Smooth Pursuit Target Speeds and Trajectories

Heiko Drewes[1], **Mohamed Khamis**[1,2], **Florian Alt**[1,3]
[1]LMU Munich, Germany, {firstname.lastname}@ifi.lmu.de
[2]University of Glasgow, UK, {firstname.lastname}@glasgow.ac.uk
[3]Bundeswehr University Munich, Germany, {firstname.lastname}@unibw.de

ABSTRACT
In this paper we present an investigation of how the speed and trajectory of smooth pursuits targets impact on detection rates in gaze interfaces. Previous work optimized these values for the specific application for which smooth pursuit eye movements were employed. However, this may not always be possible. For example UI designers may want to minimize distraction caused by the stimulus, integrate it with a certain UI element (e.g., a button), or limit it to a certain area of the screen. In these cases an in-depth understanding of the interplay between speed, trajectory, and accuracy is required. To achieve this, we conducted a user study with 15 participants who had to follow targets with different speeds and on different trajectories using their gaze. We evaluated the data with respect to detectability. As a result, we obtained reasonable ranges for target speeds and demonstrate the effects of trajectory shapes. We show that slow moving targets are hard to detect by correlation and that introducing a delay improves the detection rate for fast moving targets. Our research is complemented by design rules which enable designers to implement better pursuit detectors and pursuit-based user interfaces.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI)

Author Keywords
Eye tracking; smooth pursuits; pursuit detection; trajectories

INTRODUCTION
Gaze-only interfaces offer many desirable properties such as being hygienic (nothing to touch), maintenance-free (no moving parts), and keeping the hands free. However, despite the proverbial quick eye movements and the ease of eye movements, such interfaces are slow and cumbersome for the users. In the past, gaze-only interfaces used dwell time [12] and a typical application was eye-typing for disabled users [17]. Dwelling requires an individual eye tracker calibration before use and is therefore not suited well for instant use in public space [13], such as on ATMs and ticket vending machines.

MUM '18, November 25–28, 2018, Cairo, Egypt

ISBN 978-1-4503-6594-9/18/11…$15.00
DOI: https://doi.org/10.1145/3282894.3282913

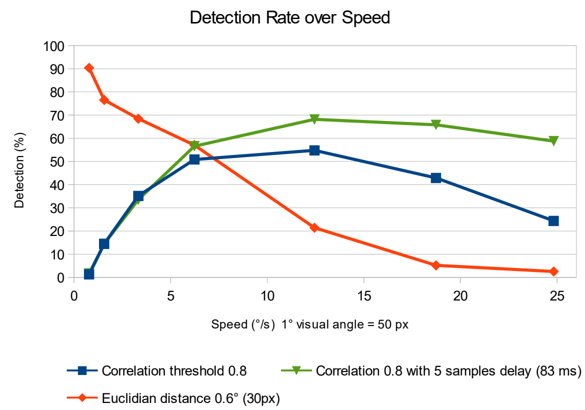

Figure 1. Detection rate over speed for correlation-based detection, correlation-based detection with pursuit delay, and Euclidian distance detection. Slow target speeds (below 4°/s) are hard to detect with the correlation method, but well detectable with the Euclidian distance method if the eye tracker is calibrated. Fast target speeds (above 10°/s) are better detectable when the correlation is calculated with a delayed target.

In 2013 Vidal et al. introduced smooth pursuit eye movements as a gaze-only interface method, called Pursuits [24]. As smooth pursuit movements are detectable from relative gaze movements they do not require calibration. In contrast to gaze gestures [5], smooth pursuits offer a target to the user and following this target with the eyes feels more natural than performing a gaze gesture. The HCI community continued research on smooth pursuits and explored applications [3, 8, 14, 19, 23, 25]. However, despite much research on pursuits, there are still many open questions that become particularly relevant as smooth pursuits find their way into novel application areas. One question is how to choose the optimal target speed and trajectory. This is valuable as designers try to integrate smooth pursuits with different user interfaces. For example, designers may want to allow selection of a button using smooth pursuit. The target could be a small circle moving along the edge of the button, hence restricting the degree of freedom regarding the target design; or designers may want to minimize distraction from the moving targets by reducing speed.

For such cases it is important to understand, how different speeds and different target trajectories influence how well pursuits eye movements can be detected. Hence, designers can make better decision as to how to design their interface.

To close this gap we developed an application which allows stimuli for the eyes to be created by displaying moving targets and simultaneously the coordinates of the targets and the user's gaze to be recorded. This allows the independent variables for a user study to be adjusted. For the evaluation of the recorded data and to calculate the dependent variables – Euclidian distance between gaze and target and correlation of target and gaze movement – we wrote an analysis tool.

