Circular Orbits Detection for Gaze Interaction Using 2D Correlation and Profile Matching Algorithms

Eduardo Velloso
Interaction Design Lab
The University of Melbourne
evelloso@unimelb.edu.au

Flavio Luiz Coutinho
School of Arts, Sciences and Humanities
University of São Paulo
flcoutinho@usp.br

Andrew Kurauchi
Computer Science Department
University of São Paulo
kurauchi@ime.usp.br

Carlos H Morimoto
Computer Science Department
University of São Paulo
hitoshi@ime.usp.br

ABSTRACT

Recently, interaction techniques in which the user selects screen targets by matching their movement with the input device have been gaining popularity, particularly in the context of gaze interaction (e.g. *Pursuits, Orbits, AmbiGaze*, etc.). However, though many algorithms for enabling such interaction techniques have been proposed, we still lack an understanding of how they compare to each other. In this paper, we introduce two new algorithms for matching eye movements: Profile Matching and 2D Correlation, and present a systematic comparison of these algorithms with two other state-of-the-art algorithms: the Basic Correlation algorithm used in *Pursuits* and the Rotated Correlation algorithm used in *PathSync*. We also examine the effects of two thresholding techniques and post-hoc filtering. We evaluated the algorithms on a user dataset and found the 2D Correlation with one-level thresholding and post-hoc filtering to be the best performing algorithm.

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**; *Interaction devices*;

KEYWORDS

Eye tracking, Pursuits, Orbits, Smooth pursuits, Gaze interaction

ACM Reference format:
Eduardo Velloso, Flavio Luiz Coutinho, Andrew Kurauchi, and Carlos H Morimoto. 2018. Circular Orbits Detection for Gaze Interaction Using 2D Correlation and Profile Matching Algorithms. In *Proceedings of ETRA '18: 2018 Symposium on Eye Tracking Research and Applications, Warsaw, Poland, June 14–17, 2018 (ETRA '18),* 9 pages.
https://doi.org/10.1145/3204493.3204524

1 INTRODUCTION

Most graphical user interfaces are largely static, displaying widgets that remain in place until the user scrolls or transitions to a different page. However, recent work has been exploring alternatives to static layouts, where elements perform distinct movements that can be matched by the user to select them [Velloso et al. 2017]. Velloso et al. refer to this principle as *Motion Correlation* [Velloso et al. 2017], encompassing techniques based on a wide range of devices, including mice [Fekete et al. 2009; Williamson and Murray-Smith 2004], accelerometers [Verweij et al. 2017b], depth cameras [Carter et al. 2016], and webcams [Clarke et al. 2016].

Of particular interest to us are techniques that employ this principle for gaze interaction. When following a moving target, our eyes engage in *smooth pursuit*, an eye movement that is difficult to fake and roughly matches the trajectory of the target [Vidal et al. 2013]. This allows us to compare the trajectory of the gaze point to those of the moving targets to detect which target is being followed. Such principle has been employed in previous works to unobtrusively calibrate the eye tracker [Pfeuffer et al. 2013], to allow for calibration-free interaction [Vidal et al. 2013], to enable interaction with small screens [Esteves et al. 2015b], to control smart environments [Velloso et al. 2016], etc.

A crucial step in these techniques is to quantify the similarity between the movement of the eyes and those of the targets. Previous works have proposed a variety of alternatives, including normalised Euclidean distance [Fekete et al. 2009], Pearson's correlation [Vidal et al. 2013], and Pearson's correlation on rotated data [Carter et al. 2016]. In this paper we propose two new algorithms for eye-target movement matching: Profile Matching and 2D Correlation. The 2D Correlation algorithm is based on the fit of the target trajectory as an approximation for the gaze point. The Profile Matching algorithm computes the similarity between normalized profiles generated by the gaze and target movements.

We compare these to two state-of-the-art algorithms used for gaze-based motion correlation detection — the original *Pursuits* algorithm [Vidal et al. 2013] and the extension proposed in *PathSync* [Carter et al. 2016]. We evaluated them in the dataset collected in the development of *Orbits* [Esteves et al. 2015b].

Our results suggest that the 2D Correlation algorithm outperforms the other three when used in conjuction with a post-hoc filter, as implemented by Carter et al. [Carter et al. 2016]. We also analysed the effects of the window size of the post-hoc filter on the recognition performance, and found that, for this dataset, a window size of 30-40 frames (1s) offered the best results.

In summary, we contribute (1) two new algorithms for motion correlation detection; (2) the first empirical comparison of these new algorithms to previous ones; (3) an analysis of the effect of the window size of the post-hoc filter on the algorithm performance. These results provide an important stepping stone for the future development of gaze interaction techniques based on smooth pursuit and for motion correlation techniques in general.

ETRA '18, June 14–17, 2018, Warsaw, Poland

ACM ISBN 978-1-4503-5706-7/18/06…$15.00
https://doi.org/10.1145/3204493.3204524

2 RELATED WORK

2.1 Selection by Motion Correlation

The idea of using motion correlation for selection has its origins in the work of Williamson and Murray-Smith [Williamson 2006; Williamson and Murray-Smith 2004]. The authors built prototypes that introduced small disturbances in the movement of each selectable interface element. Inspired by control theory, they looked for signs that the user was attempting to match or cancel these disturbances to identify which element they wanted to select. Fekete et al. took this idea further by incorporating cyclical movement into conventional buttons [Fekete et al. 2009]. Rather than clicking on the button to select it, the user had the choice to match the movement of its corresponding orbiting widget with the mouse.

Later work took this idea to contexts beyond the desktop. Carter et al. adapted the principle for interaction with public displays using a Kinect in *PathSync* [Carter et al. 2016; Cox et al. 2016]. As advantages of the technique, the authors found that it made the gestures available in the interface more visible and allowed cursorless multi-user interaction. Because these techniques do not rely on the absolute position of the input device, they can be used in sensors that are better suited to capturing relative movement, such as accelerometers. Verweij et al. demonstrated that it is possible to employ this principle by sensing hand movements using the sensors in smartwatches [Verweij et al. 2017a,b,c].

