2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) | 979-8-3503-7449-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/VRW62533.2024.00144

A Calibration Interface for 3D Gaze Depth Disambiguation in Virtual Environments

Cameron Boyd* Augusta University
Mia Thompson† Augusta University
Madeline Smith‡ Augusta University
Gokila Dorai§ Augusta University
Jason Orlosky¶ Augusta University

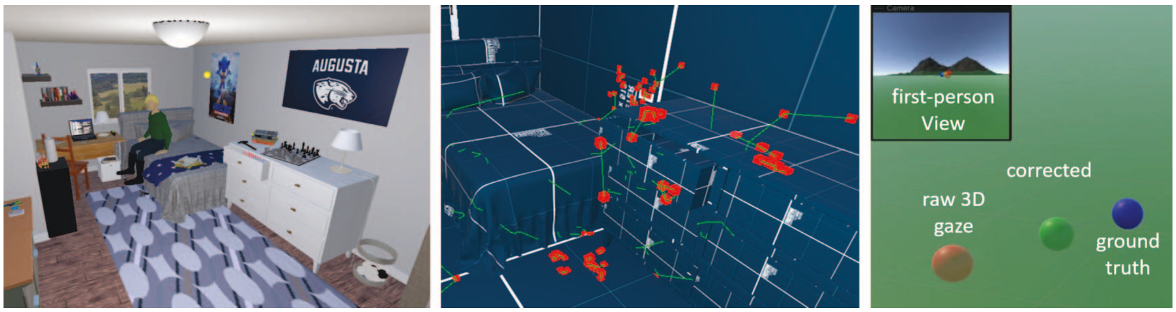

Figure 1: Images showing A) our virtual copy of a college dorm for collecting eye tracking depth data, B) a corresponding gaze plot of 3D eye-points rendered after searching the environment that shows the difficulty in discerning between points on the shelving and wall due to a lack of depth information in the line-of-site computation, and C) an image of our calibration interface, showing the calibration process in which a user follows a blue target acquire data for depth-correction modeling, where the red ball is raw gaze and the green ball is the corrected position.

Abstract

In Augmented and Virtual Reality, accurate eye tracking is a requirement for many applications. Though state-of-the art algorithms have enabled sub-degree accuracy for line-of-sight tracking, one remaining problem is that depth tracking, i.e. calculation of the gaze intersection at various depths, is still inaccurate. In this paper, we propose a 3D calibration method that accounts for gaze depth in addition to line-of-sight. By taking advantage of 3D calibration points and modeling the relationship between gaze inaccuracy and depth, we show that we can improve depth calculations and better determine the 3D position of gaze intersections in virtual environments.

1 Introduction and Prior Work

In eye-tracked augmented and virtual reality (AR/VR), eye gaze is typically provided as a single ray projected out of the center of the user's eye position, which provides a line-of site ray for computing intersections with virtual objects. This is typically calculated using the intersection between the gaze rays of the left and right eyes based on a geometric model [5]. In VR or environments where the structure of the environment is known, one major challenge is to accurately disambiguate between objects that happen to fall within the same line of site. One example is the difference between the corner of the dresser and the wall in the left image of Figure 1. The middle image shows the scan path of a user who has looked at an object on the corner of the dresser. However, due to inherent inaccuracy in the eye tracker, the gaze points are distributed between the wall and the object, obfuscating the true target.

*e-mail:cboyd@augusta.edu
†e-mail:miathompson@augusta.edu
‡e-mail:msmith40@augusta.edu
§e-mail:gdorai@augusta.edu
¶e-mail:jorlosky@augusta.edu

While using the 3D depth computed from the gaze ray intersection of the left and right eyes can be useful, the accuracy of this computation is not good enough to be able to effectively disambiguate between the two objects. In this paper, we present a calibration method to improve the accuracy of gaze depth computations, which can help with depth disambiguation for a broad range of eye tracking applications in virtual environments. We developed the environments on the left-hand side of Figure 1 as a testbed for gaze tracking in a practical environment, which resulted in a depth calibration procedure that uses a moving target to record and model 3D gaze (depth) error.

It is noteworthy that other articles have acknowledged the persisting concern regarding the calibration of eye tracking, even following the rectification of optical lens-related distortions [2]. Some approaches have used machine learning to be able to accurately estimate gaze depth, though this becomes very difficult after a certain distance [3]. Some work already exists to be able to disambiguate between the physical planes of a multi-focal display, but this approach only works for discrete distances [4]. Another study underscores the inadequacy of depth perception within uncomplicated virtual settings and underscores that rudimentary interventions do not ameliorate the precision of depth estimation [1].

2 Environment Setup and 3D Intersection

To devise an algorithm for depth correction and accurate eye tracking data measurement, we build the outdoor calibration scene shown on the right of Figure 1. The scene encapsulates three primary functions, including initialization and centering, target movement and data recording/calibration, and real-time correction. The three colored spheres shown in the rightmost image of Figure 1 function to assist with this process. Initially, a blue sphere traverses the Z-axis of the environment from 2 to 5 meters. The observer's task is to focus solely on tracking the blue sphere, whose coordinates are subsequently logged to establish a baseline correspondence between the ground truth (blue sphere) and the 3D intersection over time.