We conducted a user study (N=15) where participants were asked to follow the displayed targets with their gaze. The evaluation provides reasonable speed ranges for pursuit targets and insights on detection methods which will help to design pursuit-based interfaces. In particular we could show that calculating the detection with delayed targets increases the detectability for fast moving targets. Pursuit movements on slow targets, in contrast, are hard to detect with the correlation method. Enhancements are possible with the Euclidian distance method but this requires calibration which destroys the advantage of Pursuits. Results are summarized in Figure 1.

RELATED WORK

Work on pursuit movements was done by psychologists already in the 50s [10] and 60s [20]. The latter reference answered already the question on typical pursuit target speed and gives a range from 5° to 20°/s, which is in line with the results presented later. However, psychology focused on classifying motion types rather than on enhancing detectability.

Engel and Soechting [7] asked participants to follow a target moving on a touch-sensitive video monitor with their finger. It seems that there are similarities in smooth pursuit and manual tracking for changes in target direction [6]. This means that our results may apply to manual tracking, such as presented in the PathSync paper [2], even if we did not record and evaluate data from manual tracking.

Other authors discussed the changes in target direction [18] and the predictability of the target motion [21]. These aspects are relevant for pursuit target trajectory shapes – in particular the square and diamond shape studied in this paper.

A further paper discusses sequence learning in two-dimensional smooth pursuit eye movements [1].

The idea of 'pointing without a pointer' dates back to the year 2004 [26] followed by 'motion selection' in 2009 [9]. The concept of smooth pursuits as interaction method became popular in 2013 [24] and was soon thereafter followed by a publication of Cymek et al. [4]. Further research and possible applications are already mentioned in the introduction.

The initial pursuit paper of Vidal et al. [24] briefly comments on target speed whereas Orbits discusses target speed in more detail, using three different speeds [8]. Further statements on target speeds suggest them to be below 30°/s [25] and between 10°/s and 30°/s [11] – however without detailed discussion.

In summary, the goal of prior work was to find target properties that worked sufficiently well for the presented use case. At the same time, a more thorough analysis is still missing. In this work, we investigate seven different speeds. We identify the optimum, the limits, and provide theoretical explanations. Additionally, our research also makes specific suggestions as to how the detectable speed range can be enhanced.

USER STUDY

To understand the influence of target speed and trajectory, we collected data in a lab study and analyzed them regarding detection rates.

User Study Design

In the study we present a moving target on a screen and ask participants to follow the target with their gaze. We record the target and gaze positions in a text file. This allows the gaze movements to be evaluated with different detection methods and different parameters. Participants were presented two groups of tasks during the study.

Circular trajectories are popular in Pursuits research [8, 25]. To find the optimal speed for a pursuits target, one task was to select a single target moving on a circle with a radius of 200 pixels (4°). We chose seven different speeds in the range from 40 pixels/s to 1250 pixels/s or 0.8°/s to 25°/s, respectively as independent variables. Consequently the task consisted of seven subtasks.

The other task investigated square-shaped target trajectories. We used a square-shaped trajectory where the target moved clockwise and a diamond-shaped trajectory (a square rotated by 45°) where the target moved counter-clockwise. For both trajectories we used two different speeds, 400 pixels/s and 800 pixels/s (8° and 16°) resulting in four subtasks. The square-shaped trajectory consists on vertical and horizontal movements which means that only one muscle pair of the eye is needed to do the movement. In contrast the diamond-shaped trajectory requires simultaneous control of two muscle pairs per eye. The clockwise and counter-clockwise target movement was chosen to show that both directions of rotation produce the same effect.

The dependent variable for all task was the detection rate.

Experimental Setup

For recording the data we used a laptop with build-in eye tracker (Tobii IS4 Base AC). This is a low-cost device for the mass market and therefore a typical platform for gaze-aware interfaces.

The eye tracker was calibrated to one of the authors but not calibrated on the participants. This is to study instant use without calibration which is the desired capability for future gaze-aware interfaces and the main advantage of pursuit-based interfaces in combination with the correlation detection method.

The display was 38.4 cm wide and 21.7 cm high and had a resolution of 1920 × 1080 pixels. This means 0.2 mm for one pixel or 50 pixels per centimeter.

The distance of the eyes from the screen measured while filling the form was 53.5 cm with a standard deviation of 7.3 cm. This means around 50 pixels per degree for the visual angle.

Conducting the User Study

We invited 15 participants, aged 24 to 58, 11 male and 4 female, 6 with glasses and 9 without, to the study. In the beginning the participants filled a form with their demographic data. After this the participants got the task to follow the moving target in all the scenarios while the gaze path was recorded. One task lasted 12 seconds and every participant did each task once. The task order was randomized per participant. Every task started with a dialog, asking whether the participant was ready to start. This gave the participants the possibility for a short break between tasks. The recording of gaze data started with two seconds delay, cutting the gaze path for finding the target.