Because users are able to match movements using many body parts, previous works have shown applications that go beyond hand gestures. In *SmoothMoves*, Esteves et al. tracked the user's head movements using an augmented reality headset to match the motions being displayed in the AR interface [Esteves et al. 2017]. Clarke et al. abstracted from the body part that is matching the movement by tracking any matching body part using a webcam [Clarke et al. 2017, 2016; Clarke and Gellersen 2017]. The authors showed that the technique can work with movements ranging from a hand motion while holding a cup of tea to feet movements.

These works demonstrate the wide range of input devices that can be used with motion correlation. For an in-depth discussion of the principle and design guidelines, see [Velloso et al. 2017].

2.2 Selection by Smooth Pursuits

As well as the advantages offered to input devices discussed above, motion correlation is particularly well suited for gaze interaction. When our eyes follow a moving target, they engage in smooth pursuits, an eye movement that is characteristically smooth in comparison to saccades; that is difficult to fake, as it requires a moving object for the eyes to latch onto; and that is comfortable to perform, as the eyes are naturally drawn to movement.

The idea of using motion correlation in the context of gaze interaction was first proposed by Vidal et al. in *Pursuits* [Vidal et al. 2013]. The authors demonstrated the applicability of the technique for spontaneous interaction with public displays, gaming, and authentication. Because the eyes are naturally drawn to movement, Pfeuffer et al. leveraged this principle to calibrate the eye tracker without the user even being aware of the process [Pfeuffer et al. 2013]. One of its advantages is that it allows the procedure to know whether the user was actually following the calibration target or

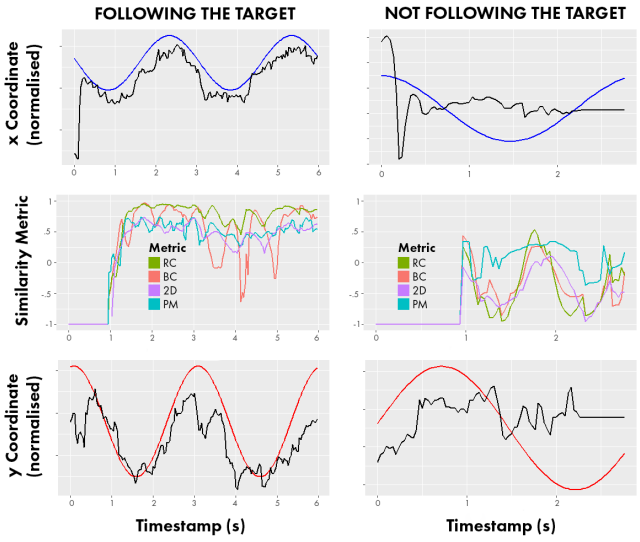

Figure 1: Behaviour of the four similarity metrics on sample data from Esteves et al. Left: the user actively follows the target. Right: the user reads the time while ignoring the moving target. Top and Bottom: the movement of the gaze point (black) in comparison to the target (blue and red). Middle: the output of each metric computed in 30-sample windows.

was distracted. Khamis et al. showed that even moving text can be used for this purpose [Khamis et al. 2016a].

Motion correlation is also helpful in enabling gaze interaction in challenging settings. Given the difficulty of estimating the gaze point in large horizontal surfaces, Newn et al. employed horizontal motion correlation to enable users to select distant targets on an interactive tabletop [Newn et al. 2016]. In *Orbits*, Esteves et al. enabled gaze interaction with smart watches by matching the user's eye movements to small targets moving in circles on the face of the watch [Esteves et al. 2015a,b]. Motion correlation allowed the authors to make more targets available than it would have otherwise been possible with touch input in such a small screen. In *AmbiGaze*, Velloso et al. explored how to enable interaction based on motion correlation with smart devices by deriving a design space of possibilities for embedding movement into the environment, including physical movements from a spinning windmill and a mechanical arm [Velloso et al. 2016].

Despite the huge variety of devices and applications discussed so far, they all share the same requirement—it is necessary to quantify the similarity between the movement of the input device and the movement of the targets. Due to the multiple ways in which this comparison can be made, many metrics have been proposed for this purpose, but without any comparison to previous metrics. As a consequence, we know little about which algorithm works best for each purpose or whether there is an optimal method that works well in all settings. In this paper, we aim at answering this question by comparing algorithms proposed in the literature on an existing dataset collected with users. In addition, we propose two new metrics and compare them to the state-of-the-art.

3 SIMILARITY METRICS

The general problem that an appropriate similarity metric must solve in the context of motion correlation is that, given a window of two synchronised time series of horizontal and vertical coordinates—one pair for the gaze point and one pair for a given target—the metric must output a high similarity value when the user is following the target, and a low similarity when they are not. Many metrics for comparing time series exist in the literature, including Euclidean distance, Dynamic Time Warping, among others. However, for our purposes, we only consider metrics that satisfy a set of criteria.

First, it must be efficient. Because our goal is to use it for interactive purposes, it must be able to make a decision in near real-time. Second, it must be scale-invariant. One of the advantages of motion correlation techniques is that they do not need any calibration between the input device and the targets. This is particularly useful for gaze interaction, where maintaining an accurate calibration is a challenge. Further, this enables interaction using devices such as eye trackers based on electrooculography, as demonstrated by Shimizu et al. [Shimizu et al. 2016] and Dhuliawala et al. [Dhuliawala et al. 2016], who implemented it for the JINS MEME EOG glasses. Therefore, the metric must not rely on information about the absolute coordinates of the gaze points or the targets, such as in Kangas et al.'s approach [Kangas et al. 2016]. Third, it must be sensitive to phase differences. This means that the algorithm should be able to distinguish between, for example, multiple targets moving around a circle at the same speed, but slightly offset from each other, as in *Orbits* [Esteves et al. 2015b]. As a fourth and final requirement, we do not consider approaches based on machine learning classifiers, such as the smooth pursuit detector based on eye movement shape features built by Vidal et al. [Vidal et al. 2012]. Though such approaches might offer potential solutions in the future, they require the collection of extensive datasets to demonstrate their validity, suffer the risk of overfitting the training data, and generalise poorly to new contexts.

