EYELASSO: REAL-WORLD OBJECT SELECTION USING GAZE-BASED GESTURES

Yuan-Fu Shao (*邵元輔*), *Chiuan Wang* (*王權*), *Chiou-Shann Fuh* (*傅楸善*)

Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan

ABSTRACT

Selecting objects in real-world settings are currently difficult to automate and require significant manual effort. We propose a gaze-based gesture approach using wearable eye trackers. However, achieving effective gaze-based selection of real-world object has several challenges, such as the issue of Double Role and Midas touch. Prior studies required explicit manual activation/de-activation to confirm the user intention, which impede fast and continuous interaction. We introduce EyeLasso - a fast gaze-based selection technique that allows users to select the target they see with only a single Lasso gaze gesture, without requiring additional manual input.

Keywords *Eye Tracking; Machine Learning; Gesture; OpenCV; GrabCut; Segmentation; Human-computer interaction.*

1. INTRODUCTION

Recent advances in wearable eye-trackers have made it possible to track what users are looking at in real time in the real world. This approach enables users to perform gaze-based marking gestures to indicate an object of interest for selection. A user can then perform a search on the selected object or zoom in on the selected target for picture taking purposes with no need to tweak the zoom-in button. However, many researchers believe that robust recognizers for such gaze-based input systems may be extremely difficult to implement and currently unfeasible [4, 6]. A typical issue of using gaze input is that the users will run into problems of the double role of gaze for observation and control. This means that the gaze is not only used as a marking input, but users will typically also perform saccadic eye movements to orient themselves in the scene, which results in poor target selection performance. In our work, we expect to show with our recognition system, we are not only able to give a satisfactory detection rate for our Lasso gaze gesture marking input technique without any extra control, but also improve the target selection rate by using machine learning (ML) techniques to compensate for deficiencies in the gaze sample.

To pick both the targets for our experiment and gaze gestures, we chose 10 different ordinary objects in our daily lives (5 indoor and 5 outdoor objects). Then we picked the Lasso selection technique for our testing gesture because it is a more unusual and less often used eye movement as compared with Rectangular Marquee gesture.

We separated this task into two parts. First of all, to help us understand how real users actually perform given gestures, we set up a 6-person study to perform the Lasso gaze gesture on given targets while using an eye tracker (Tobii Glasses 2). We categorized the types of selection patterns that real users would make while performing the gaze gesture. Then we applied a computer vision technique plus our unique solution dealing with ambiguous gestures to increase the selection performance. Second, to determine whether Lasso gaze selection requires activation/deactivation and to prevent issues like Double Role or Midas touch, we conducted another 6-person study to test whether or not our algorithm is able to recognize the activation gaze gesture without any extra input.

Our paper contributes the followings. Firstly, we develop a novel algorithm which is able to automatically locate the target of interest, even with vague gaze data, by adding a segmentation algorithm for prediction and error correction. Secondly, we use a unique gaze gesture detection system to mark the start and end of the activation gesture without providing extra input controls such as tap, nod and so on. Thirdly, we examine three well-known machine learning techniques, Artificial Neural Network, Support Vector Machine and Random Forest, and compare the performance of gesture detection respectively. Finally, we categorize a set of raw gaze movements into 6 categories and show how we overcome the challenges.

2. RELATED WORKS

2.1. Eye-based Interaction

The potential lying in eye-based interaction has been discussed [4, 12] in various studies having shown gaze interaction faster than selection with a mouse [11, 13, 17, 18]. Also with many ongoing developments in eye tracking systems allowing precisely pervasive and unobtrusive gaze interaction in everyday contexts [1, 2, 16], EyeLasso helps interpret eye movements into useful data for searching for objects in real world situations.

2.2. Gaze Gestures

Gaze gestures are defined as eye movements performed within a time interval [3], with which users can conduct visual searches on real-world objects around them [5]. [8] worked on the gaze behavior definitions and formula of the agreement score. [5] revealed that simple gaze gestures make unintended interaction occur more often. In comparison, complex gaze gestures consisting of multiple simple gaze gestures generate solid results of interaction; yet, complex gestures are not as intuitive as simple gaze gestures for user to perform [7, 19]. We have conducted a thorough user field trials study and decided on an ideal gesture to be put in use.