EVALUATION

Parameter Spaces

In smooth pursuit-based interfaces many parameters can be varied. In this study, these parameters are data window size and threshold value of the detection algorithm, pursuit speed, and size of the trajectory. The data window size is the shifting time interval for which the correlation is calculated. The time interval is equivalent to a number of data samples delivered from the eye tracker which normally runs on a fixed frequency. The threshold is the value the correlation has to exceed for a positive signal detection. For the Euclidian distance method, the threshold is the maximum distance between gaze and target position. As the Euclidian distance can be calculated per gaze data sample, we did not use a data window.

Using several values for each of these parameters results in a too large number of combinations be tested in a user study. Therefore we decided to use a set of parameters as typically mentioned in the literature and varied only the speed and the trajectory. We chose a data window of 30 samples which corresponds to 0.5 seconds on our 60 Hz eye tracker. We used a threshold value of 0.8 for the correlation method and 30 pixels (0.6°) for the Euclidian distance method which is slightly above the accuracy given for most eye trackers.

Observations for Circular Trajectories

Visual inspection of the gaze data revealed three different categories of data. The first category is perfect data, such as depicted in Figure 2. The second category is data where it can be assumed that the participant occasionally looked away from the target. An example is depicted in Figure 3. The third category is data where we assume that the eye tracker did not report correct coordinates. Figure 4 shows an example of the third category. We present the data to justify why we excluded the data of two participants and to demonstrate with which kind of data a detection algorithm has to deal. Both excluded participants wore glasses and it may be possible that reflections from the glasses fooled the eye tracker. However, four participants with glasses produced valid data.

Figure 2 shows the gaze trail of one participant for circular target movements on seven different speeds. As the eye tracker was not calibrated to the participants we used an implicit calibration (see Detection Rate section) for the visualization.

At slow speeds the eye follows the target with good accuracy and precision. As the target moves slow and the eye never stops moving, the gaze trail moves around the target position. These movements consist of micro-saccades, eye muscles' tremor and noise produced by the eye tracker which we will call simply noise in the following text. The amplitude of these movements limits the precision and is an important value for the smooth pursuit movement detectability.

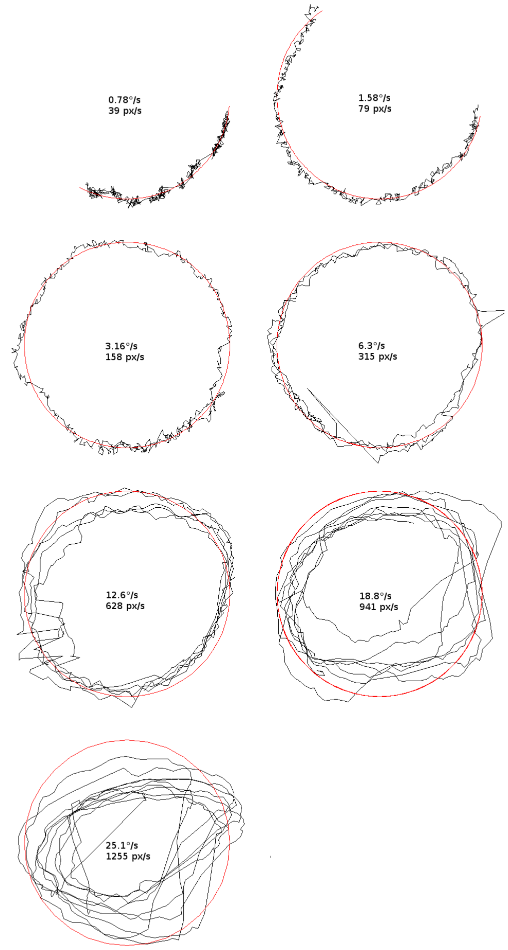

Figure 2. Target trajectory (red) and gaze trail (black) of one participant for 7 different target speeds (0.78°/s, 1.58°/s, 3.16°/s, 6.3°/s, 12.6°/s, 18.8°/s, 25.1°/s) over 10 seconds.

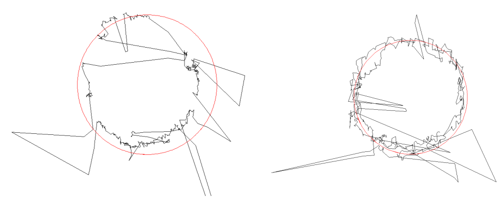

Figure 3. Target trajectory (red) and gaze trail (black) where the gaze does not follow the target perfectly.