We compare the performance of the following algorithms: the Basic Correlation (BC) used in *Pursuits* [Vidal et al. 2013], the Rotated Correlation (RC) used in *PathSync* [Carter et al. 2016], and two new ones—Profile Matching (PM) and 2D Correlation (2D). Figure 1 illustrates the behaviour of the four metrics computed on two datasets. On the left, the user is actively following a target moving along a circular trajectory. On the right, the user is reading the time on the watch while actively trying to avoid looking at the moving target. We see that all metrics output high values in the positive case and low values in the negative case. The challenge for the algorithm then becomes to fine-tune its parameters to find the best threshold that separates the two cases.

3.1 Basic Correlation

The use of Pearson's Correlation to quantify the similarity between the trajectory of the input device and the target was first suggested by Fekete et al. [Fekete et al. 2009]. The authors evaluated its performance by computing the sum of the correlation between the two axes across the window. In their empirical evaluation, the algorithm was outperformed by other metrics.

ALGORITHM 1: Basic Correlation (BC)

Input: A $T \times 2$ time series with uncalibrated 2D coordinates of the gaze point $P(In) = \{x_t^{In}, y_t^{In}\}$ at time t; a $T \times 2$ time series with the 2D coordinates $P(Out) = \{x_t^{Out}, y_t^{Out}\}$ of a target on the screen at time t; a window size $N < T$.

Output: An array c of length T, with the lowest correlation between the gaze point and the target.

for $t \in N..T$ **do**

$W_x^{In} \leftarrow \{x_{t-N}^{In}..x_t^{In}\}$;

$W_y^{In} \leftarrow \{y_{t-N}^{In}..y_t^{In}\}$;

$W_x^{Out} \leftarrow \{x_{t-N}^{Out}..x_t^{Out}\}$;

$W_y^{Out} \leftarrow \{y_{t-N}^{Out}..y_t^{Out}\}$;

$c_t \leftarrow min(cor(W_x^{In}, W_x^{Out}), cor(W_y^{In}, W_y^{Out}))$;

where $cor(a,b) = \frac{E[(a-\bar{a})(b-\bar{b})]}{\sigma_a \sigma_b}$;

end

Rather than adding the correlation coefficient for both axes into a single metric, Vidal et al. computed the two and independently checked whether both crossed the threshold set by the interface designer [Vidal et al. 2013]. Similarly, Esteves et al. discarded the axis with the highest correlation and only compared the axis with the lowest correlation [Esteves et al. 2015b]. This works better than adding the two correlation coefficients in cases where the movement is correlated in one axis but not in the other, such as when two targets are orbiting around the trajectory with the same phase but in opposite directions. Algorithm 1 formally defines the Basic Correlation method. Given that this is the most used algorithm in the related work (e.g. [Esteves et al. 2015a,b; Khamis et al. 2015, 2016a,b; Vidal et al. 2013]), we use it as our baseline case.

3.2 Rotated Correlation

A problem that arises in the Basic Correlation algorithm is when, in a given window, there is no change in the movement in one of the axes. This happens, for example, when traversing one of the edges of a square. In such cases, it is not possible to compute the correlation in this axis, because the standard deviation of the target trajectory is zero. This leads to a division by zero in the computation of the correlation coefficient.

The problem was identified by Carter et al., who built an interface with square movements similar to the Windows Metro design [Carter et al. 2016]. To address this problem, the authors suggested rotating both the target trajectory and the input device trajectory by the same amount, so as to maximise the variance of the target trajectory movement in both axes. They consider the optimal rotation to be the one 45° away from the output of a Principal Component Analysis, which is equivalent to multiplying the trajectory by a rotation matrix with the eigenvectors of the trajectory and by a 45° rotation matrix. The same approach was used in other works, including Velloso et al. and Cox et al. [Cox et al. 2016; Velloso et al. 2016]. The Rotated Correlation algorithm is defined in Algorithm 2. Despite the anecdotal improvement reported in these papers, there are no published results that quantify the improvement over the

ALGORITHM 2: Rotated Correlation (RC)

Input: A $T \times 2$ time series with uncalibrated 2D coordinates of the gaze point $P(In) = \left\{x_t^{In}, y_t^{In}\right\}$ at time t; a $T \times 2$ time series with the 2D coordinates $P(Out) = \left\{x_t^{Out}, y_t^{Out}\right\}$ of a target on the screen at time t; a window size $N < T$.

Output: An array c of length T, with the lowest correlation between the gaze point and the target.

for $t \in N..T$ **do**

$$RotMat \leftarrow \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \times \begin{bmatrix} \vec{v1}_x & \vec{v2}_x \\ \vec{v1}_y & \vec{v2}_y \end{bmatrix}$$

where $\vec{v1}, \vec{v2}$ are the eigenvectors of the matrix with $\left\{x_t^{Out_i}, y_t^{Out_i}\right\}$ as columns.

$$\begin{bmatrix} \vdots & \vdots \\ u_t^{Out_i} & v_t^{Out_i} \\ \vdots & \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \vdots & \vdots \\ x_t^{Out_i} & y_t^{Out_i} \\ \vdots & \vdots \end{bmatrix} \times RotMat$$

$$\begin{bmatrix} \vdots & \vdots \\ u_t^{In} & v_t^{In} \\ \vdots & \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \vdots & \vdots \\ x_t^{In} & y_t^{In} \\ \vdots & \vdots \end{bmatrix} \times RotMat$$

$W_u^{In} \leftarrow \{u_{t-N}^{In}..u_t^{In}\}$; $W_v^{In} \leftarrow \{v_{t-N}^{In}..v_t^{In}\}$;
$W_u^{Out} \leftarrow \{u_{t-N}^{Out}..u_t^{Out}\}$; $W_v^{Out} \leftarrow \{v_{t-N}^{Out}..v_t^{Out}\}$;
$c_t \leftarrow min\left(cor\left(W_u^{In}, W_u^{Out}\right), cor\left(W_v^{In}, W_v^{Out}\right)\right)$;

where $cor(a,b) = \frac{E[(a-\bar{a})(b-\bar{b})]}{\sigma_a \sigma_b}$;

end

Basic Correlation algorithm employed in other works, so we chose this metric as the first comparison.

3.3 2D Correlation

A problem with the approaches presented so far is that they treat the movement in each axis independently from the movement in the other axis. In order to consider both axes simultaneously, we take the basic idea of Pearson's Correlation in the context of least squares regression analysis and extend it to the two-dimensional space. We use as a similarity metric the R^2 of the target trajectory as an approximation of the gaze point trajectory. However, we must slightly modify the metric to account for the fact that in our case Y is not a function of X.