2.3. Object Recognition

Numerous recognition methods were brought up with or without gesture support. The core idea of these techniques is to distinguish user attention from unconscious eye movements [4, 7, 15], which have been causing issues such as the Midas touch problem [6]. [14] uses a fixation point to guide the object recognition framework successfully. [3] has built an autonomous eye tracker based on EOG instead of common systems using video cameras and can recognize eye gestures from EOG signals.

These methods perform well with the external support they chose while EyeLasso comes up with a standalone solution that also solves the problem perfectly.

2.4. Image Editing

Fixtag extends automatic fixation-identification to include tagging fixations to premeasure Regions-of-Interest (ROIs) [9]. GrabCut helps a user to extract desired object while a user only has to drag a rectangle loosely around an object [10]. EyeLasso can be extended with these sorts of image editing algorithms to form an effective, intuitive way of searching with gaze gestures.

3. IMAGE SEGMENTATION

We first describe the second part of the system: Image segmentation with gaze points. The input is a group of gaze points extracted from gaze activation and the picture taken by the eye-tracker. We tested the data of our previous gaze gesture studies and found that we can categorize the gaze behavior into 6 types: perfect, half-done, irregular, distraction, scatter, and un-focus. The gaze point patterns of each type are defined in Figure 1. Unintentional eye movement and the low accuracy of our gaze cause the diverse behavior of gaze point. In order to deal with the differences between various behaviors of gaze, we need a complicated algorithm to do image segmentation. We divided the image segmentation system into two main phases, named window and spotlight phase. In the window phase, we take a gaze sample with a duration lasting between 1 to 5 seconds as an input, and a window is determined by the outwardly expanded values of both the minimum and maximum of x and y values among the gaze point set to make sure the object being wrapped inside (see Figure 2a). In the spotlight phase, the system filters out the rest of the objects not of interest by applying a well-known object detection algorithm GrabCut with our gaze sample (see Figure 2b). We used a portion of inner gaze points (points close to their center mass) and inwardly contracted these points to create a foreground brush. Then we used a portion of outer gaze points and outwardly expanded these points to create a background brush. The algorithm is shown in Figure 3.

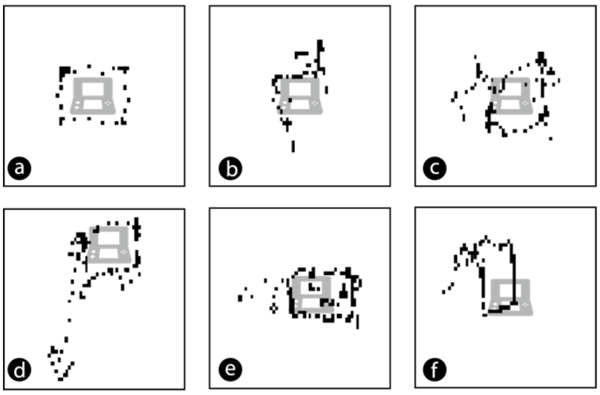

Figure 1: a) Perfect - a target is perfectly selected by Lasso, b) Half Done - Lasso is incomplete, c) Irregular - not seen like a Lasso, d) Distraction - distracted by other contents while lassoing, e) Scatter - a complete Lasso with other unnecessary gaze points overlapping, and f) Unfocused - Lasso does not completely select the target.

To analyze the challenges of gaze gestures we observed, we categorized the 6 gaze gestures. Traditional gaze point based recognition technology may find difficulty in locating the target because of the irregular gaze behavior even when the users themselves believe they have performed perfectly. Our system only uses gaze points as support for better object detection, and if we detect more than 1 object, the target with most gaze points nearby will be selected.

To understand how well our gaze assists image segmentation, we then conducted a user field trials study with 6 users to examine the performance. In this experiment, each participant was asked to sit in front of 5 given targets placed on a table in a room and then walked to a predefined outdoor route where they would see other 5 selected targets. The choices of targets were collected from our preliminary study, and for each target, the participants were asked to perform the Lasso gaze gestures three times on each target while wearing the Tobii Glasses 2.