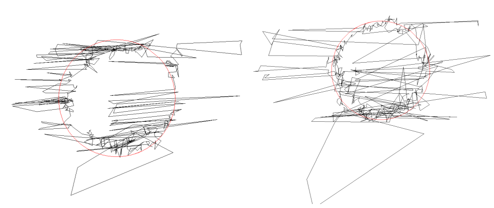

Figure 4. Target trajectory (red) and gaze trail (black) where the eye tracker did not report the gaze correctly.

At high speeds the eyes are not able to follow the target with a smooth pursuit movement and the movements turn into saccades. The eye tries to follow the target and when it lies behind it performs a saccade as a short-cut to the current target position. In such a situation the precision of the eye movements drops. These observations are in accordance with Esteves et al. who wrote: " if it is too slow it becomes a fixation; if it is too fast it turns into repeated saccades" [8].

Detection Rate for Circular Trajectories

We excluded three participants from the evaluation. For participants P2 and P8, the eye tracker did not report correct coordinates. Participant P14 moved the head out of the eye tracker's tracking area during the experiment causing a partially empty data set. The results for the circular trajectories are depicted in Figure 1.

The detection rate in this paper is the percentage of time where the algorithm signaled a detection. The participants were asked to follow the target during the time of recording and we assume that the participants were cooperative. It is worth to mention that there are other possible definitions for detection rate. For example it is possible to offer the participants pursuit targets and measure in how many cases the pursuit movement was detected. In this situation it is only necessary to have one short positive detection during the measurement for triggering the detection.

Figure 5 shows the measured detection rate over speed for all participants. The detection was done with the correlation method and works best in a range from 300 pixels/second to 800 px/s, which corresponds to 6°/s to 16°/s. For slower and faster movements the detection rate drops.

Figure 5. Detection rate over target speed for all participants.

Detection Rate on Slow Target Movements

It is easy to see in Figure 5 that the detection rate is poor for slow targets. Vidal et al. wrote "even if the pursuit is slow but analyzed for a longer period of time (w = 500ms), it can be robustly detected by Pursuits" [24]. This statement explains the relation between data window size and target speed. However, it is possible to let the target move even slower which needs the data window to be too big for practical interfaces.

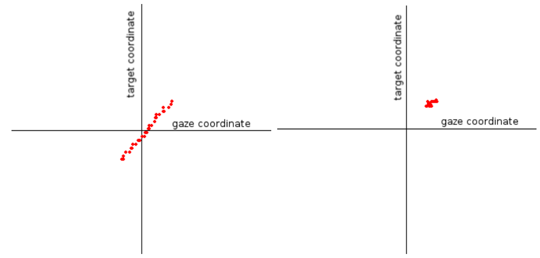

Figure 6. Regression line over 30 samples for a pursuit target moving at normal (left) and slow speed (right). For normal speed a linear dependency is clearly visible and the correlation evaluates to a value close to 1.0. For slow speed the target does not move much within the 30 sample data window. The noise in the data determines the correlation value.

The reason why slow target movements do not work well for detection with the correlation method is easy to understand. For the detection, the correlation has to be calculated for the x- and the y-coordinates and both values have to exceed a given threshold. The calculation of the correlation can be visualized plotting the samples of the data window as dots, where the dot's x-value is the x-value of the gaze position and the dot's y-value is the x-value of the pursuit target position. The correlation is a measure of the linear dependency, i.e. on how good these dots fit a straight line, called regression line.

Figure 6 shows the regression line from a data window of 30 samples for a normal pursuit speed of 6.3°/s and a slow speed of 0.8°/s. The correlation detection method does not work well on slow target speeds. The reason is that for the slow movement, target and gaze position do not move much compared to the amplitude of the gaze data noise. Hence, the slower the target movement the larger the data window has to be. As a rule of thumb we can say that the data window should have a size that the target moves at least 3–4 times the amplitude of the noise. For a slow target this means that the detection time can become very long and a classical dwell-time approach with a typical dwell-time of a second is faster.

From looking at Figure 2, we can see that the Euclidian distance method works well on slow target speeds, as long as the threshold is bigger than the amplitude of the noise. At very slow target speeds the method is just the classical dwell-time detection. However, the Euclidian distance method does not work well on high target speeds as the gaze trail is partially quite far from the target trajectory.

The problem with the Euclidian distance method is that it requires a calibrated eye tracker which destroys the main advantage of smooth pursuits. For confirming the statement that the Euclidian distance detection method works well on slow targets we evaluated the data also with this method.