We begin by centring and scaling each trajectory. We centre each trajectory at zero and use as the scaling factor the standard deviation of the axis with the largest variance. By using the same scaling factor for both axes, we ensure that the scaled trajectory has the same proportions of the original. It is important to note that by treating each axis independently, the previous algorithms scale the coordinates by different scaling factors, which may lead to false positives in cases where the trajectory of one target is a squashed or stretched version of the trajectory of another target (e.g. comparing a circle against and ellipse).

ALGORITHM 3: 2D Correlation (2D)

Input: A $T \times 2$ time series with uncalibrated 2D coordinates of the gaze point $P(In) = \left\{x_t^{In}, y_t^{In}\right\}$ at time t; a $T \times 2$ time series with the 2D coordinates $P(Out) = \left\{x_t^{Out}, y_t^{Out}\right\}$ of a target on the screen at time t; a window size $N < T$.

Output: An array c of length T, with the R^2 similarity metric between the gaze point and the target.

$SD_{In} \leftarrow max\left(SD\left(x_t^{In}\right), SD\left(y_t^{In}\right)\right)$;
$SD_{Out} \leftarrow max\left(SD\left(x_t^{Out}\right), SD\left(y_t^{Out}\right)\right)$;
$x_t^{In} \leftarrow scale\left(x_t^{In}, \mu = 0, SD_{In}\right)$;
$y_t^{In} \leftarrow scale\left(y_t^{In}, \mu = 0, SD_{In}\right)$;
$x_t^{Out} \leftarrow scale\left(x_t^{Out}, \mu = 0, SD_{Out}\right)$;
$y_t^{Out} \leftarrow scale\left(y_t^{Out}, \mu = 0, SD_{Out}\right)$;

for $t \in N..T$ **do**

$SS_{Reg} \leftarrow \sum_{i=t-N}^{t} \sqrt{\left(x_t^{In} - x_t^{Out}\right)^2 + \left(y_t^{In} - y_t^{Out}\right)^2}$;
$SS_{Total} \leftarrow \sum_{i=t-N}^{t} \sqrt{\left(x_t^{In}\right)^2 + \left(y_t^{In}\right)^2}$;
$c_t \leftarrow R^2 = \frac{SS_{Reg}}{SS_{Total}}$;

end

We then compute the "regression" sum of squares (SS_{Reg}) by computing the Euclidean distance between the corresponding points of the two normalised trajectories and by adding them up. We compute the total sum of squares (SS_{Total}) by calculating the sum of the distances of the normalised gaze points to the centroid of the trajectory. Because during the normalisation we centred the points around the origin, the centroid of the trajectory is the point $(0, 0)$. We finally compute the R^2 of the fit by dividing SS_{Reg} by SS_{Total}. Algorithm 3 formally describes this approach.

3.4 Profile Matching

The trajectory described by the eye as it follows a moving target during a time interval Δ_t could alternatively be described by the orthogonal profile of the gaze samples projected onto the `baseline` $\vec{b} = (P_M - P_0)$, with P_0 being the first point in the time interval and P_M the farthest point from P_0 during Δ_t, as seen in Figure 2. Observe that, in the particular case of circular trajectories, for long arcs (greater than π), P_M is not the endpoint of the sequence.

Because gaze points are projected onto the baseline vector as $(u(t), v(t))$, for $t \in [1, N]$, this profile representation is rotation invariant and phase information is lost. Also, $(u(t), v(t))$ is further normalized by $|\vec{b}|$ (the length of $\vec{b}$) making the transformation invariant to scaling as well (we will use $(\hat{u}(t), \hat{v}(t))$ to denote the normalized sequence).

Applying the same process to the orbits described by each target i, the trajectory profile for the gaze data can be compared to each target profile. The sum of squared differences (SSD), as defined in (1), is used to quantify the similarity between profiles, where $\hat{G}$ is the profile generated by the gaze samples and $\hat{T}_i$ the profile of a

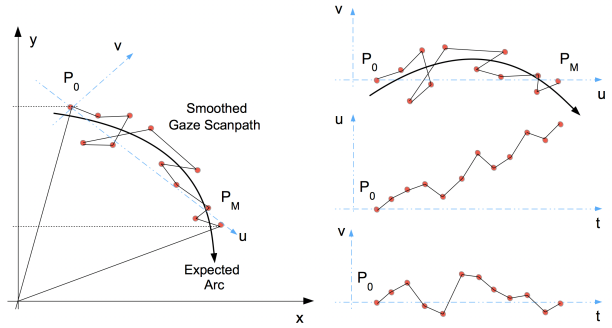

Figure 2: Profile matching: Points within the time interval are projected onto the longest baseline starting at P_0. The normalized rotated arrays $(\hat{u}(t), \hat{v}(t))$ are used for matching.

target i. Small SSD values indicate a good matching, i.e., that the eye is following an orbit with approximately the same angular velocity (but not necessarily with the same phase in the particular case of circular trajectories).

$$SSD(\hat{G}, \hat{T}_i) = \sum_{k=1}^{N} (\hat{G}(k) - \hat{T}_i(k))^2 \quad (1)$$

Phase information could be used to reduce false positives or to distinguish between targets of different phases but with the same velocity. Let α be the phase difference. The cosine of α ($cos(\alpha)$) can be computed as the dot product between the normalized baseline vectors of $\hat{G}$ and $\hat{T}_i$. The SSD and $cos(\alpha)$ values can be combined to produce an adjusted measure that reflects not just similarity in trajectory shape, but also in phase.

The complete description of the algorithm is presented in Algorithms 4 (main algorithm for *Profile Matching*) and 5 (auxiliary function *Rotate and Scale*). The last statement of Algorithm 4, in which c_t is computed, shows how SSD and $cos(\alpha)$ are combined, to produce a similarity measure in the $[-1, 1]$ range, with values next to 1 meaning high similarity in trajectory shape and phase, and smaller values meaning low similarity (due mismatching of either trajectory or phase).

