Human Factors in Design, Engineering, and Computing, Vol. 159, 2024, 814–822
https://doi.org/10.54941/ahfe1005648

Testing a Motion Matching Algorithm for Gaze-Based HMI

Cosima von Uechtritz, Verena Rist, Torsten Gfesser, and Thomas E. F. Witte

FKIE, Fraunhofer Institut für Kommunikation, Informationsverarbeitung und Ergonomie Wachtberg, Germany

ABSTRACT

This study explores gaze-based interaction, focusing on object selection, particularly in dynamic environments. Traditional methods like dwell-time selection have limitations, prompting investigation into novel approaches such as motion matching. A pilot study was conducted to compare a motion matching algorithm with dwell-time selection, indicating a tendency towards faster selection times with motion matching. Workload metrics showed very similar results between selection mechanisms, but a small bias towards reduced user frustration and enhanced satisfaction using motion matching. Challenges remain, including the Midas touch problem and technical constraints of eye tracking technology, highlighting the need for further research to refine algorithms and address limitations. Despite challenges, motion matching represents progress towards making gaze-based interaction more accessible for widespread use.

Keywords: Gaze-based interaction, Motion matching, Eye tracking, Object selection, HMI

INTRODUCTION

Humans often use their gaze as a natural way of exploring the world (Valtakari et al., 2021). The gaze is directed to identify and interact with the object of interest (Valtakari et al., 2021). This basic concept has been used by eye tracking technologies to create a new interaction method, which has been widely studied in the past years (Valtakari et al., 2021; Liao et al., 2022) and is referred to as gaze-based interaction. Using the eye as an input device can create a natural user interface, which provides a more intuitive HMI in comparison to established methods such as keyboard and mouse. Gaze-based interaction offers a potential advantage for people who are unable to use their hands. This potential can also be used as a complement to a multimodal user interface (i.e., car drivers not supposed to remove their hands from the steering wheel) (Zhai, Morimoto, & Ihde, 1999). Especially, when large distances need to be covered, the eye has the ability to move very quickly compared to other parts of the body which could lead to a potential benefit in certain application fields (Zhai, Morimoto, & Ihde, 1999; Lischke et al., 2016; Fares, Fang, & Komogortsev, 2013).

Nowadays most eye tracking technologies are based on camera or infrared sensors which can detect the pupil and calculate the gaze vector. Human beings are foveated animals, which means that objects in the foveal area,

approximately around 1–2 degree are seen in high resolution, whereas objects in the periphery appear in decreased resolution (Valtakari et al., 2021). This presents a challenge for eye tracking devices to accurately calculate the human gaze position and operate with small user interfaces such as buttons and scroll bars.

In addition, the eye is a perceptual organ, and its movements are not voluntarily per se, so that voluntary movements can be hardly differentiated to unvoluntary movements. This problem is referred to as the Midas touch problem (Stellmach & Dachselt, 2012). To use eye tracking as a natural user interface (NUI), which provides an intuitive and low-demand interaction, there is a need to develop efficient selection algorithms, which overcome the Midas touch problem.

Object Selection in Eye Tracking

Selecting an object is one of the fundamental operations in HMI and is the preceding action for many following operations (i.e., moving an item that has been selected beforehand) (Liao et al., 2022). Even though object selection based on gaze-input seems to be an easy process, several solutions have been developed in the past years. Gaze-input can be either based on gaze data only or supported by manual input devices (i.e., controller, mouse). Zhai et al. (1999) presents the MAGIC-method, which uses a manual input device overtaking the gaze selection for fine selection. Other implementations of the MAGIC-method have been explored during the past years such as the MAGIC-tab method (Stellmach & Dachselt, 2012) and the Rake Cursor Method (Blanch & Ortega, 2009). However, all these methods rely on a second input device controlled by the hands. The aim of this paper is to focus on gaze-based input without a second device, enabling hands-free interaction.

The most frequent selection algorithm used with gaze-input only is the dwell-time approach, which selects an object if the gaze remains still for a certain time threshold, which is usually around 600ms (Liao et al., 2022). Liao *et al.* (2022) reported that a dwell-time of 600ms resulted in more efficient selection types than dwell times of 200ms and 1000ms. Choosing the right dwell-time is crucial to design an efficient algorithm which leads to low user *demand* but differentiates between voluntary and unvoluntary object selection. Next, to dwell-time approaches, blinking has been used to identify object selection (Kowalczyk & Sawicki, 2019). A major constrain of blinking is that it remains under unconscious control, which might lead to unnatural user interaction. Thus, blinking might not be appropriate for a natural user interface (Liao et al., 2022). Another approach identifies pre-learned eye gestures to select an object. Eye gestures present an interesting alternative for selection algorithms to prevent unintentional object selection, delivering equally good results as dwell-time interaction (Hyrskykari, Istance, & Vickers, 2012). However, eye gestures must be pre-learned and trained before using the interaction method. This study focusses on object selection which focuses on a more intuitive form of interaction without the necessity of pre-learned gestures.

