Estimating Gaze Depth using Multi-Layer Perceptron

Youngho Lee[1,5], Choonsung Shin[2], Alexander Plopski[3], Yuta Itoh[4], Thammathip Piumsomboon[1], Arindam Dey[1], Gun Lee[1], Seungwon Kim[1], Mark Billinghurst[1]
[1]The Empathic Computing Lab., University of South Australia, Australia
[2]VR/AR Research Center, KETI, South Korea
[3]Interactive Media Design Lab, NAIST, Japan
[4]Interactive Media lab., Keio University, Japan
[5]UVR Lab., Mokpo National University, South Korea
youngho@ce.mokpo.ac.kr, cshin@keti.re.kr, plopski@is.naist.jp, itoh@imlab.ics.keio.ac.jp,
Thammathip.Piumsomboon@unisa.edu.au, arindam.dey@unisa.edu.au, gun.lee@unisa.edu.au, seungkimkr@gmail.com,
mark.billinghurst@unisa.edu

***Abstract*— In this paper we describe a new method for determining gaze depth in a head mounted eye-tracker. Eye-trackers are being incorporated into head mounted displays (HMDs), and eye-gaze is being used for interaction in Virtual and Augmented Reality. For some interaction methods, it is important to accurately measure the x- and y-direction of the eye-gaze and especially the focal depth information. Generally, eye tracking technology has a high accuracy in x- and y-directions, but not in depth. We used a binocular gaze tracker with two eye cameras, and the gaze vector was input to an MLP neural network for training and estimation. For the performance evaluation, data was obtained from 13 people gazing at fixed points at distances from 1m to 5m. The gaze classification into fixed distances produced an average classification error of nearly 10%, and an average error distance of 0.42m. This is sufficient for some Augmented Reality applications, but more research is needed to provide an estimate of a user's gaze moving in continuous space.**

Keywords*— *Eye-gaze, 3D gaze, Machine Learning, Augmented Reality, Head-mounted display

I. INTRODUCTION

Rapid commercialization of Virtual Reality (VR) and Augmented Reality (AR) technologies have led to the recent emergence of various types of head mounted displays (HMDs). HMDs such as the HTC VIVE [1] or SONY PlayStationVR [2] immerse users into a virtual environment. On the other hand, AR-focused HMDs overlay computer graphics onto the real world, either by placing graphics onto camera images of the real world (in a video see-through display (VST-HMD)) or by projecting the graphics directly into the user's field-of-view (in an optical see-through display (OST-HMDs)). Optical see-through HMDs, like the Microsoft HoloLens [3], are ideal devices for AR as they are non-obstructive, and do not block the view of the real world. OST-HMDs are used in a wide variety of scenarios, such as remote collaboration[4] and training [5].

Some VR HMDs are beginning to be developed with eye-tracking technology integrated into them. The FOVE [6] is a VR HMD that combines a virtual reality display with an eye-tracker. Companies such as SMI [7], Tobii [8], and Pupil-labs [9] have also developed eye-trackers that can be combined into existing AR and VR HMDs. Using this technology, researchers have begun to use eye-gaze as an input modality for VR [10] and AR applications [11] and to explore new interaction methods. For example, using eye-gaze in HMDs for extracting points of interest [12], or sharing of the user's gaze in a remote collaboration application [13]. Eye-gaze trackers have also been used for OST-HMD calibration [14], [15].

In some VR and AR applications it is sufficient to just know the 2D point of the eye-gaze, for example selecting from an on-screen menu. For other applications it is important to know the 3D location where the user is looking. For example, in a remote collaboration AR application using an OST-HMD it is important to know whether the user is looking at the virtual image on the HMD screen, or through the screen to real objects behind it. Various methods for measuring the gaze depth have been introduced. However, the traditional approach using corneal reflection is difficult to use for users wearing contact lenses or glasses because of lens distortion, and using the pupil center distance is suitable for desktop gaze trackers but not for a HMD eye-tracker. So, many researchers have applied machine learning algorithm to predict gaze depth.

In this paper, we describe a gaze depth estimator implemented using a Multi-layer Perceptron (MLP) neural network for HMD-based AR applications, and evaluate its performance. First, we acquire gaze vectors of both eyes looking at fixed distances and train the MLP for each user. Then during runtime, the gaze depth can be determined by using the gaze vector as input for the trained MLP. We evaluated how the system performed for binocular eye observations; if it can correctly distinguish between different focus distances, and how accurately it can predict the focus distance. We used data from 13 participants for the MLP classifier and the MLP regressor. The main contribution of this work is that it is the first attempt to evaluate and optimize the performance of an MLP regressor for real-time gaze depth classification and regression.