3.5 Thresholding and Filtering Procedures

The metrics discussed so far quantify the similarity between the trajectories of the gaze point and the target within a given window. A window is considered activated if the metric crosses a certain threshold, typically empirically determined to give more weight to precision or recall, depending on the requirements of the application. However, triggering system actions based on a decision made on a single window is prone to false activations due to spurious high correlations. Therefore, to increase the robustness of the system, we consider the output of multiple consecutive windows in a *post-hoc filter*. The simplest implementation is to trigger a system action if all (or most) of a given number of consecutive windows cross the threshold.

Though this is often sufficient, it might lead to false negatives, due to the difficulty of maintaining a high correlation for a long period of time. An alternative is to use a bi-level threshold [Negulescu et al. 2012], as implemented in *PathSync* [Carter et al. 2016]. The idea is to set two thresholds—after crossing the higher one, the user must only maintain a correlation above the lower one for a certain duration in order to activate it. If the correlation goes below the lower threshold, it must cross the higher one again. This makes the system more lenient immediately after a high correlation is found, which tends to reduce the false negative rate.

In this paper, we compare the performance of the four metrics using a one-level threshold (Simple) and a bi-level threshold (Double). In addition, we compare the results with and without a post-hoc filter and evaluate the effect of the number of consecutive windows necessary to trigger an action on the performance results of the best performing metric.

ALGORITHM 4: Profile Matching (PM)

Input: A $T \times 2$ time series with uncalibrated 2D coordinates of the gaze point $P(In) = \{x_t^{In}, y_t^{In}\}$ at time t; a $T \times 2$ time series with the 2D coordinates $P(Out) = \{x_t^{Out}, y_t^{Out}\}$ of targets on the screen at time t; a window size $N < T$.

Output: An array c of length T, with matching scores (in the [-1, 1] range) computed for each gaze-target window of size N.

for $t \in N..T$ **do**

$W_x^{In} \leftarrow \{x_{t-N}^{In}..x_t^{In}\}$;
$W_y^{In} \leftarrow \{y_{t-N}^{In}..y_t^{In}\}$;
$W_x^{Out} \leftarrow \{x_{t-N}^{Out}..x_t^{Out}\}$;
$W_y^{Out} \leftarrow \{y_{t-N}^{Out}..y_t^{Out}\}$;

$(W_u^{In}, W_v^{In}, \vec{b}^{In}) \leftarrow$ **rot_scale**(W_x^{In}, W_y^{In});
$(W_u^{Out}, W_v^{Out}, \vec{b}^{Out}) \leftarrow$ **rot_scale**(W_x^{Out}, W_y^{Out});

$ssd_t \leftarrow \sum_{i=t-N}^{t} (u_i^{In} - u_i^{Out})^2 + (v_i^{In} - v_i^{Out})^2$;

$cos(\alpha) \leftarrow \vec{b}^{In} . \vec{b}^{Out}$;

$c_t \leftarrow \frac{cos(\alpha)}{1 + ln(1 + ssd_t)}$;

end

4 EVALUATION

We evaluated the proposed approaches for quantifying the similarity of the gaze and target trajectories using the same dataset that Esteves et al. employed for defining the system parameters in *Orbits* [Esteves et al. 2015b]. In this dataset, 12 participants performed tasks where they were asked to follow a target moving on a circular trajectory, to read the time while ignoring such moving target, to read a news article, to play a video game, and to watch a video. In the target-following task, the targets moved along trajectories of three different sizes and with three different speeds. The authors found the optimal design to be the one with the largest trajectory diameter (2.6cm/2.63^o) and the medium speed (.33Hz), so we only

ALGORITHM 5: Rotate and Scale (**rot_scale**)

Input: Two N sized arrays representing x and y coordinates of a set of 2D points: $P_x = \{x_i\}$ and $P_y = \{y_i\}$.

Output: Two N sized arrays Q_u and Q_v representing u and v coordinates of input points after transformation. The transformation centres each $\mathbf{p_i} = (x_i, y_i)$ at (x_0, y_0), rotates it along the point set's baseline, and scales it by the inverse of the baseline length; also the normalised baseline vector $\vec{b}$ is returned.

$\mathbf{p_0} \leftarrow (x_0, y_0)$;
$\mathbf{p_m} \leftarrow (x_j, y_j)$ such that $|\mathbf{p_m} - \mathbf{p_0}|$ is maximum;

$$s \leftarrow \frac{1}{|\mathbf{p_m} - \mathbf{p_0}|};$$

$\vec{b} \leftarrow s(\mathbf{p_m} - \mathbf{p_0})$;
$\theta \leftarrow -atan2(b_y, b_x)$;

$$\mathbf{R} \leftarrow \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix};$$

$Q_u \leftarrow \emptyset$; $Q_v \leftarrow \emptyset$;
for $i \in 0..(N-1)$ **do**
  $\mathbf{q_i} = (u_i, v_i) \leftarrow s(\mathbf{R} \times (\mathbf{p_i} - \mathbf{p_0}))$;
  $Q_u \leftarrow Q_u \cup u_i$; $Q_v \leftarrow Q_v \cup v_i$;
end

perform our analysis on the data in this condition. For a full description of the data collection procedure, see Esteves et al. [Esteves et al. 2015b].

We segmented the time series to split it into *steps*. A step is an uninterrupted data segment with the same experimental setup conditions, meaning that every time the eye tracker lost track of the user's eyes or a new task began, we created a new segment. This yielded a total of 50 steps where the user was supposed to be following the target and 125 steps where they were not. We use the task instruction as the ground truth label. There was no overlapping data between consecutive steps. We smoothed out the gaze data in each step with a 5-sample median filter.

In conditions without a visible target (i.e. reading text, watching video, and playing video game), we simulated a moving target with the same characteristics. In each step, we computed the similarity between the gaze and target trajectories using the algorithms described in the previous section in 30-sample windows (1s with a 30Hz eye tracker), as suggested by [Esteves et al. 2015b].

