

# A Probabilistic Interpretation of Motion Correlation Selection Techniques

Eduardo Velloso  
eduardo.velloso@unimelb.edu.au  
The University of Melbourne  
Victoria, Australia

Carlos Hitoshi Morimoto  
University of São Paulo  
São Paulo, Brazil  
hitoshi@ime.usp.br

## ABSTRACT

Motion correlation interfaces are those that present targets moving in different patterns, which the user can select by matching their motion. In this paper, we re-formulate the task of target selection as a probabilistic inference problem. We demonstrate that previous interaction techniques can be modelled using a Bayesian approach and that how modelling the selection task as transmission of information can help us make explicit the assumptions behind similarity measures. We propose ways of incorporating uncertainty into the decision-making process and demonstrate how the concept of entropy can illuminate the measurement of the quality of a design. We apply these techniques in a case study and suggest guidelines for future work.

## CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**; *HCI design and evaluation methods*; *HCI theory, concepts and models*.

## KEYWORDS

motion correlation, pursuits, computational interaction, probabilistic input, gestures, gaze interaction

### ACM Reference Format:

Eduardo Velloso and Carlos Hitoshi Morimoto. 2021. A Probabilistic Interpretation of Motion Correlation Selection Techniques. In *CHI Conference on Human Factors in Computing Systems (CHI '21)*, May 8–13, 2021, Yokohama, Japan. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3411764.3445184>

## 1 INTRODUCTION

Though the dominant interaction paradigm in current gestural user interfaces is still largely based on deictic and manipulative gestures, a different kind of interaction has been gathering interest in recent years—that of *motion correlation* [28]. The principle underlying this type of interaction relies on the interface presenting moving targets, each with a distinct motion pattern, which the user can mimic in order to signal the intention to select the desired target [28]. The system then measures the similarity between the signal originating from the input device and the signals corresponding to the targets

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*CHI '21*, May 8–13, 2021, Yokohama, Japan

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8096-6/21/05...\$15.00

<https://doi.org/10.1145/3411764.3445184>

moving on the screen in order to determine which target (if any) is being followed.

The literature contains a wealth of examples that successfully employ the principle, including for gaze interaction with smart watches [10, 11], public displays [18, 32], virtual reality [17], and smart homes [30]; for manual control in body-based games [2, 6] and smart TVs [3, 4, 31]; for head control of head-up displays [12]; for one-handed ring-based input [36]; for menu selection with mice [13, 35]; among others.

Despite building upon the same principle, these works employ a wide variety of input devices, data processing pipelines, similarity measures, feedback mechanisms, and evaluation procedures. As a consequence, it is challenging to compare approaches and results in order to advance the state-of-the-art of motion correlation interaction. The standard analysis approach in related work is empirical, instantiating different designs and collecting performance data (e.g. precision, recall, selection times, and error rates) in user studies. The problem with this is that the design space of motion correlation interfaces is enormous—not only there are many design parameters to be manipulated, but these parameters are not necessarily independent.

In this paper, we propose an interpretation of motion correlation based on probability and information theory, inspired by the early work conducted by Williamson and Murray-Smith [34, 35] and aligned with current trends in computational interaction [23]. As such, we offer the following contributions: (1) We demonstrate that previous works on motion correlation can be modelled as a probabilistic reasoning problem; (2) we demonstrate a series of probabilistic techniques for better understanding combinations of interface design decisions; (3) we offer a case study of the application of these techniques in the context of a motion correlation technique, *Orbits* [10]. We argue that such account offers a holistic approach for understanding motion correlation and a consistent language for framing future work and advancing progress on the topic.

## 2 RELATED WORK

### 2.1 Motion Correlation Interfaces

A motion correlation interface presents the user  $n$  targets, each moving according to its own characteristics, such as trajectory, velocity, phase, and direction. The user signals their intention to select a target by matching the motion of the desired target. When the system detects that the user has been following one of the targets, the selection of the corresponding target occurs. The main advantage of this technique is that users can signal their intention to select a target by mimicking its motion with their eyes, hand,

head, or finger, without the need of an explicit pointing device or selection confirmation mechanism. Previous works have demonstrated a wealth of opportunities for motion correlation interfaces, including enabling calibration-free gaze interaction [30, 32], implicitly calibrating eye trackers [14, 18, 24], enabling cursorless multi-user gestural interaction [2], and enabling interaction with small displays [10, 11].

