconstrained mobile gaze tracking and could unlock a wealth of new and interesting end-user applications.

REFERENCES

[1] Karan Ahuja, Ruchika Banerjee, Seema Nagar, Kuntal Dey, and Ferdous A. Barbhuiya. 2016. Eye center localization and detection using radial mapping. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*. IEEE, 3121–3125. https://doi.org/10.1109/ICIP.2016.7532934

[2] Karan Ahuja, Rahul Islam, Varun Parashar, Kuntal Dey, Chris Harrison, and Mayank Goel. 2018. Eyespyvr: Interactive eye sensing using off-the-shelf, smartphone-based vr headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–10. https://doi.org/10.1145/3214260

[3] Karan Ahuja, Sven Mayer, Mayank Goel, and Chris Harrison. 2021. Pose-on-the-Go: Approximating User Pose with Smartphone Sensor Fusion and Inverse Kinematics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[4] Andronicus Ayobami Akinyelu and Pieter J. Blignaut. 2020. Convolutional Neural Network-Based Methods for Eye Gaze Estimation: A Survey. *IEEE Access* 8 (2020), 142581–142605. https://doi.org/10.1109/ACCESS.2020.3013540

[5] Apple. 2017. CoreML Framework. https://developer.apple.com/documentation/coreml

[6] Apple. 2017. Vision Framework. https://developer.apple.com/documentation/vision

[7] Apple. 2020. ARKit Framework. https://developer.apple.com/documentation/arkit

[8] Léon Bottou. 2010. Large-Scale Machine Learning with Stochastic Gradient Descent. In *19th International Conference on Computational Statistics, COMPSTAT 2010, Paris, France, August 22-27, 2010 - Keynote, Invited and Contributed Papers*. Physica-Verlag, 177–186. https://doi.org/10.1007/978-3-7908-2604-3_16

[9] Haibin Cai, Xiaolong Zhou, Hui Yu, and Honghai Liu. 2015. Gaze estimation driven solution for interacting children with ASD. In *2015 International Symposium on Micro-NanoMechatronics and Human Science, MHS 2015, Nagoya, Japan,*

November 23-25, 2015. IEEE, New York, 1–6. https://doi.org/10.1109/MHS.2015.7438336

[10] Marcus Carter, Joshua Newn, Eduardo Velloso, and Frank Vetere. 2015. Remote Gaze and Gesture Tracking on the Microsoft Kinect: Investigating the Role of Feedback. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction, OZCHI 2015, Parkville, VIC, Australia, December 7-10, 2015*. ACM, New York, 167–176. https://doi.org/10.1145/2838739.2838778

[11] Pierluigi Casale, Oriol Pujol, and Petia Radeva. 2011. Human activity recognition from accelerometer data using a wearable device. In *Iberian conference on pattern recognition and image analysis*. Springer, 289–296.

[12] Dario Cazzato, Marco Leo, Cosimo Distante, and Holger Voos. 2020. When I Look into Your Eyes: A Survey on Computer Vision Contributions for Human Gaze Estimation and Tracking. *Sensors* 20, 13 (2020), 3739. https://doi.org/10.3390/s20133739

[13] Jan Cech and Tereza Soukupova. 2016. Real-time eye blink detection using facial landmarks. *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague* (2016), 1–8.

[14] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. 2018. RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X (Lecture Notes in Computer Science, Vol. 11214)*. Springer, 339–357. https://doi.org/10.1007/978-3-030-01249-6_21

[15] Reza Shoja Ghiass and Ognjen Arandjelovic. 2016. Highly Accurate Gaze Estimation Using a Consumer RGB-D Sensor. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. IJCAI/AAAI Press, 3368–3374. http://www.ijcai.org/Abstract/16/476

[16] Google. 2020. uDepth: Real-time 3D Depth Sensing on the Pixel 4. https://ai.googleblog.com/2020/04/udepth-real-time-3d-depth-sensing-on.html

[17] Qiong Huang, Ashok Veeraraghavan, and Ashutosh Sabharwal. 2017. TabletGaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Mach. Vis. Appl.* 28, 5-6 (2017), 445–461. https://doi.org/10.1007/s00138-017-0852-4

[18] Sinh Huynh, Rajesh Krishna Balan, and JeongGil Ko. 2021. iMon: Appearance-based gaze tracking system on mobile devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–26.

