Eye Gesture in a Mixed Reality Environment

Almoctar Hassoumi and Christophe Hurter
French Civil Aviation University, ENAC, Avenue Edouard Belin, Toulouse, France

Keywords: Eye-movement, Interaction, Eye Tracking, Smooth Pursuit, Mixed Reality, Accessibility.

Abstract: Using a simple approach, we demonstrate that eye gestures could provide a highly accurate interaction modality in a mixed reality environment. Such interaction has been proposed for desktop and mobile devices. Recently, Gaze gesture has gained a special interest in Human-Computer Interaction and granted new interaction possibilities, particularly for accessibility. We introduce a new approach to investigate how gaze tracking technologies could help people with ALS or other motor impairments to interact with computing devices. In this paper, we propose a touch-free, eye movement based entry mechanism for mixed reality environments that can be used without any prior calibration. We evaluate the usability of the system with 7 participants, describe the implementation of the method and discuss its advantages over traditional input modalities.

1 INTRODUCTION

The standard hand gesture interactions in a mixed reality environment have been successfully used in a great number of applications (Chaconas and Hllerer, 2018; Piumsomboon et al., 2013), for example in virtual text entry (Figure 1). Yet, despite its effectiveness, many questions still persist. For example, how could we extend the interactions for accessibility. In addition, the input methods proposed in traditional systems are tedious, uncomfortable and often suffer from spatial positioning accuracy (Kytö et al., 2018). Not surprisingly, people with body weaknesses, poor coordination, lack of muscle control or motor impairments could not rely on these conventional hand gestures to communicate with computing devices. For these persons, other means or input are required. Generally, the eye muscles are not affected and could be used for interacting with systems (Zhang et al., 2017). In this work, we focus on a touch-free gesture interaction that builds on smooth pursuit eye movement. Smooth pursuit is a bodily function which allows maintaining a moving object in the fovea. The advantage of this eye movement is that users can perform it voluntarily in contrast to other types of eye movements, i.e., saccade and fixation (Collewijn and Tamminga, 1984). For example, blinking has been proposed as an interaction technique (Mistry et al., 2010). However, human often blinks subconsciously in order to protect the eyes from external irritants or spread tears across the cornea. To implement our approach, we leverage two simple modules. The user pupil center location and the user interface that draws the trajectories of the moving stimuli. Pupil images are captured using a Pupil Labs Eye camera (Kassner et al., 2014). The Hololens serves as the mixed reality device. Therefore, the user interface is displayed in the Hololens field of view (FOV). Let $\mathbf{p}_i \in \mathbb{R}^2$ be the *x,y-coordinates* of the *i-th* pupil center position in the camera frame I. Then the vector $P = (\mathbf{p}_1^\intercal, \mathbf{p}_2^\intercal, \ldots, \mathbf{p}_n^\intercal)^\top \in \mathbb{R}^{2n}$ denotes the pupil center positions in I during a smooth pursuit movement. Notice that we do not use the world camera of the eye tracker and no prior calibration is performed. We will comment shortly on the pupil center detection algorithm. The second module is the evolution of the moving stimuli with time. We will only consider the case of a constant speed $\mathbf{v} = const$. However, the speed of the moving stimuli could be accelerated or decelerated. The Pearson's correlation is used to calculate the correlation between the pupil center locations and the targets' stimuli positions (Velloso et al., 2018). The results show that our approach is not affected by geometrical transformations (scaling, rotation and translation), at least, when the trajectories are in the field of view of the user. To illustrate our approach, we investigate the specific case of PIN entry. PINs are traditionally entered using key pressing or touch input. In a mixed reality environment, a virtual keyboard appears in the FOV and is used as an input modality. People with motor disabilities are often helped by an assistant, even for entering a password in a system. This reduces considerably their privacy.