The central challenges for a motion correlation system are to detect *whether* the user is trying to select a target, and if so, *which* target they are trying to select. To accomplish this, system designers must carefully choose a set of parameters that together determine the performance of the system. These parameters include the number of targets, their motion characteristics (e.g. deterministic vs. random, path shape, velocity, acceleration, direction, etc.), the input modality and device (e.g. eye trackers, RGB cameras, depth cameras), the feedback modality and device (e.g. screens, projections, mechanical actuators), the similarity measure used to compare the projection of the user movements onto the input space with the target motions (e.g. Euclidean distance, Pearson’s correlation, frequency analysis, etc.), the amount of data being compared (i.e. the number of samples within a window), the decision criteria for determining when a selection happens (e.g. the correlation threshold), among many others (see Velloso et al. for an overview of these parameters [28]).

The large number of parameters creates a rich design space, with many opportunities for building novel interfaces. The downside is that thoroughly exploring and understanding this design space becomes a challenge without a systematic process for comparing alternatives. Previous works have taken an empirical approach for such exploration—designing interfaces with different combinations of parameters and evaluating them through user studies. However, recruiting participants is expensive and time-consuming. In this work, we propose techniques for early design space exploration that enable us to analyse design decisions a priori.

## 2.2 Probabilistic Interaction

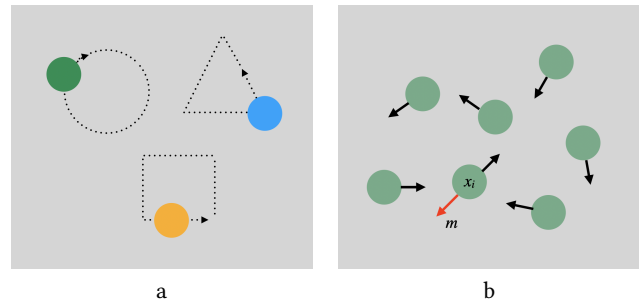
Any interaction with a system involves a level of uncertainty, including factors such as noise, user capabilities, and sensor limitations. Probabilistic interaction techniques, in contrast to conventional deterministic ones, explicitly model this uncertainty to enhance and improve the interaction. They typically draw from probability theory, information theory, control theory, machine learning, and other related areas.

For example, Rogers et al. [25] employed a particle filtering approach to improve finger touch estimation using low resolution capacitive sensing arrays, and Biswas et al. [1] used a Kalman filter algorithm to estimate the cursor position and the probability distribution of possible targets to improve the performance of pointing tasks for people with physical impairments. These approaches have in common the modelling of likelihoods of possible user states and use this to improve the interaction.

A more general framework for handling input with uncertainty was proposed by Schwarz et al. [26]. The framework tracks the probabilities of alternative inputs, provides a mechanism to dispatch the input to appropriate interactors (or interface elements), and allows the interactors to provide feedback (mediation) or other

appropriate actions to resolve ambiguous input cases. Another general framework based on information theory for modelling interaction is that of *Bayesian Information Gain*, which has been demonstrated in applications such as pan and zoom navigation and file retrieval [20, 21]. The approach interprets every user action as clue for what the user is interested in, updating a probabilistic model of the information space. The system can then update its visualisation in order to maximise the expected information gain of the system. We apply ideas inspired from these approaches to the specific problem of selection through motion correlation.

In the context of motion correlation we consider the literature about probabilistic methods used in gestural user interfaces. A gesture can be represented as a temporal sequence of measurement samples provided by at least one position or motion sensor, for example. A particular gesture can be recognised by matching the shape trajectory described by the user’s motion to expected known shapes, such as seen in Figure 1a. Sensor noise and personal motion traits are possible sources variations in the path obtained by the input device that are challenging for any gesture recognition system. Statistical shape analysis methods [9] have been designed to handle such variations in the interpretation of gestures. Active shape and appearance models are examples of such techniques used in computer vision to find objects (such as hands and faces) in images [5] and for recognition of handwritten characters [27].



**Figure 1:** a) The basic motion correlation idea selects a target moving along a known trajectory using some similarity measure; and b) shows the idea of the pointing without a pointer method proposed by Williamson and Murray-Smith. Black arrows indicate the expected motion of each target. To select target  $x_i$  the user must stabilise it moving the mouse in the opposite direction  $m$ .

Previous motion correlation methods have mostly taken a deterministic approach for detecting target selection. Typically, they measure the similarity between the projection of the user’s movements onto the input space and each target’s motion; then, if the similarity crosses a given threshold, the target is selected.