To quantify the performance of an approach, we first computed the corresponding similarity metric for all windows in a step (with one-sample overlaps), in every step. We then decided which windows get activated in each step for a range of thresholds in 0.005 increments. For this purpose, we used both a simple threshold and a bi-level threshold with a difference of .1 between the upper and lower level. Finally, we computed the true- and false-positive rates based on the number of steps that had any activated windows within it for the same range of thresholds. Ideally, a step where the user is following the target should have at least one activation, and a step where they are not should have none. To simplify the reporting of the trade-off between true and false positives, we present the results in two cases of practical interest—TP5 is the highest true positive rate for a threshold that yields a false positive rate below 5%; and FP90 is the lowest false positive rate for a threshold that yields a true positive rate above 90%. To give a more complete picture of this trade-off, we also present the ROC curve with the results for the best configuration of the four metrics. Finally, we computed these results after applying a post-hoc filter, which only triggered an activation after 30 consecutive activated windows (with 1-sample overlaps). To evaluate the effect of the post-hoc filter window size, we picked the best metric and computed the ROC curve for a range of window sizes between 0 and 60 samples long, in 5-sample increments.

Metric	Thresholding	Post-Hoc	TP5	FP90
BC	Both	No	**.92 (.91)**	**.05 (.91)**
	Simple	Yes	.80 (0.37)	.07 (.21)
	Double	Yes	.84 (.46)	.07 (.26)
RC	Both	No	.80 (.94)	.11 (.92)
	Simple	Yes	.96 (.17)	.00 (.63)
	Double	Yes	**.96 (.27)**	**.00 (.73)**
PM	Both	No	**.90 (.64)**	**.03 (.65)**
	Simple	Yes	.90 (.28)	.05 (.28)
	Double	Yes	.90 (.35)	.05 (.35)
2D	Both	No	.86 (.66)	.10 (.61)
	Simple	Yes	.96 (.01)	.00 (.07)
	Double	Yes	**.96 (.08)**	**.00 (.18)**

Table 1: True positive rate corresponding to a false positive rate below 5% (TP5) and false positive rate corresponding to a true positive rate above 90% (FP90) with corresponding coefficient thresholds.

5 RESULTS

Table 1 shows the results from our experiment, broken down by the similarity metric (BC - Basic Correlation, RC - Rotated Correlation, PM - Profile Matching, 2D - 2D Correlation), by thresholding method (Simple or Double—in the cases without the post-hoc filter, the results are the same for either thresholding approach), and by whether a post-hoc filter was applied. The number between parentheses corresponds to the threshold used. When using a bi-level threshold, the number corresponds to the upper threshold (the lower threshold is equal to 0.1 less than the upper threshold). These results are also presented graphically in Figures 3 and 4.

The best performing algorithms were the Rotated Correlation with post-hoc filtering, and the 2D Correlation with post-hoc filtering, regardless of the thresholding procedure used for both. In these conditions, we achieved a 96% true-positive rate (with a false-positive rate under 5%) and a 0% false-positive rate (with a true-positive rate above 90%). Figure 5 (left and center) shows the ROC curve with the true-positive/false-positive trade-off for the best configuration of each method with and without the post-hoc filter. The curve shows that despite the TP5 and FP90 rates being the same for the 2D and Rotated Correlations, the 2D Correlation yielded a curve closer to the upper left corner of the Cartesian space, and

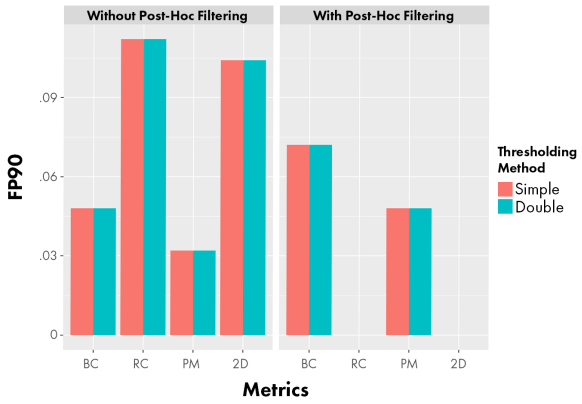

Figure 3: FP90 rates for each metric broken down by thresholding procedure and whether a post-hoc filter was applied. Though the Rotated Correlation presents the worst performance without a post-hoc filter, the false positives practically disappear when it is applied.

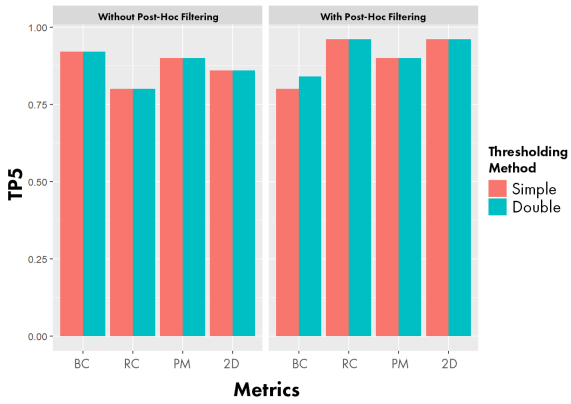

Figure 4: TP5 rates for each metric broken down by thresholding procedure and whether a post-hoc filter was applied. Though the Profile Fitting Correlation performs well when no post-hoc filter is applied, it is outperformed by the Rotated Correlation when it is applied.

therefore outperformed the Rotated Correlation. The 2D Correlation and Rotated Correlation were followed by the Profile Matching, and the Basic Correlation. For the Profile Matching, the observed performance remained almost the same regardless of the use of the post-hoc filter (just a slight increase in the FP90 rate, with TP5 remaining the same). For the Basic Correlation, performance was in fact worse when the filter was used.

Our inspiration for using a bi-level threshold came from hand gesture recognition [Negulescu et al. 2012]. In *PathSync*, users had to match the movement of the target with their hand [Carter et al. 2016]. As a consequence, maintaining a high correlation would lead to user fatigue and consequently a drop in the similarity metric. Though in such cases, a bi-level threshold can help, contrary to our previous beliefs and to what Carter et al. suggested [Carter et al. 2016], it had little effect on our results. We hypothesise that the reason for this is that the eyes do not tire as easily as the hands when following a moving target, and therefore, a drop in the similarity metric only happens when the user stops following it.

The post-hoc filter, however, had a substantially positive impact on the recognition performance for two metrics (Rotated Correlation and 2D Correlation). By discarding a large number of false positives, the filter led to an average increase in the TP5 rate of 16% and to a zero FP90 rate. The fact that by configuring these algorithms appropriately it is possible to completely remove false positives is particularly impressive given that the negative steps were substantially longer than the positive steps as the dataset had a lot more negative samples than positive ones ($N_{False} = 31,768; N_{True} = 8,613$).