II. RELATED WORKS

Our research is based on earlier work in eye-gaze tracking, depth estimation, and classification. Eye-gaze tracking commonly requires a calibration step to account for the offset between the optical and visual axes [16]. For the calibration step users are typically asked to look at a series of points located on a planar surface, thus achieving high-quality results of about 0.5°

978-1-5386-3091-4/17 $31.00 © 2017 IEEE
DOI 10.1109/ISUVR.2017.13

accuracy when viewing objects close to the calibration plane. However, this does not provide any information about the user's focus distance.

When a user looks at a 2D screen, such as a desktop, the depth is fixed. So, the gaze direction from one eye is sufficient to determine a 2D gaze point. However, if we want to get a 3D gaze point, such as looking at a point in the real world through an OST-HMD, we need two gaze directions (see Fig 1) [17].

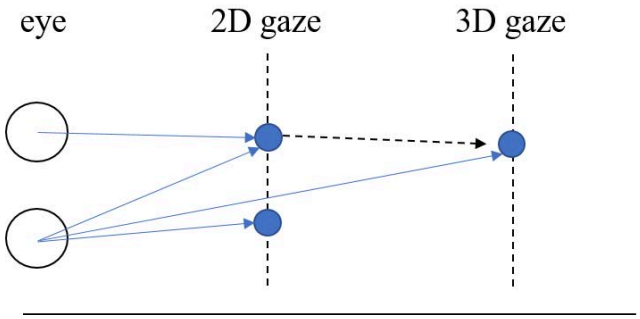

Fig 1. Gaze direction and gaze depth

The gaze depth can be obtained either through the intersection of the gaze directions, which is highly susceptible to error, or through learning of how eye features map onto corresponding distances. These features can be the pupil center distance [18], horizontal values of the gaze directions [19], or reflections of the Purkinje images[17], [20], [21].

However, the approach using corneal reflection is difficult to use for users wearing contact lenses or glasses because of a lens distortion. Using the pupil center distance for gaze depth is suitable for desktop gaze trackers but not for a HMD eye-tracker because it should calibrate two eye cameras to get the relationship between two cameras. So, machine learning algorithm has been applied to predict gaze depth. However, it is a time-consuming task to get good quality training data. Therefore, we need to evaluate estimation of gaze depth using MLP to find out proper parameters for HMD-based AR applications.

III. Technical Setup

In our research, we aim to develop a gaze depth estimation method that can be used in an OST-HMD AR interface to understand the user's focus in the real-world environment. To achieve this goal, we first measure the user's eye-gaze depth while they are looking at real-world objects at known different depths. This measurement can be used to collect training data for the MLP classification system and to test the trained system.

To do this, we used a mobile eye-tracking headset from Pupil Labs [22]. This headset is equipped with two IR eye cameras and an RGB world camera (see Fig. 2), and can be integrated into AR HMDs. The Pupil Labs eye tracking hardware is lightweight and small enough to fit on a OST-HMD such as the Epson BT-200 and HoloLens that we plan to apply our method.

The eye cameras capture images of 640x480 pixel resolution at 120Hz, and the world camera captures images of 1280x780 pixels at 60Hz. For eye-tracking, we used the Pupil v0.94 software also developed by Pupil-Labs [9]. We used the 3D eye model from the Pupil software to get the gaze normal vector. The Pupil software provides 3D mode for various parameters such as gaze normal vector, pupil center, pupil diameter, etc and streams the parameters over a network in real-time. The gaze normal vector is a three-dimensional vector indicating the direction of the eye's line of sight. For the MLP classification, we used the scikit-learn library [23] that is well known for its applications in classification and regression. We set it as a default setting such as the number of hidden layer = 100, alpha = 1e-5.

Fig. 2. Pupil-Labs eye-gaze tracker.

For calibration and training, we measured the users' gaze while they were sitting on a chair and looking at a circle on a board fixed to a tripod (See Fig. 3). We set the distances between a user and the target image as 1 m, 2 m, 3 m, 4 m, and 5 meters.

In the experiment, the user wore an eye-tracker without a HMD, because the glass of the HMD can distort the line of sight. Since the glass used for each HMD has different curvature, it would be difficult to obtain consistent data.