A notable exception was the “pointing without a pointer” technique proposed by Williamson and Murray-Smith [35]. Their idea was to present the user several targets moving according to a pseudo-random disturbance. An arrow attached to each target shows the user the direction where the target is moving to due to the disturbance. To select a target, the user must try to stabilise its motion by moving the mouse in the opposite direction of the

target’s arrow, as illustrated in Figure 1b. The position of a target  $x_i$  at time  $t$  is computed by adding the mouse motion  $m$  to the disturbance for  $x_i$ . The variance of this sum,  $\sigma_{si}$ , is computed over a time window  $w$ . The variance of the disturbance applied to each target,  $\sigma_{fi}$  is also used to compute the ratio  $r_i = \sigma_{si}/\sigma_{fi}$ . A small  $r_i$  can be used as evidence that the user is trying to select  $x_i$ .

Williamson and Murray-Smith[35] define a threshold  $v < 1$  so that, when this evidence is strong, the probability that the user is trying to select  $x_i$  increases, and decreases otherwise. Starting from an even prior for all targets, when  $r_i < v$ , they keep track of an intermediary weight value for each target, which increases additively by  $\alpha \times r_i$ , and is attenuated multiplicatively by  $\beta$  otherwise. These intermediate values are then normalised to produce the new probabilities and, when the resulting entropy falls below a threshold, the target  $i$  with the highest  $p(x_i)$  is selected. The choices of  $\alpha$ ,  $\beta$ , and  $v$  affect how quickly and how accurately the system decides that the user is making a selection.

In comparison to deterministic approaches, Williamson and Murray-Smith’s probabilistic approach has several advantages. The first is that it considers the probability of the targets jointly and provides a measure of uncertainty that can be used to indicate the most probable target when the uncertainty is low, i.e., when the entropy of the system is sufficiently small. The second is that it allows the evidence from the measurements to be integrated in a probabilistic update process as a function of prior probabilities. A third advantage compared to typical probabilistic gesture-based methods is that it neither requires training nor a specific shape model. Despite this early work, works in motion correlation since then have not used probabilistic techniques for inferring which target the user is trying to select.

In this work, we take direct inspiration from the pioneering work of Williamson and Murray-Smith, abstracting it to model later approaches and extending it to enable the analysis and design of new interfaces. Our work is framed within the larger program of *computational interaction*, which “focuses on the use of algorithms and mathematical models to explain and enhance interaction” [23]. Core to computational interaction is mathematical modelling, typically involving ways for updating the model using data observed from the user [23]. In this paper, we model motion correlation using a probabilistic approach, which enables us to derive a set of techniques for examining motion correlation designs before collecting user data.

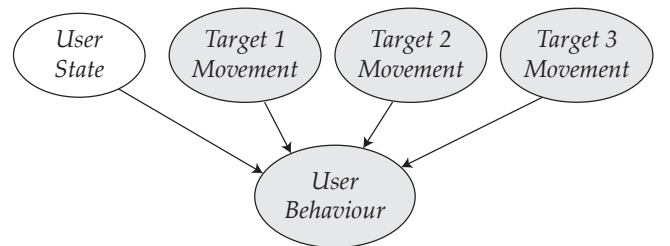
### 3 PROBABILISTIC MODELLING OF THE SELECTION TASK

Our thesis in this paper is that we can better understand the principles and make more informed and confident interface design decisions by looking at motion correlation interfaces from a probabilistic perspective. As such, our goal is to bring ideas from probabilistic interaction to researchers and practitioners working on the development of motion correlation interfaces. In this and the next few sections, we offer a didactic primer on probabilistic input and demonstrate how previous motion correlation techniques can be formulated in probabilistic terms. As we discuss how to model deterministic techniques as probabilistic, we formalise a set

of principles that are useful for the design of motion correlation interfaces.

In addition, we offer a practical example of how to employ the techniques we propose in the case study of *Orbits* [10], a gaze-only motion correlation technique for interaction with smart watches. The design of *Orbits* was done through an empirical process, collecting user data at each stage of the development. Here, we demonstrate the additional insights that are gained by analysing these interactions through a probabilistic framework.

To begin, we demonstrate how to formalise the motion correlation selection task in a notation that allows us to discuss it in probabilistic terms. The basic challenge in any selection task is to discover the user’s state (e.g. the intention to select a target, if any) given a system state (e.g. a graphical user interface with moving targets) and a sequence of user behaviours (e.g. matching the motion of the intended target or natural behaviour when not attempting to make a selection). From a probabilistic perspective, we can model these as random variables, where the user behaviour is influenced by the user state and the system state. If we break the system state into separate random variables for each target, we can build the Bayesian network in Figure 2.



**Figure 2: In a motion correlation interface the user movement is influenced by the user state and by the system state, represented here by each separate target’s motion. Observed variables are in grey.**

More formally, the user state is a random variable with a sample space containing the intention to select each target ( $x_i$ ) and the intention to not select any of them ( $x_0$ ).

$$User\ State = X \in \{x_0, x_1, \dots, x_N\} \quad (1)$$

The target motions ( $T_i$ ) and the user behaviour ( $U$ ) are sequences of two-dimensional coordinates, not necessarily in the same coordinate system—e.g. the input device might not be calibrated to the output device.