We also analysed the effect of the number of activated windows necessary for triggering an action (i.e. the post-hoc filter window). Figure 5-Right shows the ROC curve that illustrates this trade-off for the 2D Correlation algorithm, showing that the best performance is achieved for 30-40-sample windows.

6 DISCUSSION

The task instruction was used as ground truth label for the dataset. Because no check was performed to verify whether the participant was actually following the target, we assume they followed the instructions. Though the high performance results suggest they indeed followed these instructions, it is possible that a certain amount of classification error can be explained by the participant not having performed the task.

Two phenomena observed in the results deserve further discussion: the small effect observed for the double threshold, and the fact that the post-hoc filter did not improve TP5 and FP90 rates for all four evaluated metrics.

We hypothesised that because the quality of an eye movement is not expected to change in a short time frame (when compared, for instance, with hand gestures), double thresholding does not lead to significant improvements. Another possible explanation is that more fine-tuning is required for selecting the difference between thresholds. The strategy we used was to define a lower threshold 0.1 less than the upper threshold. It is possible that this variation is not enough to accommodate a sequence of oscillating correlation values (computed for a sequence of windows) that can be caused by several factors such as noise, or inability of some metric to yield a good response at every portion of the circular target trajectory. Moreover, a variation between the upper and lower thresholds that would be considered good for one metric, might not be appropriate for all metrics. Thus, further investigation on how to define the thresholds should be conducted to better understand the behaviour of this approach.

Results also showed that post-hoc filtering had different outcomes for each metric. The diverse nature of the metrics considered in this paper can explain this. Though the output of the algorithms have been normalised to the [-1, 1] range, each method presents different probability distributions. Changes in threshold values affect the rate of true positives and false positives in different ways for each algorithm. For instance, lowering the threshold for one

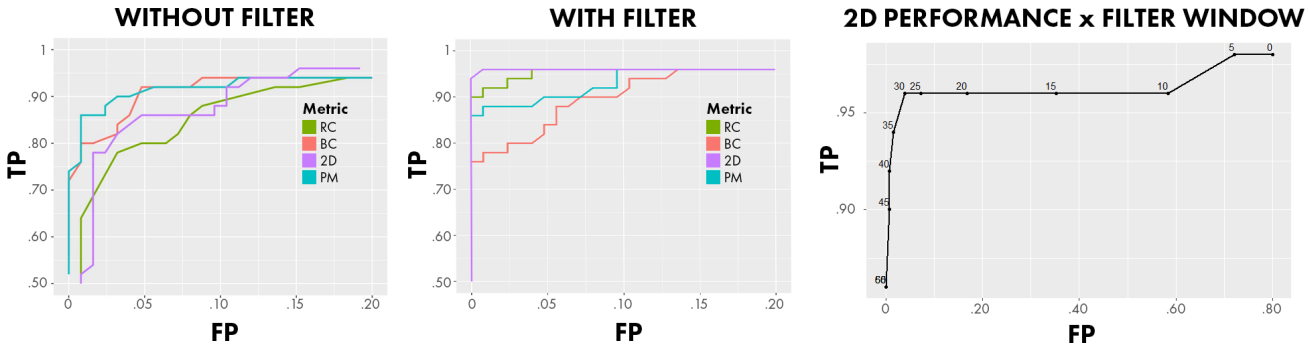

Figure 5: ROC Curve representing the trade-off between true- and false-positives as we vary the algorithm parameters. The first two figures show the ROC curves for the four metrics without and with the post-hoc filter, as we vary the decision threshold. The third curve shows the effect of the post-hoc filter window size on the performance of the 2D Correlation algorithm. Filters that wait for 30-40 activated windows offer the best performance and a similar behaviour was found for the other metrics.

method might increase the number of false positives while having little effect on the number of true positives. For some other metric, both rates may be affected in a more similar way.

The purpose of post-hoc filtering is to increase robustness in target activation by avoiding spurious high correlations that exceed a threshold in a single window, but are not sustained for a given number of consecutive windows. By being more strict, the filtering reduces both the instances of false and true positive activations. As the TP5 rate accepts a false positive ratio of up to 5%, the threshold can be lowered, accepting more false positives and potentially causing a new increase in the true positive ratio. However, there is no guarantee that it will match the true positive ratio achieved before filtering. As shown by our results, the performances of RC and 2D are improved, while for the other metrics, BC's performance is worsened and PM remains about the same. This occurs because changes in the threshold will affect each metric differently, depending on how the ratio of true positives and false positives changes as the threshold decreases.

It can be observed in the PM ROC curve without filtering (Figure 5 at left) that lowering the threshold beyond the point where a false positive ratio of 5% is observed results in a small increase in the true positive ratio and a much larger increase in the false positive ratio. Since the filter application results in a threshold reduction, there is a considerably larger increase in false positives than in true positives after the filter is applied, thus limiting its effectiveness. On the other hand, as observed for the RC and 2D metrics, lowering the threshold beyond the point where a false positive ratio of 5% is observed still causes a significant increase in the true positive ratio. Thus, we can observe that the proposed post-hoc filtering tends to work better for metrics that present lower true positive rates when applying to just a single window.

Another trade-off associated with the post-hoc filtering is relative to the activation delay. At the same time that the post-hoc filtering can be used to enhance target activation for some similarity metrics, the time required to perform the activation will be longer, which can be an issue depending on the intended application. In summary, though the post-hoc filter can be very effective in removing false positives, it also makes the system less responsive. In a scenario where faster responses are required, our proposed PM metric can be an interesting alternative. Without application of the post-hoc filtering, the ROC curve for PM is the one closer to the upper left corner of the Cartesian space, indicating that it outperforms the other metrics in this scenario.

As a final limitation for our experiment, it is important to note that we only computed the similarity metrics on 30-sample windows, following Esteves et al.'s tests on the same dataset [Esteves et al. 2015b]. Further, the dataset itself is limited in only considering a single target, with a single trajectory shape and speed. Future work should look into further movement designs.