The first step was to convert the gaze data to calibrated gaze data. We used a calibration method based on the regression line analysis. We used the slope of the regression line for scaling and the intercept as offset. For a perfectly calibrated eye tracker the gaze and the target position are identical and the slope of the regression line is one and the intercept is zero. If t is the target and g_r the reported gaze coordinate, the true gaze coordinate g_t can be calculated from slope s and intercept o as follows:

$$t = sg_r + o = g_t \tag{1}$$

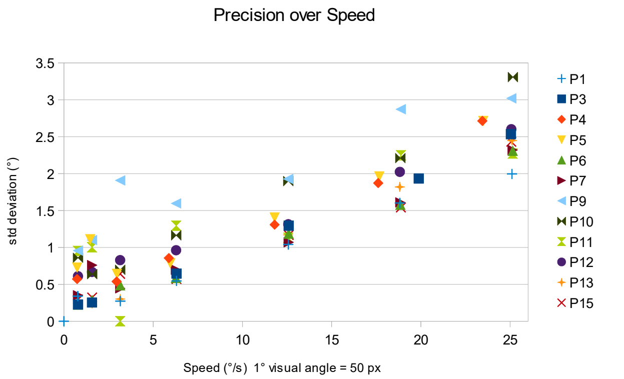

Figure 7. Standard deviation over target speed for all participants. With growing speed the gaze follows the target less precise.

This calibration method seems not to be discussed yet in the HCI community. This calibration method lies between the approach of Khamis et al. [15], which uses only offset but no scaling, and the approach of Pfeuffer et al. [19] which uses homographies. We used the complete ten second data set of a task, calculated the scaling factors and offsets and applied it to the gaze data. With the corrected data we calculated the Euclidian distance to the target and whenever the distance falls below 30 pixels we signal a positive detection. The result for Euclidian detection is shown in Figure 1.

For low speeds the Euclidian detection works very well and better than the correlation. For high speeds the Euclidian detection delivers very poor results. The reason lies in an increasing deviation of the gaze position from the target position with increasing target speed. After the post-calibration the accuracy of the gaze is perfect which means the average distance of gaze to target is zero. The aberration of the gaze from the target position can be expressed in terms of precision. There are different metrics for precision but the most common one seems to be standard deviation means the square root of the averaged squared distances. Figure 7 shows the precision over speed averaged over all participants. From this figure we can understand why the 30 pixels (0.6°) threshold works well for the Euclidian distance detection. For the high speeds the threshold for Euclidian distance detection should be increased to values above 100 pixels (2°). However for high speeds the correlation detection method is preferable.

Detection Rate on Fast Target Movements

The correlation detection works well for high target speeds even if there are already some saccadic movements. However, it seems that the gaze is a little bit behind the target. As a consequence, delaying the target before calculating the correlation improves the detection rate. Inspection of our data suggests that the gaze position is around 80 milliseconds behind the target position. There is also a delay in gaze position caused by the eye tracker but as the eye tracker delivers a sample every 16.7 milliseconds we assume that the eye tracker's delay is below this value and the effect is in the human eye.

Figure 8 shows gaze and target coordinates over time together with the correlation value and the detection indicators. The left side shows the situation for the unprocessed data and the right side shows the situation if the target coordinates are delayed by five samples (83 ms). The example shows a higher detection rate in the case of delayed target positions.

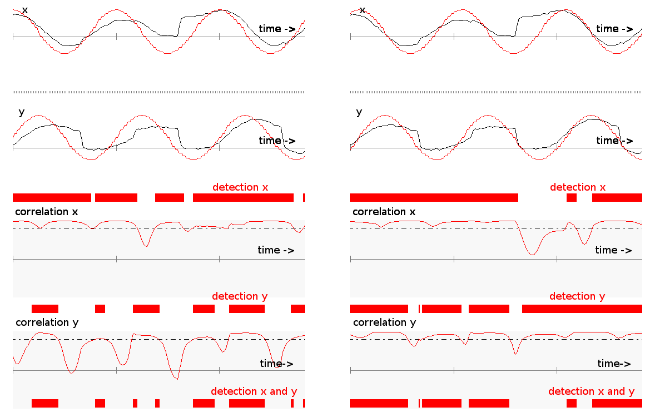

Figure 8. Coordinates of gaze and target, correlation and threshold condition over time without delay (left) and with 5 samples (83 milliseconds) delay of the target (right). The two upper rows are the gaze (black) and target (red) x- and y-coordinates over time. The two lower rows are calculated correlation values for x and y. The red bars indicate matching of the threshold condition for x, y, and both. The target moved at 25°/s. The data window was 30 samples.

Now we can compare three different detection methods. Figure 1 plots the detection rate averaged over all participants for the standard correlation method, the correlation method with a delayed target, and the Euclidian distance method.