Like all Bayesian networks, the model in the figure encodes a joint distribution over the modelled variables. Missing edges reflect independence assumptions that are necessary to make probabilistic inferences efficient in runtime (computationally) and in observations needed for model fitting (statistically). In this case, we know that the variables  $X, T_1, \dots, T_N$  are pairwise independent by construction of the target motions.

During the interaction, the target motions and the user behaviour are our *evidence* variables (denoted in the figure in grey), as they are directly observable, whereas the user state is our *query* variable (denoted in the figure in white), which we would like to infer. This

is specified as the following conditional probability, which we can unpack with Bayes' rule.

$$P(X|U, T_1, \dots, T_N) = \frac{P(U|X, T_1, \dots, T_N)P(X)}{P(U, T_1, \dots, T_N)} \quad (2)$$

The key term in this equation is the likelihood  $P(U|X, T_1, \dots, T_N)$ . This term computes for each possible user state, the probability of observing a given user behaviour given the motions available on the interface and the user state. A well-designed motion correlation interface displays motions that elicit sufficiently unusual user movements—decreasing  $P(U|x_0, T_1, \dots, T_N)$ —and that could only happen given that the motion is displayed on the interface—increasing  $P(U \approx T_i|x_i, T_1, \dots, T_N)$ , while balancing subjective user experience factors such as comfort, effort, and social acceptability. The challenge lies in how to compute this likelihood.

The motions of the targets should also be sufficiently different from each other, so that  $P(U \approx T_j|x_{i \neq j}, T_1, \dots, T_N)$ , that is, the likelihood of observing a user movement that similar to target  $j$  when trying to select target  $i$  should be small. Such difference can be created a priori by through the path design or in real-time, such as in Williamson and Murray-Smith's decorrelation approach [35]. In their approach, as the probability of a target increases, so does the range of the disturbances on the target's motion characteristics (e.g. path, speed, etc.), to make the motion of the target even more distinct than the others, as if testing the hypothesis that the user is indeed trying to select it.

With this framing, we can introduce a formal notation to known principles of motion correlation design.

**Principle 1: Motions in the interface should elicit a similar movement from the user**

We aim to maximise  $P(U \approx T_i|x_i, T_1, \dots, T_N)$ —the likelihood that given a set of target motions and given that the user is trying to select target  $i$ , the movement observed from the user should be similar to the motion of target  $i$ .

**Principle 2: Targets should elicit motions distinct from non-interaction behaviour**

The motions of the targets on a motion correlation interface should not elicit movements that frequently occur in the user's natural behaviour when they are not interacting with the system. Formally, we aim to minimise  $P(U \approx T_i|x_0, T_1, \dots, T_N)$ —the likelihood of observing a user movement similar to a target motion when the user is not interacting with the system.

**Principle 3: Targets should display motions distinct from each other**

We aim to minimise  $P(U \approx T_j|x_{i \neq j}, T_1, \dots, T_N)$ —the likelihood of observing a user movement similar to target  $j$  when the user is trying to select a different target  $i$ .

### 3.1 Case Study: Modelling Orbits

In their evaluation of the performance of *Orbits*, Esteves et al. measured the effects of the number of targets (2, 4, 8, 16), trajectory size ( $4.25^\circ$ ,  $2.62^\circ$ , and  $0.98^\circ$ ) and target speed ( $120^\circ/s$ ,  $180^\circ/s$ , and  $240^\circ/s$ ) on the true- and false-positives in a selection task [11]. In all conditions, they used a fixed window size of 1s, a 30Hz eye tracker, and a Pearson's correlation threshold of 0.8. The targets moved along a circular trajectory, with half of the targets moving in the opposite direction of the other half. This is representative of the types of evaluation procedures found in previous works.

In the case of *Orbits*, the user state is the desire to select a target or not, and their behaviour is captured by the system as  $X$ ,  $Y$  coordinates provided by the eye tracker. Because the user most often looks straight at the screen, it is expected that when following a target, the user's eyes will follow a smooth pursuit movement that closely resembles the motion of the target (Principle 1). The head-mounted eye tracker employed in *Orbits* captures data even when the user is not interacting with the system, so we must consider that the  $x_0$  state will be over-represented—that is, most of the time, the user will not be interacting with the system. However, because the eyes only engage in smooth pursuits when there is a moving target for them to track, if the system is able to accurately detect whether the eyes are in this state, Principle 2 should be satisfied. The targets move around circular trajectories. Differently from shapes with straight edges, no two windows in a circle are the same, so all targets will exhibit a different motion at all points in time (satisfying Principle 3).