Moving Object Selection in Eye Tracking

Moving object selection is an additional challenge for the creation of interaction methods. Animated visuals are found in several different applications in HMI (i. e. air traffic control, video games). Objects may have moved away from the pointer / mouse cursor before the mouse click is registered by the system (Gunn, Irani, & Anderson, 2009). Recently, algorithms using smooth pursuits, which describes the movement eyes perform when they focus onto a moving object, have been presented to facilitate object selection in moving objects (Vidal, Bulling, & Gellersen, 2013). To tackle the challenge of selecting a moving target, the correlation between eye movements with an object's trajectory is calculated. If the correlation reaches a pre-determined threshold, it is assumed that the object's motion and the eye's motion match and the target is selected. In comparison to dwell-time, which demands the user to keep the gaze unnaturally in a fixed location, using motion matching algorithms present a more natural way of interacting with moving objects. According to Vidal *et al.* (2013) motion matching is a versatile and robust technique for interaction with moving objects. It implicates independence of target size, as the object size does not matter for object selection. In addition, it facilitates the use of dynamic interfaces with animated visuals (Vidal, Bulling, & Gellersen, 2013). Motion matching algorithms prioritize tracking the overall movement of the eye rather than focusing on the precise accuracy of individual points. With the advancement of such algorithms, there may be a potential shift in the necessity for calibration. As these algorithms may require lower levels of accuracy, calibration could become unnecessary. Eye tracking devices without calibration would offer a broad range of possible applications, where the user must not be familiar with the handling of the device, such as public displays. However, next to the potential advantages there are also possible disadvantages of the motion matching algorithm. The constant movement could lead to higher fatigue levels for users in comparison to dwell-time method.

The main aim of this study is to develop a motion matching algorithm to improve the selection of moving objects. The following pilot study explores the differences between motion matching and dwell-time selection.

METHODS

The experimental setup consisted of a self-developed stimulus application running on a Windows computer and an eye tracker (Tobii Pro Fusion Eye Tracker, 250hz). The study is designed as a repeated-measures design with two experimental conditions and two prior-trainings to familiarize the participant with the two different selection algorithms. In total the participant completed the stimulus application four times: training with motion matching, experiment with motion matching, training without motion matching and experiment without motion matching. The experimental condition order was randomized to avoid a learning effect and the related training condition was conducted before each experimental

condition. After each experimental condition the participants were asked to fill in the NASA TLX questionnaire.

The eye tracker was connected to a computer (display width: 39.8cm height 22.4 and resolution: 2560x1600) and calibrated via the Tobii Eye Tracker Manager Software (ETM Version 2.6.1). Gaze data was transferred with a frequency of 250Hz via the.NET Framework provided by Tobii to an eye tracking module written in Java. Before using the eye tracking data to control the stimulus application the data were filtered in this module with a velocity threshold filter (I-VT) based on Tobii the white paper (Olsen, 2012) and a modified DBSCAN algorithm (see *I.C*). Light conditions in the experimental setup were held constant by closing all windows and using a ceiling light. Dependent on the condition either dwell-time (see *I.D*) or motion-matching algorithm (see section *I.E*) was used to select an object.

Stimulus Application

The stimulus application was written in Java (version 16.1) and JavaFx. The stimulus application includes ten waves with a duration of 24s each. Each wave consists of moving circles, with one target and multiple distractors ranging between 2–10 in a controlled-randomized order. The circles have a radius of 35 pixels, which move with a mean velocity of $mean = 30px/s$ in a controlled randomized order on the screen. All the circles are grey except for the target circle which is blue and turns rose as soon as it is selected. The participants were asked to select the blue object as fast as they can by looking at it and keep focusing on the target object with their eyes after selection. After the end of a wave all the circles disappeared before the next wave started.