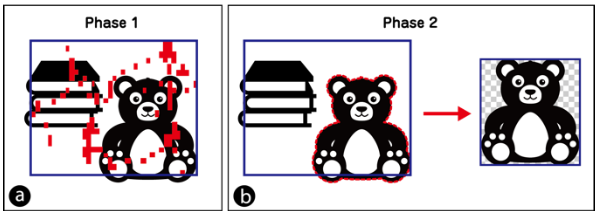

Figure 2: a) Window phase - roughly draw a square region based on the farthest gaze point, b) Spotlight phase - use GrabCut for the target detection (Target with most gaze points nearby will be picked), then redraw the region.

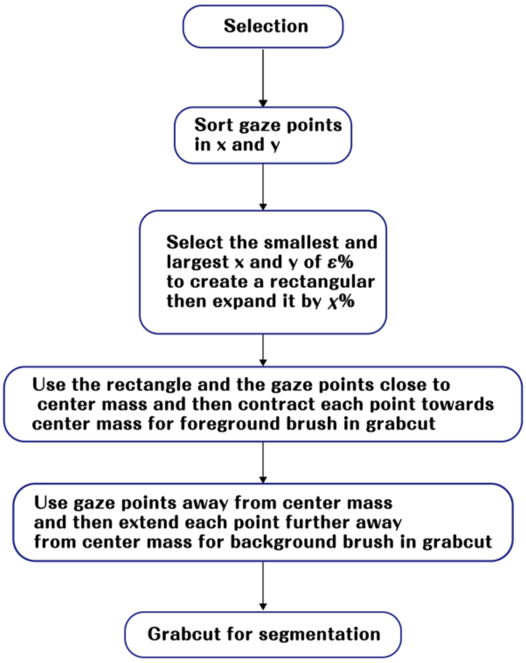
Selection
Sort gaze points in x and y
Select the smallest and largest x and y of ε% to create a rectangular then expand it by χ%
Use the rectangle and the gaze points close to center mass and then contract each point towards center mass for foreground brush in grabcut
Use gaze points away from center mass and then extend each point further away from center mass for background brush in grabcut
Grabcut for segmentation

Figure 3: Processing flow of our 2-phase image segmentation system.

The 5 objects indoors we selected: a textbook, a lego idol, a bottle of beer, a cookie box, and a portrait on screen. 5 outdoor objects: a signboard, flowers, a car, a bag, and an image on the capsule toy. We used these images and sensor values to tune the parameters in our algorithm. We truncated the smallest and largest 5% and expanded the edge 200% to create the rectangle area for GrabCut. Then, 60% of inner points were shrunk by 50% from the mass center to create foreground brush, and 60% of outer points were expanded by 50% from the mass center to create background brush.

The user study is depicted in figure 4. The user wore the eye-tracker. Our algorithm extracts the target object by image took from the eye tracker. We found that outdoor objects are easily segmented than that of indoors because the appearance of outdoor objects usually does not resemble its background colors. From figure 5c and 5d, our objects are surrounded by many other things from the view of the user. Therefore, the results of our algorithm fails to segment the target from other objects nearby.

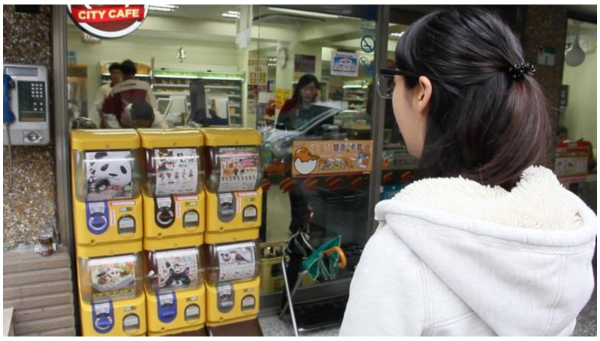

Figure 4: a) a user wore the eye-tracker and selected a textbook indoor b) a user wore the eye-tracker and selected the image on the capsule toy machine.