## 4 INTERACTION AS TRANSMISSION OF INFORMATION

The discussion above emphasises that ultimately, we seek to compute likelihoods. The novel insight in the motion correlation literature was that we can estimate these likelihood from the similarity between the motion of the target and the movement that user is making as captured by the input device. We can model this interaction from an information theory perspective, as *transmission of information* [16]. From this perspective, when the user is attempting to select a target, that target motion is like a message passing through a noisy channel and manifesting itself as the user behaviour. As such, the measurements  $U(t)$  can be seen as a distorted version of the selected target trajectory (or none). These distortions might include noise, delays, and spatial transformations introduced by the sensors, processing hardware, geometric setup, and user movement inaccuracies. We represent these distortions as a spatial transformation matrix  $A$ , a time delay  $\tau$ , and a noise component  $\epsilon$ . Note that here we assume linear distortions, where distortions found in the real world, such as those introduced by camera lenses, might not be linear. More complex distortions require more sophisticated modelling approaches or different assumptions made by the interaction technique.

$$X = x_i \implies U(t) = A \cdot T_i(t - \tau) + \epsilon(t) \quad (3)$$

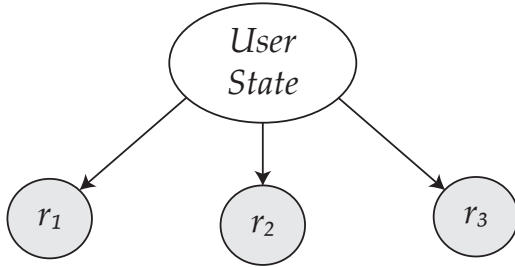
By treating the interaction process as transmission of information, we can detect the evidence of whether a selection is taking place by measuring how much information is being transmitted. The *mutual information*—the amount of information you gain about

one variable by learning the value of another [22]—can then be used as evidence that the interaction is taking place and that a selection is being attempted. This consolidates Principles 1–3 into a unified principle:

**Principle 4: The intent to select is evidenced by the amount of mutual information between the target and user behaviours**

$P(x_i) \propto I(U; T_i)$ —that is, the higher the mutual information between the motions of the user and the target, the higher the likelihood that the user intends to select that target.

The insight that mutual information is the key idea behind the selection means that the term “correlation” in “motion correlation” should be understood in the broad English sense of the word rather than as Pearson’s correlation coefficient. In practice, however, computing the mutual information is challenging [33], so we must estimate it using other *similarity measures*, such as Pearson’s product-moment correlation coefficient, as used in Vidal et al.’s *Pursuits* [32]. We denote this similarity measure as  $r(U, T_i) = r_i$ . Therefore, instead of computing  $P(X|U, T_1, \dots, T_N)$ , we can compute  $P(X|r_1, \dots, r_N)$ , which is much easier. Our network then becomes the one shown in Figure 3.



**Figure 3: By measuring the similarity between the user movement and the motion of each target, we can more easily compute the probability that the user is in a given state. Observed variables are in grey.**

Besides Pearson’s coefficient, many other similarity functions have been used in previous work. We have already described how Williamson and Murray-Smith used a ratio of variances as their similarity function [35]. Fekete et al. proposed several measures computed on the path derivatives, including the Euclidean distance, the normalised Euclidean distance, and Pearson’s correlation, ultimately opting for the normalised Euclidean distance [13]. Carter et al. also used Pearson’s correlation, but prior to computing the coefficient, they rotated the data to ensure an even distribution of variance between the horizontal and vertical axes [2]. Velloso et al. proposed a version of Pearson’s correlation that considered the variance in both axes simultaneously [29]. Drewes et al. used the linear regression slope [8]. Zhang et al. compared the dominant frequencies of the signal [36].

The choice of similarity measure depends on the assumptions about the data. For example, if we assume no calibration between

the input device and the display—i.e. we do not know the transformation matrix  $A$ —it becomes difficult to find appropriate thresholds for using the linear regression slope or anything based on Euclidean distances. Though measures based on Pearson’s correlation work well in these cases, they are susceptible to statistical factors known to affect this coefficient, such as the distributions of the data being compared, the variability of the data, the lack of linearity, the presence of outliers, etc.—for a review, see Goodwin and Leech [15]. Among them, one that is of particular relevance to the design of motion correlation interfaces is the fact that it expects a similar distribution (preferably normal) in the samples being compared. This is difficult to guarantee with regular shapes, such as the square, where there might be windows with no variation in one of the coordinates when the target is traversing one of the edges. The correlation coefficient is also susceptible to the presence of outliers, and this effect is more pronounced in small windows. As a rule of thumb, Drewes et al. suggest that the window size should be set to allow the motion of the target by 3–4 times the amplitude of the measurement noise [7]. The choice of measure also depends on which properties characterise the differences between the target motions. Certain measures, like Zhang et al.’s frequency analysis [36] work best when targets display motions with different *frequencies*, whereas others, like Pearson’s correlation, work best when the targets display motions with different *phases*, but the same frequency.