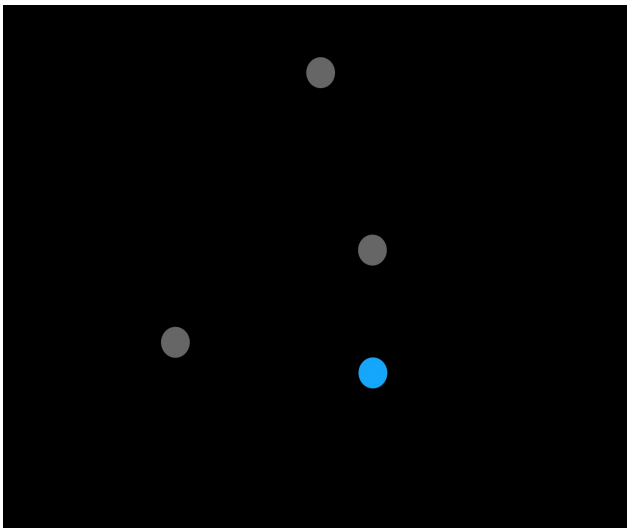

Figure 1: Stimulus application. Blue target is not yet selected.

DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise), is a popular clustering algorithm known for its ability to identify clusters of varying shapes and sizes in a dataset (Gertsiy & Botvin, 2020). Unlike traditional clustering algorithms like k-means, DBSCAN does not require the user to specify the number of clusters beforehand, making it particularly useful in scenarios where the number of clusters is not known a priori (Eraslan, Yesilada, & Harper, 2020). According to Li *et al.* (2016) a distance-based algorithm improves the accuracy of fixation and saccade calculation in oculomotor fixation identification. The modified DBSCAN has been applied together with an I-VT based filter to the data. The modified DBSCAN based on Li *et al.* (2016) redefines a core point, where a core point p contains at least min points that are within distance d to the point and these points form a consecutive subsequence of the dataset. The DBSCAN algorithm joins the fixation points up into clusters. A cluster is built if it contains at least k neighbors within distance d. The core point is used as the point of view. If a saccade is following a fixation the clusters are deleted and the process is repeated (Li et al., 2016).

Dwell Time Algorithm

The dwell-time algorithm selects an object if the gaze center is directed for a time threshold t within a buffer size d. Using the optimal selection time is crucial for an efficient algorithm, where objects are not selected too fast by unvoluntary gaze direction or too slow, which might be strenuous for users. Liao *et al.* (2022) showed that a selection time of 600ms is most efficient with a buffer size of 81px.

Motion Matching Algorithm

The implemented algorithm is based on Vidal *et al.* (2013) with some further adaptations. The basis of the algorithm is to calculate the correlation between the eye movement and the objects on the trajectory. The correlation coefficient is calculated using the Pearson's product-moment correlation method. The method returns the target with the highest correlation coefficient, which is then selected without any further user input. The coefficients are calculated isolated for the horizontal dimension x and the vertical dimension y and merged in a further step. To improve performance and efficiency of the algorithm only targets within distance $d = 200$px of the eye gaze center are evaluated for correlation. The inputs are synchronized position points in screen coordinates (x, y) for the target and the eye movement. If the correlation is higher or equal to the threshold 0.4 in both dimensions the object is added to the target list. The object with the highest mean correlation coefficient within the window size ($w = 3$) is selected. To adapt to the speed of an object, the correlation threshold for this dimension is set to zero if an object's speed is lower than 2px/s in one dimension. Otherwise, the calculated correlation coefficient would be zero and thus the target would be neglected by default.

RESULTS

In total, data of seven participants have been recorded. In total participants were measured in four conditions: training with motion matching, experiment with motion matching, training without motion matching and experiment without motion matching. The time needed to click the target was measured (time to first click now referred to as time), the target click count and the distractor click count within all four conditions. Due to the small sample size and thus low power the data was analyzed descriptive.

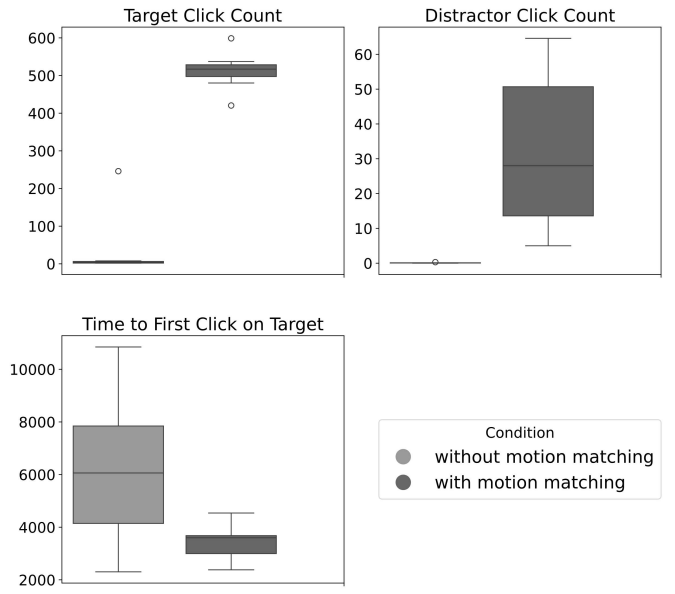

Figure 2: Boxplot between experimental conditions with motion matching algorithm and without motion matching algorithm.