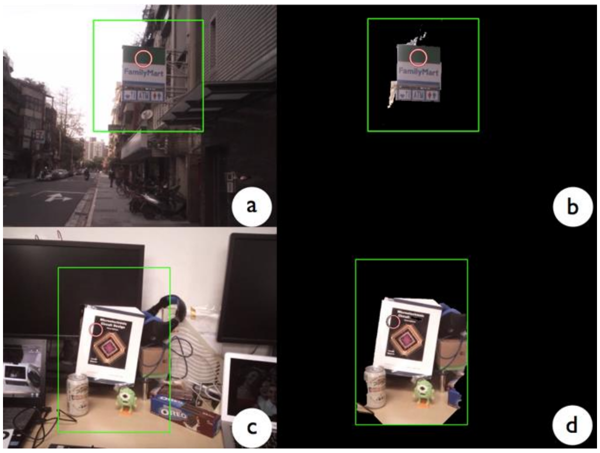

Figure 5: a) the image of an isolated object used as selection target, b) the segmented image by our algorithm (Note that the green rectangular is created in selection phase 1), c) the image selection target overlapped by various objects and d) the image after our algorithm fails to segment the individual target object

4. GAZE SEARCH ACTIVATION

The issue of when and how to activate the gaze gesture detector is a major problem. Since user gaze is always on, how to identify the beginning and end of an eye-selection gesture is the first problem we need to solve. That is, a system requires some mechanisms to trigger the eye tracker to avoid users from unintentionally performing actions. Prior studies solve this issue by adding some explicit manual controls to confirm user intention. However, this kind of solution may impede fast and continuous interaction. Our goal is to allow the user to

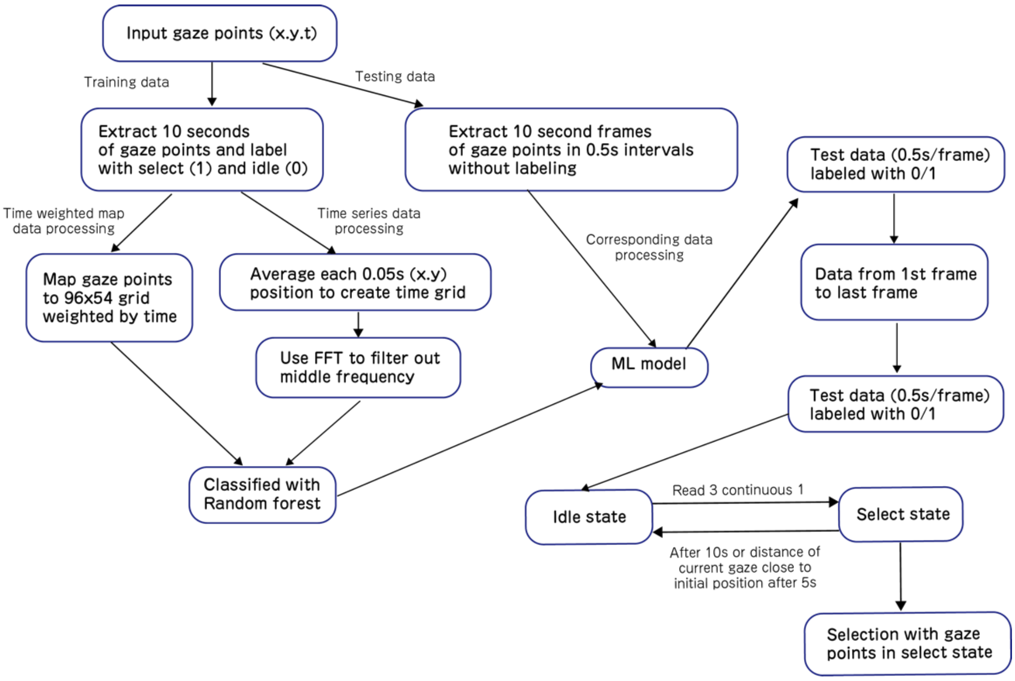
Input gaze points (x.y.t)
Training data
Testing data
Extract 10 seconds of gaze points and label with select (1) and idle (0)
Extract 10 second frames of gaze points in 0.5s intervals without labeling
Test data (0.5s/frame) labeled with 0/1
Time weighted map data processing
Time series data processing
Corresponding data processing
Map gaze points to 96x54 grid weighted by time
Average each 0.05s (x.y) position to create time grid
Data from 1st frame to last frame
Use FFT to filter out middle frequency
ML model
Test data (0.5s/frame) labeled with 0/1
Classified with Random forest
Read 3 continuous 1
Idle state
Select state
After 10s or distance of current gaze close to initial position after 5s
Selection with gaze points in select state

Figure 7: Lasso gesture detection with Random Forest learning. Note: Every tick must have a gaze point. The closest gaze point is picked for missing gaze data.

select a real target with a simple gaze gesture while minimizing the effort to trigger the selection agent.