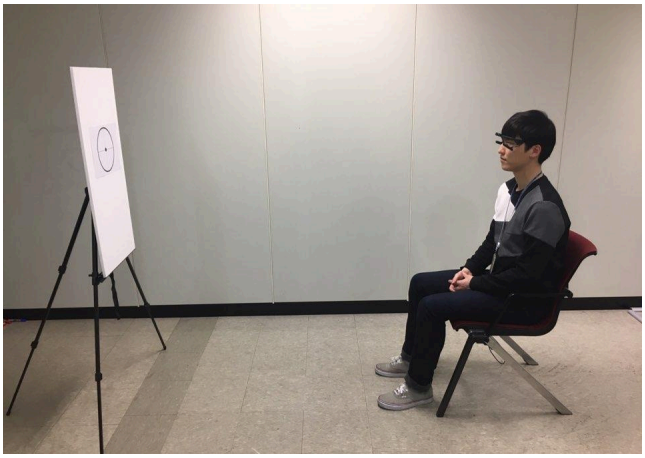

Fig. 3. A user who wears eye-gaze tracker sees an object in the distance.

IV. System Training and Evaluation

We recruited 13 participants (5 female and 8 male) for eye-tracking data acquisition and testing. The users were between 23 and 33 years old (mean = 28, standard deviation (SD) = 3.23), and all had normal eye sight. We refer to the participants as p0 to p12. During the data capture, users wore the Pupil labs eye

tracker and observed a target object at five pre-defined distances. Before measuring the gaze vector, we ran the normal Pupil Labs calibration process for each subject. For each user, we recorded the gaze normal vector of the user's eyes and the focus distance, as shown in Fig. 3. We randomly selected 1000 samples from each distance of 1m, 2m, 3m, 4m, 5m that were used for training the MLP classifier. The remaining data was left for evaluating the prediction. The data for training and prediction were obtained as shown in Table 1, where NST is the total number of samples for training, and NSP the number of samples for predicting.

TABLE 1. PARTICIPANTS AND DATA SET.

User	p0	p1	p2	p3	p4	p5	p6
NST	5000	5000	5000	5000	5000	5000	5000
NSP	3961	3734	4339	7039	4782	3988	4763
User	**p7**	**p8**	**p9**	**p10**	**p11**	**p12**	
NST	5000	5000	5000	5000	5000	5000	
NSP	4159	4197	4009	4335	4394	5125	

[a] NST: number of samples for training, NSP: number of samples for prediction

The obtained data was used to classify the gaze data at different focus depths, or to estimate the focus depth as a continuous distance value. For classification, we used an MLP classifier and an MLP regressor class in the sci-kit learn library. The MLP classifier classifies input values into discrete categories, while the MLP regressor estimates the predicted continuous value by fitting a regression line to the input values. We evaluated the prediction results by calculating the accuracy of each method. Fig. 4 shows the results of the MLP classification and Fig. 5 shows the results of the MLP regression.

The classification error rate of each participant was calculated according to the formula in equation (1). We counted the number of samples that have the same target and predicted values. The error rate was obtained by dividing the summation by the total number of samples (N) for prediction. Thus, 0% is the case where all predictions are correct. As shown in Fig. 4, the error rate of the classification varied from 0 % to 30% among the participants, although for most of the participants (10 out of 13) it was less than 15%. The median was 6.86%. The overall mean classification error rate was 9.92%, with a SD of 10.55, which indicates the method had a 90% average prediction accuracy.

$$error\ rate = \frac{100}{N}\sum_{i=1}^{N} 1 \ \ (if\ target = prediction) \qquad (1)$$

We measured the average error distance of the predicted focus depth using MLP regression according to the formula in equation (2). Given N samples, the accuracy of the regression is given as

$$error\ distance = \frac{1}{N}\sum_{i=1}^{N}\|d_i - \bar{d}_i\|, \qquad (2)$$

where d_i is the predicted distance and $\bar{d}_i$ the ground truth for sample i. Fig. 5 shows the accuracy of the prediction using the MLP regression model. When we used the gaze normal vector from two eyes, the average of error was 0.42m, with a SD of 0.23m. The minimum is 0.19 m, and the maximum was 1.05 m. While the two errors were proportional for most of the participants, p1 and p2 had relatively higher error rate in classification compared to the error distance with regression. The participant p4 also had a high error rate for classification, but had a very high regression error.

Finally, to test how the sampling target distance affects the accuracy of the regression, we trained the MLP regressor with training data with a target distance of 1, 3, and 5 meters, and evaluated the predicted values with the same data. As shown in Fig. 6, the error distances at 1 m, 3 m, and 5 m were almost the same, but the average error distance at 2 meters and 4 meters were about 4 times worse. The predictions at 2 and 4 meters where we didn't use training data had large errors, as expected. However, in the prediction at 1, 3, and 5 meters, case B was more accurate than case A. This shows the importance of capturing training data from the target distances of interest.