Hassoumi, A. and Hurter, C.
Eye Gesture in a Mixed Reality Environment.
DOI: 10.5220/0007684001830187
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 183-187
ISBN: 978-989-758-354-4

2 RELATED WORKS

This work builds upon recent studies in eye movement research. The use of eye movement in Human-Computer interfaces systems such as PDAs, ATMs, smartphones, and computers has been well studied (Feit et al., 2017). Smooth pursuit, scanpath, saccades, and vestibulo-ocular reflex are some of the common ways to use gaze gestures in order to interact with a system. Recently, there has been a great variety of cheap eye-tracking systems that enable estimating user gaze position accurately (Kassner et al., 2014). Eye gaze interactions have been proposed as a reliable input modality (Zhang et al., 2017), especially for people with motor impairments. The Dwell method which allows selecting a target after a predefined time is one of the most used methods (Mott et al., 2017). However, this approach requires a prior calibration session where the user fixates on a series of stimuli placed at different locations (Santini et al., 2017; Hassoumi et al., 2018). The most accurate systems require a 9-points calibration before the gaze direction is accurately estimated (Kassner et al., 2014). In this work, we investigate a novel calibration-free approach that leverages the potential of smooth pursuit eye movement to select a target by following its movement. In addition, the dwell approach is limited by the time threshold. For example, if the threshold is defined for 200 ms, the user cannot fixate on a target for more than 200 ms, otherwise, a selection is triggered. Recently smooth pursuit eye movement has allowed a calibration-free gaze-based interaction. In SmoothMoves, Esteves et al. (2015) computed the correlation between targets on-screen movements and user's head movement for selection. Delamare et al. (2017) proposed *G*3, a system for selecting different tasks based on the relative movements of the eyes. Orbits (Esteves et al., 2015) and Pa*th*Word (Almoctar et al., 2018) allowed selecting a target by matching its movement. Subsequently, a great number of applications using smooth pursuit have been proposed. See (Esteves et al., 2015) for a review.

3 IMPLEMENTATION

We use a Microsoft Hololens with a Pupil Labs Eye tracker. The device is equipped with one eye camera [1]. The computer vision algorithms used to detect and track the pupil center positions reduced the frame rate by 5%. A C# desktop software was built using the EmguCv 3.1 [2] on an XPS 15 9530 Dell Laptop

[1]Sampling rate: 120 Hz, resolution: 640×480 pixels

[2]An OpenCV 3.1 wrapper for C#

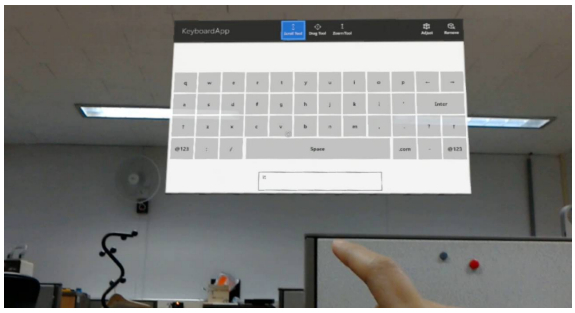

Figure 1: A keyboard interface in a mixed reality environment. Selection is made with Air Tap and Bloom gestures.

Figure 2: Our proposed PIN entry interface that uses user relative smooth pursuit eye movement for selection.

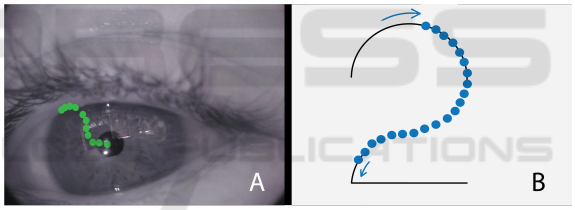

Figure 3: A user trying to select the digit ②. A-A user eye under infrared light, captured by the camera. See how the pupil is darker compared to other features of the eye. B-The digit being selected.

64 bits with an Intel(R) Core(TM) I7-4712HQ CPU 2.30GHz, 4 core(s), 8 processes, 16GB of Random Access Memory, 2GB swapping Memory.

3.1 Overview of the Method

Digits provide a useful and simple way to enter a password in a system. The method has been efficiently implemented on mobile devices to protect users against shoulder surfing and smudge attacks (Almoctar et al., 2018). We examine the method in a mixed reality environment. Numbers from zero to nine are drawn on the user interface using simple mathematical formulas (Circles and Lines equation only, see Figure 5). For instance, the digit ③ consists of two half circles and the digit ② is drawn using a three-quarter circle, another quarter circle and a segment between