The mean time to first click in the condition with motion matching is $m = 3410ms$ +/– 695 and $m = 6170ms$ +/– 3986 for the experimental condition without motion matching, indicating a faster selection time for the motion matching algorithm. The mean target click count is $m = 38$ +/– 91 for the condition without motion matching and $m = 512$ +/– 54 for the condition with motion matching, indicating a higher target click count during the experimental condition with the motion matching algorithm. The mean distractor click count is $m = 0.1$ +/– 0.14 for the condition without motion matching and $m = 32$ +/– 23 for the condition with motion matching, indicating a higher distractor click count during the experimental condition with the motion matching algorithm. In general, it can be concluded that the click count is higher for the motion matching algorithm.

In addition, to the performance variables, workload has been measured with the aid of the NASA TLX questionnaire. The NASA TLX questionnaire measures the following six items: frustration, satisfaction, time demand,

mental demand, physical demand, overall demand. The item data represents discrete data on a scale form 0–20. The data is analyzed on for each item individually (Hart, 2006).

The mean for the item mental demand is $m = 3$ +/– 1 for without motion matching algorithm and $m = 3.12$ +/– 1 for the motion matching algorithm. The physical demand results present $m = 3.3$ +/– 3.4 without motion matching and 3.6 +/– 3.3 for the motion matching condition. Next, the time demand is 1.8 +/– 2 without motion matching and $m = 2.3$ +/– 2 with motion matching. The level of frustration is $m = 6$ +/– 2.3 without motion matching and $m = 5.8$ +/– 2.7 with motion matching. The level of satisfaction is higher for the motion matching algorithm with $m = 13.5$ +/– 5.8 in comparison to a mean of $m = 10.1$ +/– 6.4 without motion matching. The overall demand is 7.4 +/– 4.5 with motion matching and $m = 5.8$ +/– 2.8 without motion matching.

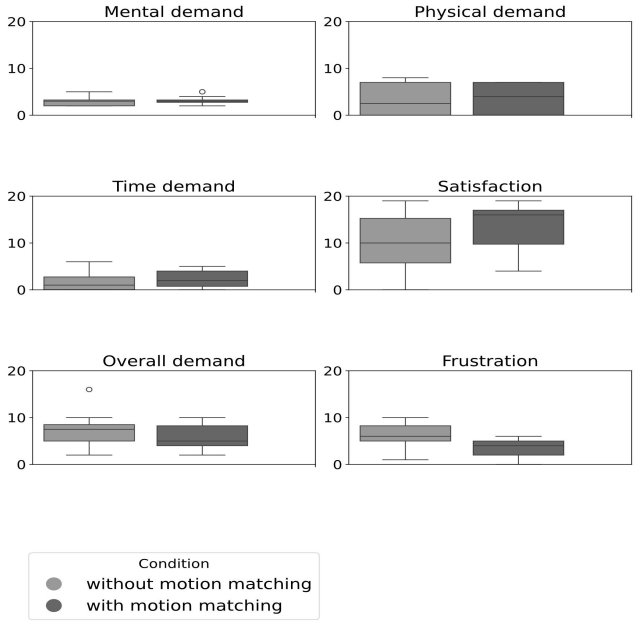

Figure 3: Boxplot for each item of the NASA TLX.

DISCUSSION

The introduction of eye tracking as a new interaction method represents an advancement in the field of natural user interfaces. Within the past years different selection methods in eye tracking interaction have been investigated. Motion matching algorithm presents one possibility which might be able to be a solution for the Midas touch problem. The main aim of this study was to explore this selection method.

The collected data shows faster selection times for the motion matching algorithm and a higher number of target clicks compared to the dwell-time selection method. This could be a first support for the formulated hypothesis

that motion matching could be the key to utilizing gaze-based interaction for moving objects. The results of this study should be interpreted with caution due to the small sample size, so further research is needed to gather more evidence for this hypothesis. A faster selection time and a higher number of target clicks, as well as a lower frustration level and a higher satisfaction level compared to the dwell-time method, indicate that the motion matching algorithm could provide a more efficient and less strenuous user interface for object selection for moving targets. Utilizing the natural smooth tracking movements of the eye could provide an advantage in eye-based interaction that reduces user workload and takes advantage of natural user interaction.