Most systems assume a sufficiently small lag  $\tau$  and error  $\epsilon(t)$ , and a stable calibration matrix  $A$ . However, these can be measured in pilot tests and incorporated into the calculation of the similarity, either by shifting the windows being compared (in the case of a substantial  $\tau$ ) or by filtering the input data (in the case of a substantial  $\epsilon(t)$ ).

**Principle 5: The appropriateness of a similarity measure is determined by how well it estimates the mutual information**

Similarity measures should be chosen based on the possible distortions of the channel and how much information about the user state can be determined from the knowledge of the target states.

#### 4.1 Case Study: Orbits as Transmission of Information

Because *Orbits* was designed for interaction with smart watches while the user is wearing a head-mounted eye tracker, we cannot assume that the gaze data will be in the same coordinate system as the GUI in the watch. This means that the matrix  $A$  in Equation 3 will contain a scaling factor and that the noise component  $\epsilon(t)$  will have a constant translation offset together with a random component. We assume that the two devices will be synchronised, making  $\tau$  sufficiently small to be ignored. The user’s head and the watch might not be perfectly aligned. This is a problem, particularly in the case of targets moving around a circle, as the rotated image will lead to ambiguity. This can either be fixed by using the scene camera of the eye tracker/the IMU in the watch to re-align them or

by assuming that this alignment is a pre-condition for the interaction, so we will not assume that  $A$  contains a rotation component. These assumptions mean that certain similarity measures cannot be applied. Ones based on Euclidean distance cannot be used because they are not invariant to translation and scale [13]; one based on the linear regression slope cannot be applied because it is not invariant to scale [8]; and one based on frequency analysis would fail due to the fact that all targets move with the same frequency [36].

## 5 COMPUTING LIKELIHOODS

Previous works typically make a decision about which target to select deterministically, by simply thresholding the similarity at an empirically obtained value  $\lambda$ . We can model these approaches, by using the following conditional probability density functions.

$$p(x_i|r_i) = \begin{cases} 1 & \text{if } r_i > \lambda, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$p(x_\emptyset|r_i) = 1 - \sum_i^N p(x_i|r_i) \quad (5)$$

This formulation already exposes a few problems with this approach. First, it does not deal well with cases where more than one target cross the threshold—all targets that cross get assigned an equal likelihood of 1. In practical implementations, this is typically solved by only selecting the target with the highest similarity. Nevertheless, it would be good to be able to incorporate this ambiguity into the decision to whether make a selection and provide some form of user feedback, reflecting in the interface the uncertainty in the system (for example, Williamson and Murray-Smith showed a growing circle around the target as its likelihood increased in relation to the other targets [35] and Schwarz et al. delayed system actions that were difficult to reverse until the system was confident enough of the user intent [26]).

The second insight is that the step function lacks nuance. Particularly for values near the threshold, this can cause the probability to alternate between 0 and 1 with small perturbations in the output of the similarity measure. A smoother thresholding function could help address this problem and also assign higher probabilities to targets with a higher similarity. A simple solution is to use a probability density function with a soft threshold, such as the probit or logit distributions. These approaches do not suffer from the discontinuity at the threshold, smoothly increasing its outputs as the likelihood of the input increases. The downside of these approaches is that they still require the system designer to manually specify a mean threshold.

A potentially better approach is to collect user data to learn these distributions. To do so, we must collect data of users attempting to select targets, along with the target motion data in order to compute the similarity between them. Likewise, we must also collect data of users not trying to select any target, but rather, performing tasks appropriate for the context of the system.

For example, a video-on-demand application as featured in *AmbiGaze* should consider collecting data not only of users attempting to control the interface widgets, but also of users watching a variety of videos [30]. One can then simulate target motions to compute the similarity values against the natural behaviour data. The system

can then compute the similarity between the gaze behaviour and the true target and estimate a probability density function  $p(r_i|x_i)$ , where  $x_i$  is the true state. The other distributions  $p(r_i|x_j)$ , where  $i \neq j$ , can be computed by modifying the gaze or trajectory data (e.g. if the target motions differ in phase, this can be simulated by shifting the data). Finally, the trajectory data of the target motion can be compared to the natural behaviour dataset to generate the distribution  $p(r_i|x_\emptyset)$ .