two points (Figure 5). The remaining digits are drawn similarly. A moving stimulus (blue circle in Figure 2) is displayed on each digit and moves along the trajectory defined by the digit. Therefore, whenever the user needs to select a digit, he carefully follows the moving stimuli (Figure 3B) that moves on the digit's shape. While following the stimulus, the user pupil center moves accordingly and implicitly draws the shape of the digit. The $2D$ points representing the pupil trajectory $P = (\mathbf{p}_1^\intercal, \mathbf{p}_2^\intercal, \ldots, \mathbf{p}_n^\intercal)^\top \in \mathbb{R}^{2n}$ are compared against every digit in order to select the best match. The digit which points are more correlated in both **x** and **y**-axes, is likely to be the selected number. However, to avoid false activation and subsequent errors, the correlations must exceed a predefined threshold. On a scale of -1 to 1, we set the threshold to 0.82 based on a pilot study with participants. Remarkably, since the stimuli are moving constantly, the probability to start following it, at the start of its trajectory is very low. In most cases, the user will start following the stimulus after it has already started its movement. However, the simple mathematical shapes used to draw the digits allow obtaining a unique representation for every digit. Figure 4 below shows the representation of the digits in the **x**-axis and **y**-axis separately. Notice how the shapes are different from each other in both axes separately.

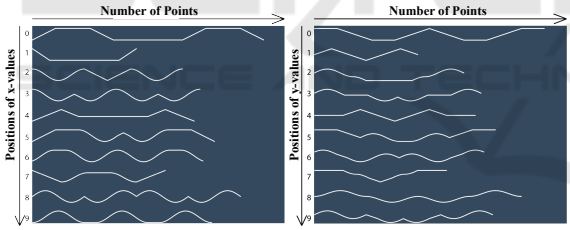

Figure 4: Representation of the digit from zero to nine in **x** (left) and **y**-axis (right).

The lengths of each digit in **x** and **y**-axes are different, that is, the number of points used to draw each digit is different. For instance, the digit ① is drawn with fewer points than the digit ⑧. The unique form of the shapes on 1-*dimension* (**x**- or **y**-axis only) is sufficient to obtain accurate results, however, in order to make the algorithm more robust, we used both axes, that is, the correlation in both x and y axes are calculated.

3.2 Interaction and Metric

Figure 2 shows the user interface of the proposed approach. The user selects a single digit by following the blue stimulus moving on its shape using their eye. For example, inserting the PIN ②①⑧⑦, begins by following the blue stimulus moving on the shape

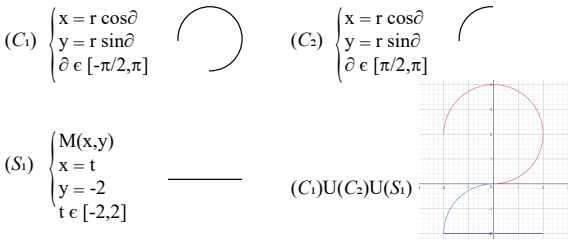

Figure 5: Composition of the digit ② using simple mathematical formulas. (C_1) represents a three-quarter circle, (C_2) a quarter of a second circle and (S_1) is the segment representing the bottom line of the digit.

defined by the digit ② in the user interface (Figure 3). Thereafter, the subsequent digits are selected similarly. Since the blue circle moves gradually at a constant velocity, the pupil positions of the user change accordingly in the eye camera imaging frame. The positions of the blue circles on all digits along with the positions of the pupil centers are stored for further processing. A mathematical measurement that gives the strength of the linear association between two sets of data is computed. This measure is the Pearson product-moment correlation coefficient (Vidal et al., 2013), named in honor of the English statistician Karl Pearson (1857-1936). Previous work used this metric, and to the best of our knowledge, Vidal et al. (2013) were the first to initiate this approach for eye tracking interaction. Examples of high (Figure 6) and low (Figure 7) $pupil - target\ stimulus$ correlations are shown below.

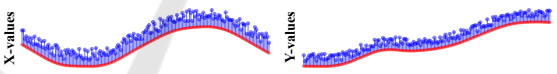

Figure 6: Illustration of a high similarity between pupil center positions (*blue circles*) and a moving stimulus position (*red circles*).

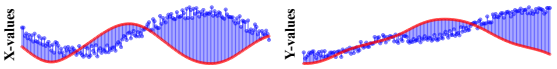

Figure 7: Illustration of a dissimilarity between pupil center positions (*blue circles*) and a moving stimulus position (*red circles*).