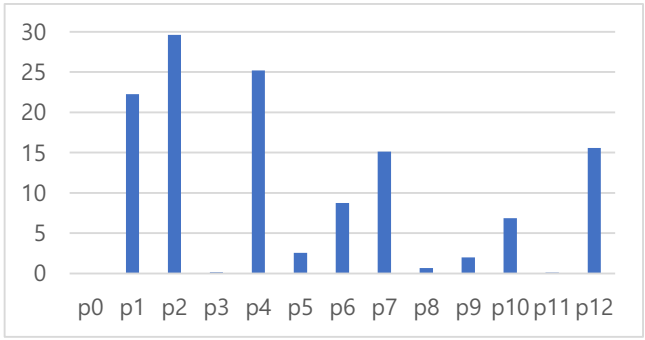

Fig. 4. Error rate of Classification (in percent).

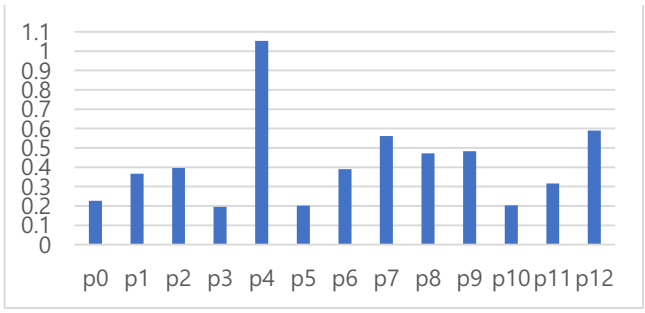

Fig. 5. Error of Regression (in meters).

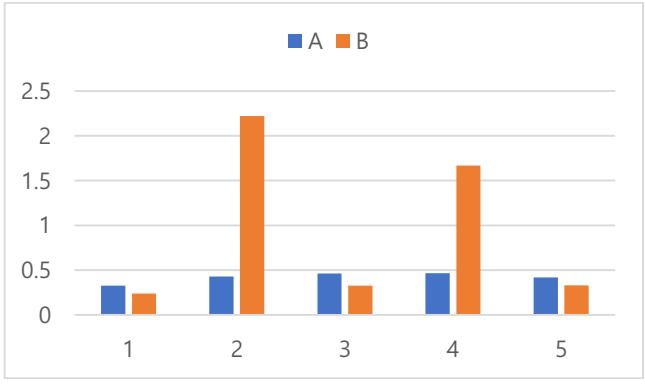

Fig. 6. Comparison of the average error distance between regression trained with the whole dataset (A) and with a subset of data at 1,3,5 meters (B).

V. CONCLUSION

In this paper, we have presented a MLP neural network based gaze depth estimator that can be used to measure the gaze depth from the measured gaze normal vector of both eyes. This estimator could be useful for developing AR user interfaces based on gaze interaction for OST-HMD. To evaluate the performance, we applied data from 13 participants to MLP classification and regression. The mean error rate was 9.92%, and the SD was 10.55. The average error distance of the regression was 0.42m, and the SD was 0.23. The results indicate that our method would be sufficient and useful for applications that need coarse (approximately 1 meter) depth estimation of the user's gaze. For example, the Epson BT-200 has a focal length of 2.5m, so this method could be ideal for identifying if the user's focus of attention was on the BT-200's virtual image, or on objects in the real world at a closer distance.

However, there are some limitations with our approach. In our experiment, we used the default 3D eye model in Pupil v.094. If we use a more accurate 3D eye model, measured from the user themselves, the accuracy should be increased. The current system only works for discrete depth distances, and when we trained the data with two-meter intervals, the error distance was more than 2 meters. So, we need to find out how to optimize the MLP regression model with more granular distance intervals., especially for applications requiring continuous gaze depth estimation in continuous space with finer accuracy. This could include collecting a data set from finer target distance levels, and also from a larger number of participants. Using various eye tracking measurements other than the gaze normal vector, such pupil center distance, or horizontal values of 2D gaze direction could be alternative approaches worth to investigate further.

Several questions remain, and there are opportunities for future work. In our experiments, users focused on objects in front of them. As users are likely to rotate their eyes while exploring the surroundings, we plan to investigate how different viewing directions could impact the calibration results. We also need to consider how saliency can help track the focus depth. We could improve the accuracy if we turned the machine learning algorithm and used different parameters. We could apply Deep Nerual Network (DNN) instead of MLP for higher accuracy. Finally, we also plan to investigate gaze depth estimation within arm's reach, a common occurrence in AR tasks.