Our system flow is shown in figure 7 and consists mainly a ML technique to detect the activation gesture. To develop a ML model requires training data. Gaze points are time weighted and subsequently mapped to a lower resolution grid image (96 x 54) for quicker computational speed and an elimination of noise. With time-weighted data, the motion of eye movement is obtained and becomes potentially recognizable by a computer. On the other hand, gaze points undergo Fast Fourier Transformation (FFT) to remove middle frequencies caused by distractions of eye movement. Thus, ML model is trained from both the time-weighted map data and time series data to assist our system for eye movement recognition.

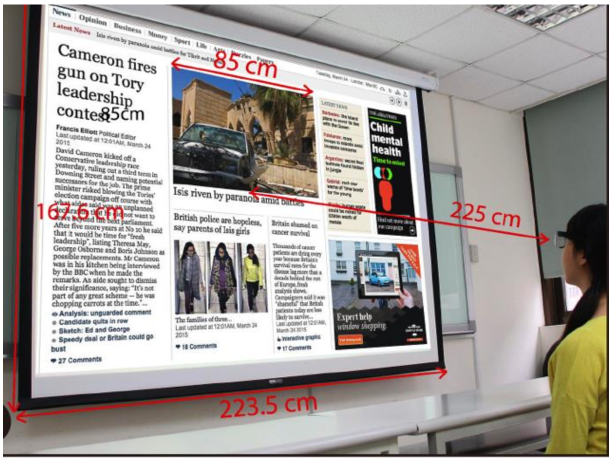

Figure 6: Lab study - image selection (visual angle 0.37)

In order to collect training data, we first tested the data under a lab control environment. Participants were asked to sit in front of a 110 inch (223.5cm * 167.6cm) projector screen in a space that distance between 150 and 250 centimeters, which means participants can freely move their body and head. A digital article with images was shown on the screen, and we asked users to select our assigned image target (85cm in width) by performing the Lasso gaze gesture.

The procedure of the experiment was that 1) participant is first asked to continuously perform the Lasso gaze gesture 10 times on the assigned target, 2) the participant is then asked to randomly make a total of 5 Lasso selections using gaze on the same target within 2 minutes, 3) then continuously making 10 gaze selection again on the same object and finally 4) 8 minutes freely looking around without intentionally performing the selection gaze gesture. The design of this procedure is to avoid the issue of eye tiring, since we were interested in collecting as much training data as possible. In addition, each participant was asked to go through this procedure twice, thus for each participant, we did obtain 40 continuous Lasso selections and 16 minutes of non-selection gaze. To cover different time length of gaze gestures, we define 4 seconds to be the duration of each of our input frame. After labeling the video for training, we transfer the gaze tracks into a time weighted gaze points map and a (x, y) time series. Then we process the data to filter out unintentional gaze, and plug it into ML model. After several tests we select random forest as our ML model. Next, we use this model for testing our continuous gaze input. Our system has a time-sliding window size of 4 seconds and determines in every 0.5 seconds whether the activation gesture is performed. To avoid misdetection, only 3 contiguous time-sliding windows being determined as the activation gesture will change the state machine of the system to “select state” and trigger the image segmentation. The system returns to “idle state” after the time-sliding window size (4 seconds) or after 2 seconds if the user completes the gesture in less time. The image segmentation process is based on the gaze points during “select state”.

5. RESULTS

With the ground truth of the actual gesture length, Random Forest outperforms the others with 90% of correct labeling and only has 1 unintended selection. 5 users out of 6 were successfully classified without any unintended selection or false negative. However, there is a user that has 6 false negatives because the participant has very different selection behavior between the training phase and testing phase (see Table 1). This indicates that a bad training phase or incongruous gaze between training and testing period will cause our machine learning algorithm be error-prone. Moreover, Figure 7 shows the complete procedure of our algorithm in our system.