Once this ground truth data has been recorded, a second real-time correction phase is activated and our our depth calibration algorithm is applied to the user's new gaze input, resulting in a

979-8-3503-7449-0/24/$31.00 ©2024 IEEE
DOI 10.1109/VRW62533.2024.00144

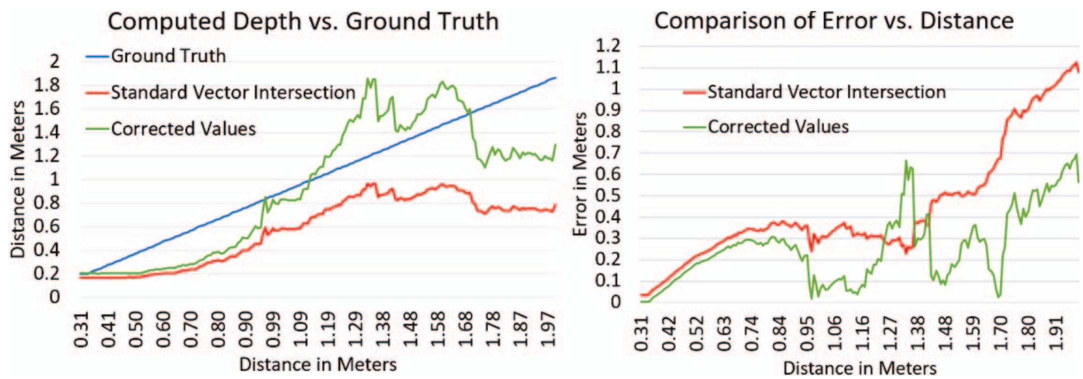

Figure 2: The left chart shows how our depth calibration (green) improves the accuracy of the corresponding vector-based 3D gaze position calculation (red), with respect to ground-truth (blue). The right chart shows the relative error of the standard vector intersection (red) versus values corrected with our depth calibration (green) from 30 centimeters to 2 meters (right).

refined determination of 3D gaze that is corrected for depth. This integration enhances the precision of eye tracking data, the details and results of which are described next.

3 Depth Calibration Algorithm

To obtain corrected depth points based on the entire input data set, we utilize the SciPy Python library to fit a curve to the data and model its error. This takes the form of the following formula, which is fit by the exponential function:

$$\text{outputDepth} = a \cdot e^{b \cdot \text{inputDepth}},$$

where a and b are coefficients returned by the python script, inputDepth is the raw, uncorrected depth of the 3D gaze point, i.e., the distance from the HMD to the intersection, and outputDepth is the corrected depth. This formula is then sent back to Unity, where we parse it, reformat the perameters to C# syntax, and align the resulting depth with the user's head position and gaze direction.

Note that though our interface is set up to record multiple directions for the gaze vectors, we have only included the data for the depth in the face-forward direction. Based on initial observations, these other gaze directions do not appear to precisely fit the corrective model used for the face-forward direction, and we have a directional model planned as future work.

4 Testing and Discussion

In order to record and test data, we developed an interface in which a blue gaze target gradually moved from 30cm to 2m. In the calibration phase, a user follows this sphere by gazing at it while it moves. Data from both eyes are recorded during this process, and when movement is finished, the gaze data is sent to text file that is then fed into the Python script that produces the model that corrects the depth.

Once the depth-correction model is computed, it is fed back into the virtual environment, and the erroneous computations (red in the graphs) are corrected (green in the graphs), as shown in Figure 2. To re-measure the accuracy of our approach compared to ground truth, we re-render the target, ask the user to view the target again, and record both the erroneous and corrected positions for a new test set of data. The blue line on the left figure represents ground truth, i.e. the path of the gaze target as it moves from 30cm to 2m, which are the initial measurements from one of the author's eye recordings.

Though the calibration process actually records data up to 5m, results after 2m are not included since the human eye's ability to verge at distances greater than 2m decreases to a point that it is not discernible from gazing at infinity, making it impossible to correct gaze past that point. This is one limitation of our system and a limitation of gaze-depth correction techniques in general.

4.1 Applications in Training and Education

A practical application scenario of this work is to address a crucial gap in training methodologies for first responders and or investigators. The purpose of the dorm room in Figure 1 is actually to simulate a first responder call. Our work can help with the disambiguation of gaze targets, which can assist with digital forensics training, evidence collection via scene reconstruction, triage, and officer training. These activities include de-escalation techniques, simulation of encounters, and evaluations post-encounter, all aimed at improving the accuracy with which we can track or measure trainee gaze patterns.

5 Conclusion

In this paper, we presented a calibration method for improving depth calculations of eye gaze in virtual environments. Though this process becomes increasingly difficult as a user's gaze rays become more parallel at great distances, our initial results suggest that depth calibration can be effective for distances over two meters. This can be used to help disambiguate line-of-site eye tracking for use in educational or training environments where determining the correct gaze target can provide insights into user behavior, performance, and techniques that use gaze to classify or categorize scene context.

Acknowledgments

This work was funded in part by NSF grant #2223035.

References

[1] C. Armbrüster, M. Wolter, T. Kuhlen, W. Spijkers, and B. Fimm. Depth perception in virtual reality: distance estimations in peri-and extrapersonal space. *Cyberpsychology & Behavior*, 11(1):9–15, 2008.

[2] B. Krajancich, P. Kellnhofer, and G. Wetzstein. Optimizing depth perception in virtual and augmented reality through gaze-contingent stereo rendering. *ACM Transactions on Graphics (TOG)*, 39(6):1–10, 2020.

[3] Y. Lee, C. Shin, A. Plopski, Y. Itoh, T. Piumsomboon, A. Dey, G. Lee, S. Kim, and M. Billinghurst. Estimating gaze depth using multi-layer perceptron. In *2017 International Symposium on Ubiquitous Virtual Reality (ISUVR)*, pp. 26–29. IEEE, 2017.

[4] T. Toyama, J. Orlosky, D. Sonntag, and K. Kiyokawa. A natural interface for multi-focal plane head mounted displays using 3d gaze. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pp. 25–32, 2014.

[5] S. Wibirama and K. Hamamoto. A geometric model for measuring depth perception in immersive virtual environment. In *SIGCHI Proceedings of The 10th Asia Pacific Conference on Computer Human Interaction*, pp. 325–330, 2012.