ACKNOWLEDGEMENTS

This research is funded in part by Grant-in-Aid for Young Scientists (B), #17K12726 from Japan Society for the Promotion of Science (JSPS), Japan, and in part by the Development of Medical 3D Printing Technology based on ICT.

REFERENCES

[1] HTC Corporation, "HTC Vive." [Online]. Available: http://www.htcvr.com/. [Accessed: 20-Apr-2017].

[2] "PlayStation VR." [Online]. Available: https://www.playstation.com/en-au/explore/playstation-vr/.

[3] "Microsoft HoloLens." [Online]. Available: https://www.microsoft.com/en-au/hololens. [Accessed: 20-Apr-2017].

[4] S. Fanello, S. O. C. Rhemann, M. Dou, V. Tankovich, C. Loop, and P. Chou, "Holoportation : Virtual 3D Teleportation in Real-time," *CHI*, pp. 741–754, 2016.

[5] B. Schwald, B. Schwald, B. DeLaval, and B. DeLaval, "An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context," *11th Int. Conf. Cent. Eur. Comput. Graph. Vis. Comput. Vis.*, pp. 425–432, 2003.

[6] "FOVE." [Online]. Available: https://www.getfove.com/. [Accessed: 20-Apr-2017].

[7] "SensoMotoric Instruments (SMI)." [Online]. Available: https://www.smivision.com/. [Accessed: 20-Apr-2017].

[8] "Tobii." [Online]. Available: https://www.tobii.com/. [Accessed: 20-Apr-2017].

[9] "Pupil-labs." [Online]. Available: https://github.com/pupil-labs/pupil. [Accessed: 20-Apr-2017].

[10] T. Piumsomboon, G. Lee, and R. W. Lindeman, "Exploring Natural Eye-gaze-based Interaction for Immersive Virtual Reality," in *3D User Interfaces (3DUI)*, 2017, pp. 36–39.

[11] H. M. Park, S. H. Lee, and J. S. Choi, "Wearable augmented reality system using gaze interaction," in *Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, (ISMAR 2008)*, 2008, pp. 175–176.

[12] B. Steichen, G. Carenini, and C. Conati, "User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities," in *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, 2013, pp. 317–328.

[13] K. Gupta, G. A. Lee, and M. Billinghurst, "Do you see what i see? the effect of gaze tracking on task space remote collaboration," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 11, pp. 2413–2422, 2016.

[14] A. Plopski, Y. Itoh, C. Nitschke, K. Kiyokawa, G. Klinker, and H. Takemura, "Corneal-Imaging Calibration for Optical See-Through Head-Mounted Displays," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 4, pp. 481–490, 2015.

[15] Y. Itoh and G. Klinker, "Interaction-Free Calibration for Optical See-Through Head-Mounted Displays based on 3D Eye Localization," in *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, 2014, pp. 75–82.

[16] K. R. Moser and J. E. Swan, "Evaluating optical see-through head-mounted display calibration via frustum visualization," *Virtual Real. (VR), 2015 IEEE*, p. 371, 2015.

[17] J. W. Lee, C. W. Cho, K. Y. Shin, E. C. Lee, and K. R. Park, "3D gaze tracking method using Purkinje images on eye optical model and pupil," *Opt. Lasers Eng.*, vol. 50, no. 5, pp. 736–751, 2012.

[18] J. Ki and Y. M. Kwon, "3D gaze estimation and interaction," in *2008 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, 3DTV-CON 2008 Proceedings*, 2008, pp. 373–376.

[19] T. Toyama, J. Orlosky, D. Sonntag, and K. Kiyokawa, "A natural interface for multi-focal plane head mounted displays using 3D gaze," *Proc. 12th Int. Work. Conf. Adv. Vis. Interfaces (AVI 2014)*, vol. 2, pp. 25–32, 2014.

[20] Y. Itoh, K. Kiyokawa, T. Amano, and M. Sugimoto, "Monocular Focus Estimation Method for a Freely-Orienting Eye using Purkinje-Sanson Images," in *Virtual Reality (VR), 2017 IEEE*, 2017.

[21] V. V Krishnan, D. Shirachi, and L. Stark, "Dynamic measures of vergence accommodation.," *Am. J. Optom. Physiol. Opt.*, vol. 54, no. 7, pp. 470–3, 1977.

[22] M. Kassner, W. Patera, and A. Bulling, "Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction," *Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Adjun. Publ.*, pp. 1151–1160, 2014.

[23] "Sci-kit learn." [Online]. Available: http://scikit-learn.org/stable/. [Accessed: 20-Apr-2017].