Observation on Square-shaped Trajectories

Horizontal and vertical target movements create a problem when calculating the correlation. In the formula for the correlation (2), where g are the x- or y-coordinates of the gaze and t are the corresponding target coordinates, a horizontal or vertical target movement means constant values for t. This leads to a zero value for the difference of the sums and consequently a division-by-zero error occurs.

$$r = \frac{n\sum_{i=1}^{n} g_i t_i - \sum_{i=1}^{n} g_i \sum_{i=1}^{n} t_i}{\sqrt{n\sum_{i=1}^{n} g_i^2 - (\sum_{i=1}^{n} g_i)^2}\ \sqrt{n\sum_{i=1}^{n} t_i^2 - (\sum_{i=1}^{n} t_i)^2}} \quad (2)$$

An easy way out of the problem is to rotate the whole geometry, means gaze and target coordinates. This method is mentioned in the literature [22] and helps detecting pursuit movements on square-shaped trajectories. Preferably the angle should be chosen that the intervals for calculating the correlation are equal for x- and y-coordinates. If the rotation is only some degrees the change of a coordinate value within the data window will be small and the situation is similar to the slow moving target as depicted on the right side in Figure 6. In other words the horizontal or vertical line should have a 45° direction. Just for completeness, there is also another way to avoid the problem. If the data window is larger than the time the target needs to traverse a side the division-by-zero error will not occur.

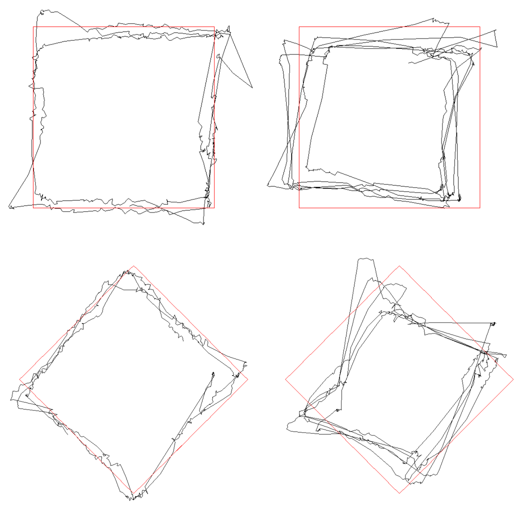

Figure 9. Target trajectory (red) and gaze path (black) for slow and fast square trajectories (target speeds clockwise 400, 800 px/s or 8, 16°/s) and slow and fast diamond trajectories (target speeds counter-clockwise 400, 800 px/s or 8, 16°/s).

When the gaze follows a target moving on a rectangular trajectory the direction change at the corner is unexpected. Therefore the gaze path shows overshooting at the corners. As the gaze position is now in a wrong position the consequent gaze movements try to compensate this and the movement has a different slope than the side of square. The resulting gaze path is a twisted square and the effect gets stronger with increasing target speed (see Figure 9).

Detection Rate for Square-shaped Trajectories

The detection rate for the square-shaped trajectory was 55% for a speed of 400 pixels/s (8°/s) and 40% for a speed of 800 pixels/s (16°/s). Looking at Figure 1 tells us that the detection rate is comparable to the circular trajectory for the lower speed and a little bit lower for the faster speed.

For the diamond-shaped trajectory the values were 56% and 33% and show the same trend. With only two speeds the data basis is not enough to show significance.

Applying the five sample delay on the detection for the square and diamond-shaped trajectories show a similar improvement for the detection rate as observed for the circular trajectories. Table 1 summarizes the improvements in the detection rate.

The delay mechanism compensates effects in time but does not consider spatial aspects as overshooting and compensation movements. Calculating the the detection with a slightly turned trajectory will most probably increase the detection rate further.

Discussion

From the recorded data we can see that we should not expect getting perfect data from the eye tracker. It is not quite clear which part of the imperfectness comes from technical aspects such as the eye tracker and the light conditions and which comes from the user. Technical problems may vanish with further technical development. The imperfectness in the human gaze may reduce with training. However, it seems that users' mental workload influences gaze behavior as Kosch et al. showed recently [16]. For the practical development of pursuit-based interfaces we should work with the assumption that the human gaze is 'buggy' and advanced techniques for pursuit detection such as outlier removal should be considered.

Trajectory	Speed	no delay	delay
square-shaped	8°/s	55%	62%
square-shaped	16°/s	40%	58%
diamond-shaped	8°/s	56%	63%
diamond-shaped	16°/s	33%	59%

Table 1. Detection rate for square- and diamond shaped trajectories at two different speeds and the improvements by calculation with a 5 sample delay (83 ms) for the target.