3.3 Pupil Detection

Our system uses the eye camera of a Pupil Labs Eye tracker. The camera is set up with an Infrared passing filter. A Near-Infrared LED illuminator is located in the immediate vicinity of the camera to illuminate the eye which conducts to corneal reflections in the subjects eye image. The presence of the visible light is circumvented and it becomes easier to separate the pupil from the iris. The pupil appears then as a darker

circular blob in the eye image.

Initially, the pupil center of the subject is detected and tracked by the infrared camera. An accurate pupil center detection is essential in this prototype. The pupil detection algorithm implemented in this study locates the features of the dark pupil present in the IR illuminated eye camera frame. The algorithm is implemented so that the user can move their head freely, thus, we do not use pupil corneal reflection to compensate small head movements. A 640×480 frame is grabbed from the IR illuminated eye camera. The image pixels color are thus converted from 3 channels RGB color space to 1-channel gray intensity value. The grayscale image is, afterward, used for automatic pupil area detection. We call this part *automatic thresholding*: starting with a user-defined threshold value $T = 21$ chosen experimentally, thresholding is used to create a binary image. T must be comprised between 0 and 255. Thresholding consists of replacing each pixel of a grayscale image into black or white. The pixel which has a gray intensity value smaller than the defined threshold (I(i,j) $<$ T) is transformed into black and the pixel having a gray value intensity greater than the threshold (I(i,j) $>$ T) is transformed into black. Since People have different pupil darkness, the user is allowed to define a range of black pixels that will define their pupil pixels. By default, we set a range r =[2000-4000] (chosen empirically). The algorithm checks if the number of black pixels is included in that range. If so, the next step of the pupil detection process is executed, otherwise, the defined threshold value is incremented and the algorithm checks again if the number of black pixels is included in that range. The process is repeated again until the number of black pixels is included in the range.

However, it is important to note that choosing a high value for the range's maximum value may lead to an increase or number of false black pixel appertaining to the pupil. Choosing a small value for the range's minimum may lead to getting small pupil area, thus providing an inaccurate pupil center. If a pupil area is found, the algorithm detects closed contours in the thresholded image Using (Fitzgibbon et al., 1999) algorithm (Figure 8), the ellipse that best fits, in a least-square sense, each contour found in the thresholded image is detected (step 4 of Figure 8). The center is then saved. The convex Hull (Sklansky, 1982) of the points representing each contour is found. The Convex Hull that has the highest isometric quotient, i.e. the best circularity is considered as the one representing the pupil. Its corresponding best fit ellipse center is stored as the pupil center for this frame.

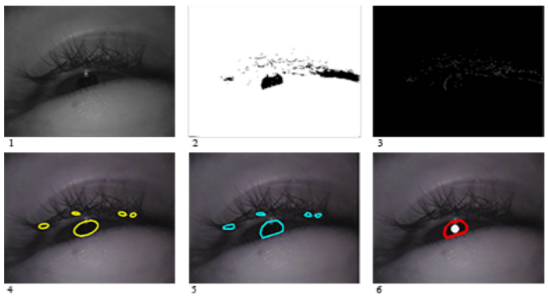

Figure 8: Illustration of pupil tracking and detection pipeline.

4 PILOT STUDY RESULTS

We have investigated the perceived task load of the technique using a NASA-TLX questionnaire. 7 healthy participants (4 females) were recruited for the experiment. a 5-minutes acquaintance period was given to the participants in order to be familiar with the interaction. The tasks were counterbalanced to reduce learning effects. The primary results indicated that, among the feature tested, the *frustration* gave the lowest work load as shown in Figure 9, ($\mu = 4.16$, $\sigma = 3.81$). The highest load was obtained for *Effort* ($\mu = 11.66$, $\sigma = 3.72$). During the individual interview with the participants, we found that they indicated a high effort load because this is their first smooth pursuit interaction attempt. The remaining perceived loads were as follows: $\mu_{Mental} = 9.0$ ($\sigma_{Mental} = 4.33$), $\mu_{Physical} = 8.33$ ($\sigma_{Physical} = 6.15$), $\mu_{Temporal} = 8.83$ ($\sigma_{Temporal} = 4.21$), $\mu_{Performance} = 7.5$ ($\sigma_{Mental} = 4.41$). It can be noted that the *effort* could be reduced by changing different parameters of the algorithm, for example, the size or the orientation of the targets. In addition, the speed could be modified for each user.