By applying Bayes' rule, we notice a few more issues:

$$P(x_i|r_1, \dots, r_N) = \frac{P(r_i|x_i, r_1, \dots, r_{i-1}, r_{i+1}, \dots, r_N)P(x_i)}{P(r_1, \dots, r_N)} \quad (6)$$

In order to obtain the step function used in deterministic approaches, the underlying assumption in these works is that when the user is following target  $i$ ,  $P(r_i|x_i, r_1, \dots, r_{i-1}, r_{i+1}, \dots, r_N) = 1$ , and it equals zero otherwise. This implies that it is impossible to observe a correlation above the threshold if the user is not following the target and impossible to observe a correlation below the threshold when they are. Clearly this is a very strong assumption, easily disproven with empirical tests. The consequence is that the system might make very confident decisions based on flawed assumptions, leading to a higher error rate.

This formulation also reveals another issue with previous approaches. Typically, they compare each target individually and threshold the similarity without taking into account the similarity with other targets. In practice, this means that these techniques consider the following to be true

$$P(r_i|x_i, r_1, \dots, r_{i-1}, r_{i+1}, \dots, r_N) = P(r_i|x_i) \quad (7)$$

This is not necessarily a problem: by making the similarities conditionally independent given  $x_i$ , we reduce the complexity of the inference—this is the assumption that all Naïve Bayes models make. However,  $r_i$  are not necessarily independent: if the motion of a target  $T_1$  is correlated with the motion of target  $T_2$  and  $T_2$  is correlated with  $T_3$ , most likely  $T_1$  is also correlated with  $T_3$ <sup>1</sup>. The problem is that we often *only* consider  $P(r_i|x_i)$  when computing the likelihood, when in fact all other  $P(r_j|x_i)$  also give us evidence for the probability that  $P(x_i|r_1, \dots, r_N)$ .

Shifting our attention to the other terms in the equation, the denominator  $P(r_1, \dots, r_N)$  is just a normalisation factor to make the probabilities add up to 1. The prior probability  $P(x_i)$ , on the other hand, can open up opportunities for tuning the algorithm. A naïve—though standard—approach would be to set all prior probabilities to be the same, effectively making this term irrelevant for the state estimation. This is the approach underlying current deterministic approaches, even though there is no real reason to assume that all prior probabilities would be the same in every case. The beauty of probabilistic input is precisely in enabling developers to incorporate prior knowledge in a quantifiable manner.

Another approach would be to use domain knowledge or separate classifiers to estimate reasonable values. For example, a motion correlation interface based on smooth pursuits might use a separate classifier to identify when the eyes engage in a smooth pursuit and

<sup>1</sup>It is important to note that  $T_1$  and  $T_3$  will only be positively correlated if  $r(T_1, T_2)$  and  $r(T_2, T_3)$  are sufficiently close to 1, but not necessarily true otherwise, as demonstrated by Langford et al. [19]—in other words, being positively correlated is not a transitive property.

use its output as a prior. Alternatively, a gesture recognition algorithm could be used to determine the probability that the gesture has the right shape before trying to determine which target along that shape is being followed. Yet another approach would be to collect statistical data of application usage to compute these values. This is particularly valuable in cases where users exhibit consistent usage patterns, such as when typing or navigating hierarchical menus. A final possibility is using the probabilities computed at previous points in time to adjust our confidence over time. From this perspective, the Bayesian networks we presented so far can be extended into temporal inference models, leveraging techniques such as Kalman filters [1] and particle filters [25].

### 5.1 Case Study: Computing likelihood distributions

Esteves et al. used a single similarity threshold  $\lambda = .8$  to determine a selection. As we discuss above, this approach has several limitations. To demonstrate a probabilistic approach based on empirical likelihoods, we analysed the dataset collected as part of the development of *Orbits* [10]. This dataset contained data of users following circular targets with their eyes, as well as other natural gaze data, including reading the time, watching a video, playing a video game, and reading a news article. The gaze data was sampled at 30Hz and compared in windows of 30 samples.

We computed the similarity between the gaze data and the target using three metrics: the Pearson’s correlation coefficient of the lowest axis proposed by Vidal et al. [32], the modification of this measure involving a rotation of the windows to maximise the variance across axes proposed by Carter et al. [2], and the complement of the ratio of the sum of squares computed in two dimensions proposed by Velloso et al. [29]. In the cases where the user was following a target ( $X = x_i$ ), the similarity was computed between the gaze data and the target positions. In the case where the user was not following any target ( $X = x_0$ ), we simulated a target using the same characteristics of the targets in the positive condition. Figure 4 shows the probability density functions  $P(r_i|x_i)$  and  $P(r_i|x_0)$ , computed using a Gaussian kernel density estimate. As predicted, the empirical probability density functions have a much softer distribution compared to the step function assumed in deterministic approaches. These curves can then be used to choose an appropriate threshold that gives more weight to robustness or responsiveness depending on the requirements of the system.