DESIGN RECOMMENDATIONS

The user study explored a part of the parameter space. In the following we make recommendations for the choice of parameter, depending on how free the designer is with regard to their choice. The degree of freedom may depend on whether an entirely new interface is being designed (usually no constraints) or whether pursuits is integrated with an existing interface (strong constraints). In some cases, modifications of the interface may be possible (few constraints).

No constraints. For a pursuit-based interfaces that allow freely choosing the parameters we recommend to use a 30 sample data window, 0.8 as threshold for correlation detection, a radius for a circular trajectory of 4° and a target speed of 5°/s to 15°/s.

Few Constraints. If there is a possible choice for the parameters but also a few constraints, a small user study may help to find the best parameters within the constraints. In general it is good to have predictable target movements. In particular, the target movement should not have abrupt direction changes, to avoid effects of overshooting and the connected compensation movements. The target speed should not be too high but high enough so that the gaze path covers a considerable distance and is hence detectable with the correlation method.

Strong constraints. If designers are not free to choose the parameters, for example because the pursuit target is part of a video, the knowledge from this study can be used to enhance the detection quality:

- If the target moves fast, a delay for the target coordinates improves the detectability. If the trajectory is not circular it can be a successful strategy to simulate overshooting and compensation and apply the detection algorithm on the adjusted trajectory.
- If the target is slow, the first improvement can be done by increasing the time for the data window. If this is not sufficient because the target is very slow, the only way is to switch to Euclidian detection. The slower the target gets the more the situation degenerates to classical dwell time detection. Using Euclidian detection means to loose the advantage of not having to calibrate.

- If the target moves with variable speeds it may be possible to use the higher speeds for implicit calibration and use this for detection with the Euclidian distance when the target moves slow.

CONCLUSION & FUTURE WORK

In this paper we explored how the speed and trajectory of a smooth pursuits stimulus impacts on detection accuracy. If employed by the designers, they can optimize the detection quality of their interface. Furthermore, we presented some coping strategies in case designers are not capable of freely choosing the optimal parameters. In this way we hope to contribute to smooth pursuits findings their way into more applications.

Still, there are a lot of opportunities for future work. Firstly, this study varied the speed on a circle trajectory but did not vary the radius. The radius however seems to have an influence on the optimal speed. A circular movement is not a uniform movement if looking at the right-left and up-down movements separately. The muscle pairs which control the eye movements have to accelerate and decelerate to perform the circular movement and acceleration relates directly to muscle force. Secondly, this study did not investigate the user experience. Future work could investigate how the modification of parameters impact on how users perceive such interfaces.

ACKNOWLEDGEMENTS

This work was funded, in part, by the Bavarian State Ministry of Education, Science and the Arts in the framework of the Center Digitization.Bavaria (ZD.B). This research was supported by the German Research Foundation (DFG), Grant No. AL 1899/2-1.

REFERENCES

1. Melanie Rose Burke and Graham R. Barnes. 2007. Sequence learning in two-dimensional smooth pursuit eye movements in humans. *Journal of Vision* 7, 1 (1 2007), 1–12. DOI:`http://dx.doi.org/10.1167/7.1.5`

2. Marcus Carter, Eduardo Velloso, John Downs, Abigail Sellen, Kenton O'Hara, and Frank Vetere. 2016. PathSync: Multi-User Gestural Interaction with Touchless Rhythmic Path Mimicry. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3415–3427. DOI: `http://dx.doi.org/10.1145/2858036.2858284`

3. Christopher Clarke, Alessio Bellino, Augusto Esteves, Eduardo Velloso, and Hans Gellersen. 2016. TraceMatch: A Computer Vision Technique for User Input by Tracing of Animated Controls. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 298–303. DOI: `http://dx.doi.org/10.1145/2971648.2971714`

4. Dietlind Helene Cymek, Antje Christine Venjakob, Stefan Ruff, Otto Hans-Martin Lutz, Simon Hofmann, and Matthias Roetting. 2014. Entering PIN codes by smooth pursuit eye movements. *Journal of Eye Movement Research* 7, 4 (2014). `https://bop.unibe.ch/index.php/JEMR/article/view/2384`

5. Heiko Drewes and Albrecht Schmidt. 2007. Interacting with the Computer Using Gaze Gestures. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II (INTERACT'07)*. Springer-Verlag, Berlin, Heidelberg, 475–488. `http://dl.acm.org/citation.cfm?id=1778331.1778385`

6. Kevin C. Engel, John H. Anderson, and John F. Soechting. 2000. Similarity in the Response of Smooth Pursuit and Manual Tracking to a Change in the Direction of Target Motion. *Journal of Neurophysiology* 84, 3 (2000), 1149–1156. DOI: `http://dx.doi.org/10.1152/jn.2000.84.3.1149` PMID: 10979990.