The number of clicks on distractors is higher when using the motion matching algorithm than when using the dwell-time algorithm. In general, there is a higher number of clicks with the motion matching algorithm than with the dwell-time algorithm. As participants were asked to continue to focus their gaze on the target even after the target was selected, these results are unexpected. Either the participants " truly" looked at the distractors and the algorithm correctly recognized the focus, or the klick was incorrect. If the subject " truly" looked at the target, this may have been either involuntary or voluntary. Differentiating between these movements is a well-known challenge in gaze-based interaction and is described by the Midas Touch Problem in the literature. The data collected in this study does not allow a valid distinction between involuntary and voluntary movements. A measurement method to distinguish these movements would be of great help to validly evaluate selection algorithms. A possible adjustment to improve the accuracy of the algorithm would be to increase the correlation threshold. By increasing the threshold value, fewer clicks could be triggered, allowing a clearer distinction to be made between voluntary and involuntary movements. Similar to the dwell-time algorithm, identifying the optimal threshold is crucial for the development of a powerful algorithm. If the algorithm is implemented in an application, an optimization could be made so that targets that have already been selected cannot be selected again in order to reduce the processor load. It should also be taken into account that simultaneous head and eye movements change the eye movement in such a way that it is no longer possible to calculate the correlation coefficient. The insights gained in this pilot study lead to some specific challenges that need to be addressed to improve the performance of the motion matching algorithm.

This study shows that using the smooth tracking movements of the eye could be beneficial to develop an efficient algorithm with less workload for the user. Especially in environments where the user has to interact with moving objects, this algorithm could be beneficial. Environments such as air traffic control, surveillance tasks or games with animated visuals offer interesting fields of application. Further steps are required to validate the results, for example through an experiment with a more realistic task and a higher sample size to increase statistical power. Motion matching brings gaze-based interaction a step closer to widespread application, but there are still limitations that need to be overcome. In particular, technical problems with the eye tracker and the need for calibration make it difficult to implement in everyday life. Investigating the hypothesis that motion matching algorithms

could lead to a reduction in calibration processes would be necessary to propose a solution to one of the challenges.

REFERENCES

Blanch, R., & Ortega, M. (2009). Rake cursor: Improving pointing performance with concurrent input channels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Eraslan, S., Yesilada, Y., & Harper, S. (2020). The best of both worlds! Integration of web page and eye tracking data driven approaches for automatic AOI detection. *ACM Transactions on the Web (TWEB)*, 14(1), 1–31.

Fares, R., Fang, S., & Komogortsev, O. (2013). Can we beat the mouse with MAGIC? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Gertsiy, A., & Botvin, M. (2020). Comparative analysis and research of digital image encoding algorithms. *Editor Coordinator*, 1237.

Gunn, T. J., Irani, P., & Anderson, J. (2009). An evaluation of techniques for selecting moving targets. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems* (pp. 3329–3334).

Hart, S. G. (2006). NASA-Task Load Index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*.

Hyrskykari, A., Istance, H., & Vickers, S. (2012). Gaze gestures or dwell-based interaction? In *Proceedings of the Symposium on Eye Tracking Research and Applications*.

Kowalczyk, P., & Sawicki, D. (2019). Blink and wink detection as a control tool in multimodal interaction. *Multimedia Tools and Applications*, 78, 13749–13765.

Li, B., et al. (2016). Modified DBSCAN algorithm on oculomotor fixation identification. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*.

Liao, H., Zhang, C., Zhao, W., & Dong, W. (2022). Toward gaze-based map interactions: Determining the dwell time and buffer size for the gaze-based selection of map features. *ISPRS International Journal of Geo-Information*, 11(2), 127.

Lischke, L., Schwind, V., Friedrich, K., Schmidt, A., & Henze, N. (2016). MAGIC-Pointing on large high-resolution displays. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 1706–1712).

Olsen, A. (2012). The Tobii I-VT Fixation Filter Algorithm description. Tobii. https://www.vinis.co.kr/ivt_filter.pdf

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 246–253).

Stellmach, S., & Dachselt, R. (2012). Look & touch: Gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Valtakari, N. V., et al. (2021). Eye tracking in human interaction: Possibilities and limitations. *Behavior Research Methods*, 1–17.

Vidal, M., Bulling, A., & 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*.