7 CONCLUSION

Our goal in this paper was to develop a better understanding of algorithms for implementing gaze-based motion correlation interfaces. Given the recent interest in this type of interaction, multiple techniques had been proposed in the literature, without a systematic comparison to previous approaches. This paper set out to fill this gap by comparing four similarity metrics, two thresholding procedures, and one post-hoc filter for triggering activations. We conducted an experiment on Esteves et al.'s [Esteves et al. 2015b] dataset and found that an algorithm that uses the 2D Correlation as a similarity metric and filters out false positives using a post-hoc filter with a window of 30 samples achieved the best performance.

These results are important considering that the large majority of systems published that employ selection by motion correlation use the metric that we call Basic Correlation, which in our experiment was outperformed by all alternatives.

ACKNOWLEDGMENTS

The authors would like to thank the financial support from the São Paulo Research Foundation (FAPESP), grants No. 2016/10148-3 and No. 2017/50121-0. This work was partly supported by the FAPESP-University of Melbourne SPRINT Grant and by a University of Melbourne Early Career Researcher Grant.

REFERENCES

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. https://doi.org/10.1145/2858036.2858284

Christopher Clarke, Alessio Bellino, Augusto Esteves, and Hans Gellersen. 2017. Remote Control by Body Movement in Synchrony with Orbiting Widgets: An Evaluation of TraceMatch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 45 (Sept. 2017), 22 pages. https://doi.org/10.1145/3130910

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. https://doi.org/10.1145/2971648.2971714

Christopher Clarke and Hans Gellersen. 2017. MatchPoint: Spontaneous Spatial Coupling of Body Movement for Touchless Pointing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 179–192. https://doi.org/10.1145/3126594.3126626

Travis Cox, Marcus Carter, and Eduardo Velloso. 2016. Public DisPLAY: Social Games on Interactive Public Screens. In *Proceedings of the 28th Australian Conference on Computer-Human Interaction (OzCHI '16)*. ACM, New York, NY, USA, 371–380. https://doi.org/10.1145/3010915.3010917

Murtaza Dhuliawala, Juyoung Lee, Junichi Shimizu, Andreas Bulling, Kai Kunze, Thad Starner, and Woontack Woo. 2016. Smooth Eye Movement Interaction Using EOG Glasses. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction (ICMI 2016)*. ACM, New York, NY, USA, 307–311. https://doi.org/10.1145/2993148.2993181

Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015a. Orbits: Enabling Gaze Interaction in Smart Watches Using Moving Targets. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers (UbiComp/ISWC'15 Adjunct)*. ACM, New York, NY, USA, 419–422. https://doi.org/10.1145/2800835.2800942

Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015b. 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. https://doi.org/10.1145/2807442.2807499

Augusto Esteves, David Verweij, Liza Suraiya, Rasel Islam, Youryang Lee, and Ian Oakley. 2017. SmoothMoves: Smooth Pursuits Head Movements for Augmented Reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 167–178. https://doi.org/10.1145/3126594.3126616

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. https://doi.org/10.1145/1518701.1518748

Jari Kangas, Oleg Špakov, Poika Isokoski, Deepak Akkil, Jussi Rantala, and Roope Raisamo. 2016. Feedback for Smooth Pursuit Gaze Tracking Based Control. In *Proceedings of the 7th Augmented Human International Conference 2016 (AH '16)*. ACM, New York, NY, USA, Article 6, 8 pages. https://doi.org/10.1145/2875194.2875209

Mohamed Khamis, Florian Alt, and Andreas Bulling. 2015. A Field Study on Spontaneous Gaze-based Interaction with a Public Display Using Pursuits. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers (UbiComp/ISWC'15 Adjunct)*. ACM, New York, NY, USA, 863–872. https://doi.org/10.1145/2800835.2804335

Mohamed Khamis, Ozan Saltuk, Alina Hang, Katharina Stolz, Andreas Bulling, and Florian Alt. 2016a. 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. https://doi.org/10.1145/2971648.2971679

Mohamed Khamis, Ludwig Trotter, Markus Tessmann, Christina Dannhart, Andreas Bulling, and Florian Alt. 2016b. EyeVote in the Wild: Do Users Bother Correcting System Errors on Public Displays?. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM '16)*. ACM, New York, NY, USA, 57–62. https://doi.org/10.1145/3012709.3012743

Matei Negulescu, Jaime Ruiz, and Edward Lank. 2012. A Recognition Safety Net: Bi-level Threshold Recognition for Mobile Motion Gestures. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 147–150. https://doi.org/10.1145/2371574.2371598

Joshua Newn, Eduardo Velloso, Marcus Carter, and Frank Vetere. 2016. Multimodal Segmentation on a Large Interactive Tabletop: Extending Interaction on Horizontal Surfaces with Gaze. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 251–260. https://doi.org/10.1145/2992154.2992179

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. https://doi.org/10.1145/2501988.2501998

Junichi Shimizu, Juyoung Lee, Murtaza Dhuliawala, Andreas Bulling, Thad Starner, Woontack Woo, and Kai Kunze. 2016. Solar System: Smooth Pursuit Interactions Using EOG Glasses. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)*. ACM, New York, NY, USA, 369–372. https://doi.org/10.1145/2968219.2971376

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. https://doi.org/10.1145/3064937

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. https://doi.org/10.1145/2901790.2901867

David Verweij, Augusto Esteves, Vassilis-Javed Khan, and Saskia Bakker. 2017a. Smart Home Control Using Motion Matching and Smart Watches. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 466–468. https://doi.org/10.1145/3132272.3132283

David Verweij, Augusto Esteves, Vassilis-Javed Khan, and Saskia Bakker. 2017b. WaveTrace: Motion Matching Input Using Wrist-Worn Motion Sensors. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 2180–2186. https://doi.org/10.1145/3027063.3053161

David Verweij, Vassilis-Javed Khan, Augusto Esteves, and Saskia Bakker. 2017c. Multi-User Motion Matching Interaction for Interactive Television Using Smartwatches. In *Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video (TVX '17 Adjunct)*. ACM, New York, NY, USA, 67–68. https://doi.org/10.1145/3084289.3089906

Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2012. Detection of Smooth Pursuits Using Eye Movement Shape Features. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 177–180. https://doi.org/10.1145/2168556.2168586

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. https://doi.org/10.1145/2493432.2493477

John Williamson. 2006. *Continuous uncertain interaction.* Ph.D. Dissertation. University of Glasgow.

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. https://doi.org/10.1145/985921.986076