7. Kevin C. Engel and John F. Soechting. 2000. Manual Tracking in Two Dimensions. *Journal of Neurophysiology* 83, 6 (2000), 3483–3496. DOI: `http://dx.doi.org/10.1152/jn.2000.83.6.3483` PMID: 10848564.

8. Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 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)*. ACM, New York, NY, USA, 457–466. DOI: `http://dx.doi.org/10.1145/2807442.2807499`

9. Jean-Daniel Fekete, Niklas Elmqvist, and Yves Guiard. 2009. Motion-pointing: Target Selection Using Elliptical Motions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 289–298. DOI: `http://dx.doi.org/10.1145/1518701.1518748`

10. Westerheimer G. 1954. Eye movement responses to a horizontally moving visual stimulus. *A.M.A. Archives of Ophthalmology* 52, 6 (1954), 932–941. DOI: `http://dx.doi.org/10.1001/archopht.1954.00920050938013`

11. Igi Ardiyanto Herlina, Sunu Wibirama. 2018. Similarity measures of object selection in interactive applications based on smooth pursuit eye movements. In *International Conference on Information and Communications Technology (ICOIACT) 2018 (UbiComp '16)*.

12. Robert J. K. Jacob. 1990. What You Look at is What You Get: Eye Movement-based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. ACM, New York, NY, USA, 11–18. DOI: `http://dx.doi.org/10.1145/97243.97246`

13. Mohamed Khamis, Florian Alt, and Andreas Bulling. 2016. Challenges and Design Space of Gaze-enabled Public Displays. In *Ajunct Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 10. DOI: `http://dx.doi.org/10.1145/2968219.2968342`

14. Mohamed Khamis, Axel Hoesl, Alexander Klimczak, Martin Reiss, Florian Alt, and Andreas Bulling. 2017. EyeScout: Active Eye Tracking for Position and Movement Independent Gaze Interaction with Large Public Displays. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 155–166. DOI: http://dx.doi.org/10.1145/3126594.3126630

15. Mohamed Khamis, Ozan Saltuk, Alina Hang, Katharina Stolz, Andreas Bulling, and Florian Alt. 2016. TextPursuits: Using Text for Pursuits-based Interaction and Calibration on Public Displays. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 274–285. DOI: http://dx.doi.org/10.1145/2971648.2971679

16. Thomas Kosch, Mariam Hassib, Paweł W. Woźniak, Daniel Buschek, and Florian Alt. 2018. Your Eyes Tell: Leveraging Smooth Pursuit for Assessing Cognitive Workload. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 436, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3174010

17. Päivi Majaranta and Kari-Jouko Räihä. 2002. Twenty Years of Eye Typing: Systems and Design Issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM, New York, NY, USA, 15–22. DOI: http://dx.doi.org/10.1145/507072.507076

18. Soechting J.F. Mrotek L.A., Flanders M. 2006. Oculomotor responses to gradual changes in target direction. *Experimental Brain Research* 172, 2 (2006), 175,192. DOI: http://dx.doi.org/10.1007/s00221-005-0326-1

19. Ken Pfeuffer, Melodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 261–270. DOI: http://dx.doi.org/10.1145/2501988.2501998

20. D A Robinson. 1965. The mechanics of human smooth pursuit eye movement. *The Journal of Physiology* 180, 3 (1965), 569–591. DOI: http://dx.doi.org/10.1113/jphysiol.1965.sp007718

21. J. F. Soechting, H. M. Rao, and J. Z. Juveli. 2010. Incorporating Prediction in Models for Two-Dimensional Smooth Pursuit. *PLoS ONE* 5 (Sept. 2010), e12574. DOI: http://dx.doi.org/10.1371/journal.pone.0012574

22. Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. 2017. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Trans. Comput.-Hum. Interact.* 24, 3, Article 22 (April 2017), 35 pages. DOI: http://dx.doi.org/10.1145/3064937

23. Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. 2016. AmbiGaze: Direct Control of Ambient Devices by Gaze. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 812–817. DOI: http://dx.doi.org/10.1145/2901790.2901867

24. Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 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 (UbiComp '13)*. ACM, New York, NY, USA, 439–448. DOI:http://dx.doi.org/10.1145/2493432.2493477

25. Oleg Špakov, Poika Isokoski, Jari Kangas, Deepak Akkil, and Päivi Majaranta. 2016. PursuitAdjuster: An Exploration into the Design Space of Smooth Pursuit –based Widgets. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 287–290. DOI: http://dx.doi.org/10.1145/2857491.2857526

26. John Williamson and Roderick Murray-Smith. 2004. Pointing Without a Pointer. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04)*. ACM, New York, NY, USA, 1407–1410. DOI: http://dx.doi.org/10.1145/985921.986076