[19] Reza Jafari and Djemel Ziou. 2015. Eye-gaze estimation under various head positions and iris states. *Expert Syst. Appl.* 42, 1 (2015), 510–518. https://doi.org/10.1016/j.eswa.2014.08.003

[20] Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*. ACM, New York, 1151–1160. https://doi.org/10.1145/2638728.2641695

[21] Petr Kellnhofer, Adrià Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. 2019. Gaze360: Physically Unconstrained Gaze Estimation in the Wild. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, New York, 6911–6920. https://doi.org/10.1109/ICCV.2019.00701

[22] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. 2017. Intel(R) RealSense(TM) Stereoscopic Depth Cameras. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, New York, 1267–1276. https://doi.org/10.1109/CVPRW.2017.167

[23] Mohamed Khamis, Florian Alt, and Andreas Bulling. 2018. The past, present, and future of gaze-enabled handheld mobile devices: survey and lessons learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2018, Barcelona, Spain, September 03-06, 2018*. ACM, New York, 38:1–38:17. https://doi.org/10.1145/3229434.3229452

[24] Andy Kong, Karan Ahuja, Mayank Goel, and Chris Harrison. 2021. EyeMU Interactions: Gaze+IMU Gestures on Mobile Devices. In *ICMI '21: nternational Conference on Multimodal Interaction, Canada, October 18-22, 2021*. ACM, New York, 1–10. https://doi.org/10.1145/3462244.3479938

[25] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra M. Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, New York, 2176–2184. https://doi.org/10.1109/CVPR.2016.239

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90. https://doi.org/10.1145/3065386

[27] Christian Lander, Markus Löchtefeld, and Antonio Krüger. 2017. hEYEbrid: A hybrid approach for mobile calibration-free gaze estimation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4 (2017), 149:1–149:29. https://doi.org/10.1145/3161166

[28] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. 2017. Human activity recognition from accelerometer data using Convolutional Neural Network. In *2017 ieee international conference on big data and smart computing (bigcomp)*. IEEE, 131–134.

[29] Jianfeng Li and Shigang Li. 2014. Eye-Model-Based Gaze Estimation by RGB-D Camera. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, New York, 606–610. https://doi.org/10.1109/CVPRW.2014.93

[30] Dongze Lian, Ziheng Zhang, Weixin Luo, Lina Hu, Minye Wu, Zechao Li, Jingyi Yu, and Shenghua Gao. 2019. RGBD Based Gaze Estimation via Multi-Task CNN. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2488–2495. https://doi.org/10.1609/aaai.v33i01.33012488

[31] Song Liu, Danping Liu, and Haiyang Wu. 2020. Gaze Estimation with Multi-Scale Channel and Spatial Attention. In *ICCPR 2020: 9th International Conference on Computing and Pattern Recognition, Xiamen, China, October 30 - Vovember 1, 2020*. ACM, New York, 303–309. https://doi.org/10.1145/3436369.3437438

[32] Sven Mayer, Gierad Laput, and Chris Harrison. 2020. Enhancing Mobile Voice Assistants with WorldGaze. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*. ACM, New York, 1–10. https://doi.org/10.1145/3313831.3376479

[33] Christopher McMurrough, Vangelis Metsis, Jonathan Rich, and Fillia Makedon. 2012. An eye tracking dataset for point of gaze detection. In *Proceedings of the 2012 Symposium on Eye-Tracking Research and Applications, ETRA 2012, Santa Barbara, CA, USA, March 28-30, 2012*. ACM, New York, 305–308. https://doi.org/10.1145/2168556.2168622

[34] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. 2014. EYEDIAP: a database for the development and evaluation of gaze estimation algorithms from RGB and RGB-D cameras. In *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*. ACM, New York, 255–258. https://doi.org/10.1145/2578153.2578190

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 8024–8035.

[36] Wen Qi, Hang Su, Chenguang Yang, Giancarlo Ferrigno, Elena De Momi, and Andrea Aliverti. 2019. A fast and robust deep convolutional neural networks for complex human activity recognition using smartphone. *Sensors* 19, 17 (2019), 3731.

