ISSN 1392 – 124X INFORMACINĖS TECHNOLOGIJOS IR VALDYMAS, 2004, Nr. 1(30)

# AN ALGORITHM TO COUNTERACT EYE JITTER IN GAZE-CONTROLLED INTERFACES

**Darius Miniotas, Oleg Špakov**

*Unit for Computer-Human Interaction*
*University of Tampere, FIN-33014 Tampere, Finland*

**Abstract**. One of the major concerns in developing efficient gaze-controlled user interfaces is inherent eye jitter, which presents a key limitation on the pointing accuracy achievable with an eye tracker. To counteract eye jitter, we developed a grab-and-hold algorithm. The efficiency of the algorithm was tested experimentally in a target acquisition task. Results suggest that the grab-and-hold algorithm affords a dramatic 57% reduction in error rates overall. The reduction is as much as 68% for targets subtending 0.35 degrees of visual angle. However, there is a cost which surfaces as a slight increase in movement time (10%). These findings indicate that measures like the algorithm presented here have the potential of making eye gaze a more suitable input modality.

## 1. Indroduction

Gaze-controlled interfaces constitute the area of user interfaces intended primarily for the community of motor-disabled people. To access computers and communication devices, eye trackers are used as input devices. Currently, there is a large variety of commercially available eye trackers having different technical specifications and spanning over relatively broad price range.

Despite this variety, one common feature about recent eye-tracking equipment is the limitation of pointing accuracy to one degree of visual angle [2]. This is dictated by the size of the fovea – the portion of the retina providing high acuity vision of the object of current interest. As a result, targets must subtend at least one degree of visual angle for sufficiently reliable pointing with an eye tracker.

Furthermore, the size of the fovea is not the only factor limiting the practical accuracy of eye tracking. Even when the eyes appear steady on the surface during observation of a particular object, it fact they do not stay still. Instead, they make micro movements to allow visual perception of the scene. This phenomenon is known as inherent eye jitter.

Practically, eye jitter implies that during a single gaze, or, in the terminology of eye movement literature – a *fixation* – only a fraction of the gaze points belonging to this fixation will enter the target (see Figure 1 for illustration). Obviously, this presents a challenge for target acquisition.

We developed a software for an eye tracker, called a grab-and-hold algorithm (GHA), aimed to counteract the negative impact of eye jitter on eye-based pointing performance. The mechanism of the algorithm is quite simple as described in subsection 2.3.

To evaluate the algorithm, an experiment was conducted in which onscreen targets had to be selected using eye gaze. Pointing performance was assessed by measurement of movement time and error rate.

## 2. Method

### 2.1. Participants

Twelve un-paid volunteers (8 male, 4 female) participated in the experiment. All were students at a local university and had normal or corrected vision. Four of the participants had prior experience with eye tracking technology.

### 2.2. Apparatus

The experiment was conducted on a Pentium III 500 MHz PC with a 17-inch monitor with a resolution of 1024 x 768. A head-mounted eye tracking system *EyeLink*™ from SensoMotoric Instruments served as the input device. The participant PC was connected to another PC (Celeron 466 MHz) for analysis of the captured eye images.

### 2.3. Procedure

Participants were seated at a viewing distance of approximately 70 cm. The experiment used a simple point-select task (Figure 1). At the onset of each trial, a home box appeared on the screen. It was visible to participants as a 20-by-20-pixel square (thick outline in Figure 1). The actual size of the home box, however, was 120 x 120 pixels (thin outline). The expansion in motor space facilitated homing through increased tolerance to instabilities in calibration of the eye tracker. On the other hand, making only the central portion of the home box visible ensured bringing the gaze closer to the center of the box.

Upon fixating on the home box for one second, a rectangular target appeared in the peripheral field of view. Participants were instructed to look at the target as quickly as possible (timing started), and fixate upon it until selection (timing ended). A window of three seconds was given to complete a trial. If no selection occurred within three seconds, a TNC-type (trial not completed) error was recorded. Then, the next trial followed.

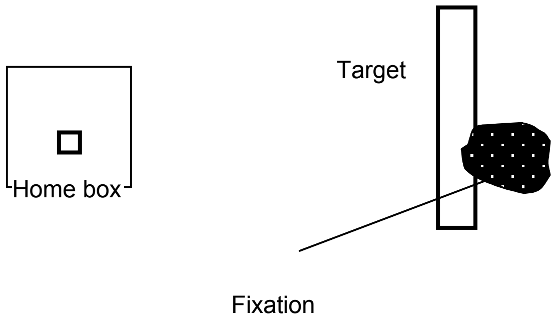


**Figure 1.** Experimental task

For the experimental condition where the grab-and-hold algorithm was turned on, target acquisition proceeded as follows. Upon appearance of the target, there is a settle-down period of 200 ms during which the gaze is expected to land in the target area and stay there. Then, the algorithm filters the gaze points until the first sample inside the expanded target area is logged. When this occurs, the target is highlighted and the selection timer triggered. The selection timer counts down a dwell time (*DT*) interval that was set at 1250 ms.