Table 1: Random Forest outperforms the other learning algorithms

ML Method	*Unintended Selection*	*False Negative*
Random Forest	0.17	0.5 out of 5
Support Vector Machine	0.83	0.83 out of 5
Artificial Neural Network	1	1.33 out of 5

We found that users have poor gaze performance when selecting outdoor targets. This could be caused by various factors, such as 1) social acceptability issues, for example when users may feel awkward by wearing the eye tracking glasses while walking on the street, or 2) device limitation, such as poor calibration for different targets, deviation by wearing eye contacts and attraction of other objects. However, during both the window and spotlight period in our second implementation, we are able to dismiss noisiest data and relocate the target of interest to improve the rate of target selection.

6. CONCLUSION AND FUTURE WORK

EyeLasso allows users to select the target they see with a single Lasso gaze gesture without requiring additional manual input. As gesture recognition system, EyeLasso is able to label most samples correctly (avg. 80% accuracy) for complete lasso gestures. As a target selection system, EyeLasso using our two-phase algorithm is able to perfectly locate the target of interest.

For future work, we are interested in merging the two systems and building a prototype for real time selection as well as combine it with an search engine for instant lookup purposes, such as selecting a restaurants sign to look up its ratings and reviews. We will test performance and run benchmarks to compare our approach with traditional search paradigms, such as pulling out a smartphone and entering keywords for looking up information.

REFERENCES

[1] Tobii Technology - world leader in eye tracking and gaze interaction. http://www.tobii.com/.

[2] Bednarik, R., Vrzakova, H., and Hradis, M. What do you want to do next: a novel approach for intent prediction in gaze-based interaction. In *Proceedings of the symposium on eye tracking research and applications*, ACM (2012), 83–90.

[3] Bulling, A., Roggen, D., and Tro ̈ster, G. It’s in your eyes: towards context-awareness and mobile hci using wearable eog goggles. In *Proceedings of the 10th international conference on Ubiquitous computing*, ACM (2008), 84–93.

[4] Istance, H., Bates, R., Hyrskykari, A., and Vickers, S. Snap clutch, a moded approach to solving the midas touch problem. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, ACM (2008), 221–228.

[5] Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., and Vickers, S. Designing gaze gestures for gaming: an investigation of performance. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ACM (2010), 323–330.

[6] Jacob, R. J. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems (TOIS)* 9, 2 (1991), 152–169.

[7] Kieras, D. E., and Hornof, A. J. Towards accurate and practical predictive models of active-vision-based visual search. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, ACM (2014), 3875–3884.

[8] Mardanbegi, D., Hansen, D. W., and Pederson, T. Eye-based head gestures. In *Proceedings of the symposium on eye tracking research and applications*, ACM (2012), 139–146.

[9] Munn, S. M., and Pelz, J. B. Fixtag: An algorithm for identifying and tagging fixations to simplify the analysis of data collected by portable eye trackers. *ACM Transactions on Applied Perception (TAP)* 6, 3 (2009), 16.

[10] Rother, C., Kolmogorov, V., and Blake, A. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 309–314.

[11] Sibert, L. E., and Jacob, R. J. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM (2000), 281–288.

[12] Sibert, L. E., Templeman, J. N., and Jacob, R. J. Evaluation and analysis of eye gaze interaction. Tech. rep., DTIC Document, 2001.

[13] Sˇpakov, O., Isokoski, P., and Majaranta, P. Lookand lean: accurate head-assisted eye pointing. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM (2014), 35–42.

[14] Toyama, T., Kieninger, T., Shafait, F., and Dengel, A. Gaze guided object recognition using a head-mounted eye tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM (2012), 91–98.

[15] Tseng, Y.-C., and Howes, A. The adaptation of visual search strategy to expected information gain. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2008), 1075–1084.

[16] Turner, J., Bulling, A., and Gellersen, H. Extending the visual field of a head-mounted eye tracker for pervasive eye-based interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM (2012), 269–272.

[17] Vertegaal, R. A fitts law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the 10th international conference on Multimodal interfaces*, ACM (2008), 241–248.

[18] Ware, C., and Mikaelian, H. H. An evaluation of an eye tracker as a device for computer input2. In *ACM SIGCHI Bulletin*, vol. 17, ACM (1987), 183–188.

[19] Wobbrock, J. O., Aung, H. H., Rothrock, B., and Myers, B. A. Maximizing the guessability of symbolic input. In *CHI'05 extended abstracts on Human Factors in Computing Systems*, ACM (2005), 1869–1872.