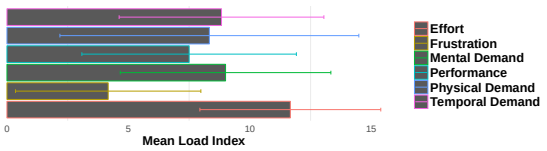

Figure 9: Results of the pilot study evaluation.

5 LIMITATIONS

Our approach suffers from common known issues. For example, Lighting conditions, head poses, and eyelashes reduce the accuracy of the pupil detection. Moreover, the digits must have an acceptable size, otherwise, the pupil center positions will appear to be

static or unchanged. This problem may also occur when the user interface is displayed far from the user FOV. We tested our approach with 7 participants and although the initial feedbacks are positive and encouraging, we plan to conduct an experiment in real-world scenarios and report the results in follow-up studies. As we are improving the system, we are implementing algorithms and methods to detect the digits faster. In addition, the algorithm was tested with user with full mobility. Additional evaluation will help understand the effectiveness of the approach in real-world scenarios with subjects restricted in their motor skills.

6 CONCLUSION

In this paper, we presented a novel eye-based interaction entry that uses smooth pursuit eye movement. A key point of this paradigm is that a selection implies that the user has followed a moving target, thus eliminating the Midas touch problem. Other applications may benefit from this interaction technique, for example entering a flight level in a virtual Air Traffic Control Simulator. Future work will explore digits recognition time and investigate the potential of this method on alphanumeric characters.

REFERENCES

Almoctar, H., Irani, P., Peysakhovich, V., and Hurter, C. (2018). Path word: A multimodal password entry method for ad-hoc authentication based on digits' shape and smooth pursuit eye movements. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, pages 268–277, New York, NY, USA. ACM.

Chaconas, N. and Hllerer, T. (2018). An evaluation of bimanual gestures on the microsoft hololens. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1–8.

Collewijn, H. and Tamminga, E. P. (1984). Human smooth and saccadic eye movements during voluntary pursuit of different target motions on different backgrounds. *The Journal of Physiology*, 351(1):217–250.

Esteves, A., Velloso, E., Bulling, A., and Gellersen, H. (2015). Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology*, UIST '15, pages 457–466, New York, NY, USA. ACM.

Feit, A. M., Williams, S., Toledo, A., Paradiso, A., Kulkarni, H., Kane, S., and Morris, M. R. (2017). Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1118–1130, New York, NY, USA. ACM.

Fitzgibbon, A., Pilu, M., and Fisher, R. B. (1999). Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480.

Hassoumi, A., Peysakhovich, V., and Hurter, C. (2018). Uncertainty visualization of gaze estimation to support operator-controlled calibration. *Journal of Eye Movement Research*, 10(5).

Kassner, M., Patera, W., and Bulling, A. (2014). Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 1151–1160, New York, NY, USA. ACM.

Kytö, M., Ens, B., Piumsomboon, T., Lee, G. A., and Billinghurst, M. (2018). Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 81:1–81:14, New York, NY, USA. ACM.

Mistry, P., Ishii, K., Inami, M., and Igarashi, T. (2010). Blinkbot: Look at, blink and move. In *Adjunct Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 397–398, New York, NY, USA. ACM.

Mott, M. E., Williams, S., Wobbrock, J. O., and Morris, M. R. (2017). Improving dwell-based gaze typing with dynamic, cascading dwell times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2558–2570, New York, NY, USA. ACM.

Piumsomboon, T., Clark, A., Billinghurst, M., and Cockburn, A. (2013). User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 955–960, New York, NY, USA. ACM.

Santini, T., Fuhl, W., and Kasneci, E. (2017). Calibme: Fast and unsupervised eye tracker calibration for gaze-based pervasive human-computer interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2594–2605, New York, NY, USA. ACM.

Sklansky, J. (1982). Finding the convex hull of a simple polygon. *Pattern Recogn. Lett.*, 1(2):79–83.

Velloso, E., Coutinho, F. L., Kurauchi, A., and Morimoto, C. H. (2018). Circular orbits detection for gaze interaction using 2d correlation and profile matching algorithms. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pages 25:1–25:9, New York, NY, USA. ACM.

Vidal, M., Bulling, A., and Gellersen, H. (2013). Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 439–448. ACM.

Zhang, X., Kulkarni, H., and Morris, M. R. (2017). Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2878–2889, New York, NY, USA. ACM.