The target is selected irrespective of the actual location of the gaze point at the moment of the *DT* expiry, provided no interruptions (i.e., interspersing saccades) occurred in the fixation throughout the *DT* interval. Thus, the gaze is virtually held on the target once it is "grabbed". This way some intelligence is added to the interpretation of the eye tracker data: the gaze point is allowed to deviate from its intended destination as long as this deviation does not extend beyond the boundaries of the current fixation.

If the eye makes a saccade before the end of the *DT* countdown, however, the target is de-highlighted resetting the selection timer. Then the algorithm starts hunting for the next gaze point in the expanded target area, and the process is repeated.

Meanwhile, for the other experimental condition with the grab-and-hold algorithm turned off, the following target acquisition procedure was used. In the absence of the algorithm, the target is highlighted whenever the gaze is over it. A highlight starts the selection timer. Selection occurs only if the gaze does not leave the expanded target area for the duration of the *DT* interval. If an exit occurs during this interval, the target is de-highlighted, and the selection timer resets, starting the countdown for a new *DT*. The process is repeated until either the gaze meets the stringent no-quit criterion for target selection, or the three-second time limit expires.

### 2.4. Design

The experiment was a 2 × 4 × 3 × 3 × 3 × 3 repeated measures factorial design. The factors and levels were as follows:

| | |
|---|---|
| GHA | on, off |
| Direction | left, right, up, down |
| Distance (*D*) | 128, 256, 512 pixels |
| Width (*W*) | 12, 24, 36 pixels |
| Expansion Factor (*EF*) | 1, 2, 3 |
| Trial | 1, 2, 3 |

Although no learning effects were expected due to the highly intuitive nature of eye-gaze based pointing, participants were still randomly assigned to one of two groups. The order of presenting the GHA conditions was counterbalanced between the groups.

For each GHA condition, participants performed 12 blocks of trials (3 blocks per movement direction) in one session. The two sessions were run over consecutive days with one session lasting approximately half an hour. Each block consisted of the 27 *D-W-EF* conditions presented in random order. For each *D-W-EF* condition, 3 trials were performed. The trial for any condition, however, was not repeated within the same block, but was administered in a separate block to allow resting the eyes. Thus, a block consisted of 27 trials. The conditions above combined with 12 subjects resulted in 7776 total trials in the experiment.

The 27 *D-W-EF* conditions were chosen to cover a range of task difficulties spanning 1.13 to 5.45 bits, according to Fitts' index of difficulty [1]:

$$ID = \log_2 (D/W+1)$$

The dependent measures were movement time (*MT*) and error rate (*ER*).

## 3. Results

The grand means on the two dependent measures were 1805 ms for movement time and 18.2% for error rate. The main effects and interactions on each dependent measure are presented below.

### 3.1. Speed

As can be seen in Figure 2, disengagement of the algorithm reduced selection times by 9% on average, the main effect being significant ($F_{1,11}$ = 27.6, $p$ < .0001). There is thus an additional time cost attributed to the inner workings of the algorithm. The main effect of *EF* on *MT* was also significant ($F_{2,22}$ = 141.8, $p$ < .0001), as was the algorithm × *EF* interaction ($F_{2,22}$ = 35.9, $p$ < .05).

### 3.2. Accuracy

Figure 3 shows the effect of the algorithm on the error rate. In the algorithm absent condition, the error rate increased to 25.6%. It was 2.3 times higher than that obtained for the algorithm present condition (10.9%). The main effect of the algorithm on the error rate was significant ($F_{1,11}$ = 69.4, $p$ < .0001), as was the case for expansion factor ($F_{2,22}$ = 217.7, $p$ < .0001) and the algorithm x *EF* interaction ($F_{2,22}$ = 152.3, $p$ < .0001).