[37] Brian A. Smith, Qi Yin, Steven K. Feiner, and Shree K. Nayar. 2013. Gaze locking: passive eye contact detection for human-object interaction. In *The 26th Annual ACM Symposium on User Interface Software and Technology, UIST'13, St. Andrews, United Kingdom, October 8-11, 2013*. ACM, New York, 271–280. https://doi.org/10.1145/2501988.2501994

[38] Dan Su, Youfu Li, and Hao Chen. 2019. Toward Precise Gaze Estimation for Mobile Head-Mounted Gaze Tracking Systems. *IEEE Trans. Ind. Informatics* 15, 5 (2019), 2660–2672. https://doi.org/10.1109/TII.2018.2867952

[39] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. 2014. Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, New York, 1821–1828. https://doi.org/10.1109/CVPR.2014.235

[40] Li Sun, Zicheng Liu, and Ming-Ting Sun. 2015. Real time gaze estimation with a consumer depth camera. *Inf. Sci.* 320 (2015), 346–360. https://doi.org/10.1016/j.ins.2015.02.004

[41] Tobii Pro AB. 2014. Tobii Pro Lab. Computer software. http://www.tobiipro.com/

[42] Marc Tonsen, Julian Steil, Yusuke Sugano, and Andreas Bulling. 2017. InvisibleEye: Mobile Eye Tracking Using Multiple Low-Resolution Cameras and Learning-Based Gaze Estimation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3 (2017), 106:1–106:21. https://doi.org/10.1145/3130971

[43] Akihiro Tsukada, Motoki Shino, Michael Devyver, and Takeo Kanade. 2011. Illumination-free gaze estimation method for first-person vision wearable device. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*. IEEE Computer Society, New York, 2084–2091. https://doi.org/10.1109/ICCVW.2011.6130505

[44] Nachiappan Valliappan, Na Dai, Ethan Steinberg, Junfeng He, Kantwon Rogers, Venky Ramachandran, Pingmei Xu, Mina Shojaeizadeh, Li Guo, Kai Kohlhoff, and Vidhya Navalpakkam. 2020. Accelerating eye movement research via accurate and affordable smartphone eye tracking. *Nature Communications* 11, 1 (Sept. 2020). https://doi.org/10.1038/s41467-020-18360-5

[45] Kang Wang and Qiang Ji. 2016. Real time eye gaze tracking with Kinect. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*. IEEE, 2752–2757. https://doi.org/10.1109/ICPR.2016.7900052

[46] Eric Whitmire, Laura C. Trutoiu, Robert Cavin, David Perek, Brian Scally, James Phillips, and Shwetak N. Patel. 2016. EyeContact: scleral coil eye tracking for virtual reality. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers, ISWC 2016, Heidelberg, Germany, September 12-16, 2016*. ACM,

184–191. https://doi.org/10.1145/2971763.2971771

[47] Erroll Wood and Andreas Bulling. 2014. EyeTab: model-based gaze estimation on unmodified tablet computers. In *Eye Tracking Research and Applications, ETRA '14, Safety Harbor, FL, USA, March 26-28, 2014*. ACM, New York, 207–210. https://doi.org/10.1145/2578153.2578185

[48] Xuehan Xiong, Qin Cai, Zicheng Liu, and Zhengyou Zhang. 2014. Eye gaze tracking using an RGBD camera: a comparison with a RGB solution. In *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*. ACM, New York, 1113–1121. https://doi.org/10.1145/2638728.2641694

[49] Pingmei Xu, Krista A. Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. 2015. TurkerGaze: Crowdsourcing Saliency with Webcam based Eye Tracking. *CoRR* abs/1504.06755 (2015). arXiv:1504.06755 http://arxiv.org/abs/1504.06755

[50] Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. 2017. Smartphone-Based Gaze Gesture Communication for People with Motor Disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017*. ACM, New York, 2878–2889. https://doi.org/10.1145/3025453.3025790

[51] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. 2020. ETH-XGaze: A Large Scale Dataset for Gaze Estimation Under Extreme Head Pose and Gaze Variation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 12350)*. Springer, 365–381. https://doi.org/10.1007/978-3-030-58558-7_22

[52] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-based gaze estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, New York, 4511–4520. https://doi.org/10.1109/CVPR.2015.7299081

[53] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, New York, 2299–2308. https://doi.org/10.1109/CVPRW.2017.284

[54] Zhengyou Zhang. 2012. Microsoft kinect sensor and its effect. *IEEE multimedia* 19, 2 (2012), 4–10.

[55] Xiaolong Zhou, Haibin Cai, Youfu Li, and Honghai Liu. 2017. Two-eye model-based gaze estimation from a Kinect sensor. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*. IEEE, New York, 1646–1653. https://doi.org/10.1109/ICRA.2017.7989194