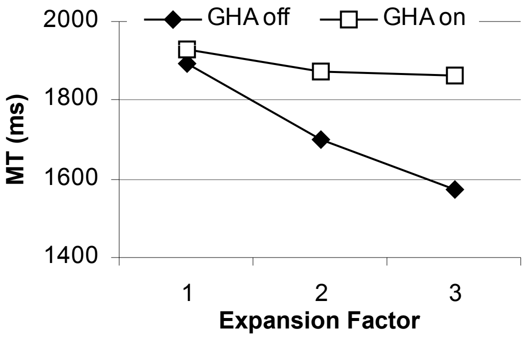


**Figure 2.** Movement time vs. expansion factor for the two conditions

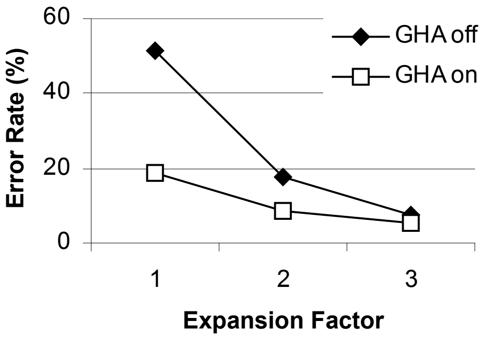


**Figure 3.** Error rate vs. expansion factor for the two conditions

The impact of the algorithm on accuracy is particularly apparent when error rates are plotted against the effective target width, i.e., *W* x *EF* (Figure 4). For the smallest target width without expansion (i.e., 12 pixels, corresponding to 0.35 degrees of visual angle), there was a 68% reduction in *ER* when the algorithm was turned on. Facilitation was also observed for effective target widths of 24 and 36 pixels, the Student-Newman-Keuls pair-wise differences being quite reliable ($p$'s < .001). Meanwhile, for effective *W* ≥ 48 pixels, the algorithm's effect was not significant.

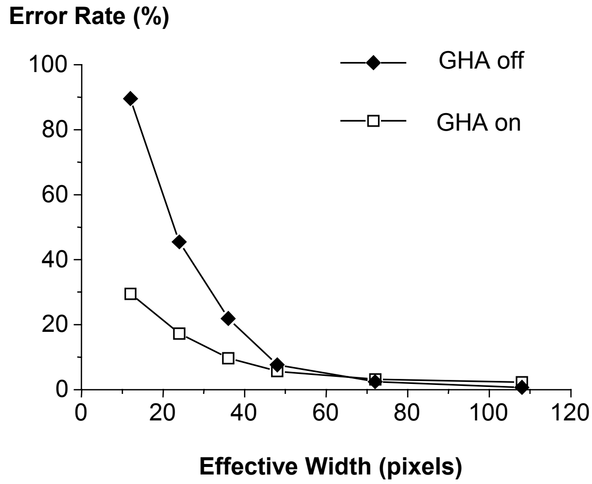


**Figure 4.** Error rate vs. effective width for the two conditions

At the effective width of 48 pixels (1.4 degrees), error rate did not exceed 8% in both the conditions. This finding is consistent with that of [4]. Employment of the grab-and-hold algorithm yields error rates under 10% even for a target as narrow as 12 pixels with a threefold expansion.

## 4. Conclusions

Our results indicate that limited accuracy of eye gaze as an input technique can be addressed by finding ways to increase tolerance to inherent eye jitter. As evidenced by the performance of our grab-and-hold algorithm, adding some intelligence to the dwell-time based selection technique can bring eye gaze input one step closer to supporting interactions with standard GUI widgets, such as scrollbars and pull-down menus. From the traditional viewpoint of eye gaze control, such targets are just too small for the interaction to be feasible. We believe, however, that novel approaches such as the algorithm presented here might help in ultimately redefining the domain of applications eye-based systems can be used in.

More work is needed before eye gaze interaction can find its way in more realistic settings involving numerous objects. A more sophisticated algorithm will be required for handling eye jitter under the constraints imposed by multiple expanding targets getting close to one another. In the future, we also intend to supplement our grab-and-hold algorithm with an eye drift correction technique similar to that suggested in [3].

### Acknowledgements

This research was funded in part by grant #103174 to Miniotas from the Academy of Finland. Support from the UCIT Graduate School for Spakov is also greatly appreciated.

## References

[1] **P.M. Fitts.** The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 1954, 381-391.

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

[3] **D.M. Stampe, E.M. Reingold.** Selection by looking: a novel computer interface and its application to psychological research. *In J.M. Findlay et al. (Eds.), Eye Movement Research*. *Elsevier Science*, 1995, 467-478.

[4] **C. Ware, H.H. Mikaelian.** An evaluation of an eye tracker as a device for computer input. *Proc. Conference on Human Factors in Computing Systems and Graphics Interfaces CHI + GI 1987*. *ACM Press*, 1987, 183-188.