## 6 CONSIDERING ALL TARGETS SIMULTANEOUSLY

In the previous section, we discussed how prior approaches only take into account the similarity between the target motion and the projection of the user movement onto the input space when computing the likelihood that the user is trying to select that target. However, even if the user is following the target perfectly, without any distortion (i.e.  $U = T_i$ ) this motion can nevertheless present a degree of positive correlation with the other targets, especially if the trajectories are not carefully designed. Ideally, if the true state is  $x_i$ ,  $r(U, T_i)$  should be much higher than  $r(U, T_j)$  for all  $j \neq i$ . However, depending on the choice of similarity measure, trajectory shapes, and window sizes, this is not always achievable. Therefore,

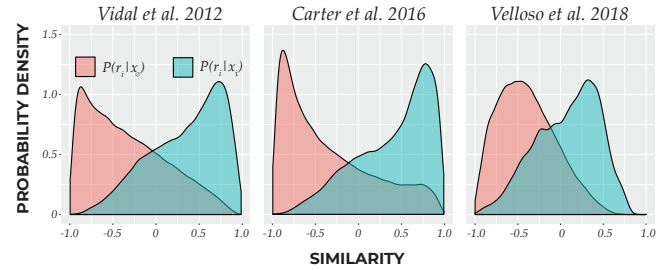


Figure 4: Probability density functions for three similarity measures found in the literature. Note that all measures yield distributions that overlap and are therefore substantially different to the step function assumed by deterministic methods.

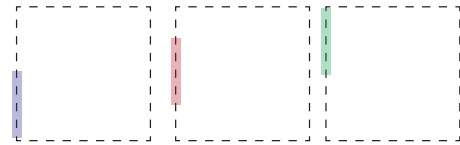
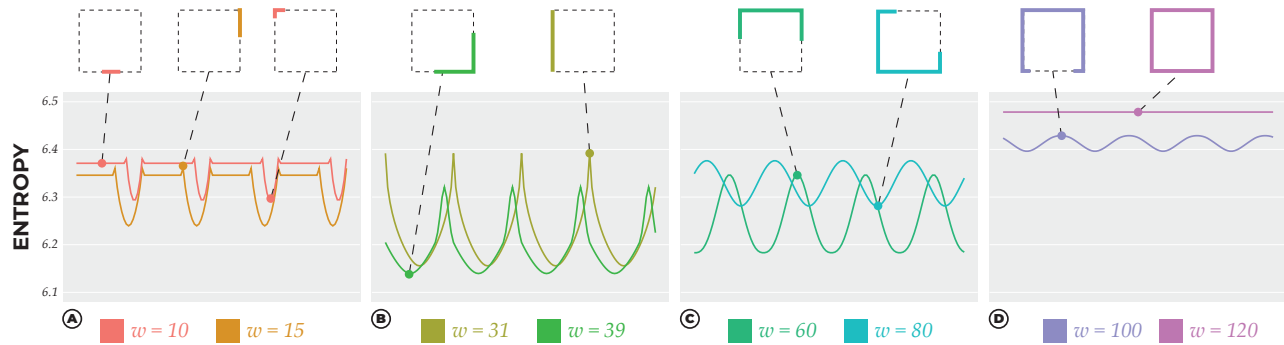


Figure 5: When the path has straight edges that are longer than the distance traversed by the target within the window, multiple trajectory windows will look the same, making it impossible to distinguish between them. In the example, we see three targets traversing the edge of a square path one after another. Even though they are temporally offset, their relative trajectories will look the same until they reach the corner.

understanding the expected levels of similarity between  $U$  and all  $T_j$  when the user is following  $T_i$  can help us make decisions about the quality of the trajectory design.

This problem is exacerbated in the case of polygons where distance traversed by the targets within a window is smaller than the length of the edges, such as the one in Figure 5. The figure shows a square where targets traverse half the length of the edge within a window and all three targets move in the same direction. The trajectories in all three windows (as well as all other windows between them) look exactly the same in terms of their relative motion. If a calibration-free system observes a straight vertical line from the input device, it will know that the user is following *one* of them, but not *which* of them. In this case, the system would have to wait until the targets start hitting the corner of the polygon to be able to tell them apart. From an information theory perspective, the corner of the square has more information than the edge, making it more distinguishable.

In practice, the differences in the information content of the windows of a trajectory mean that the point in time where the user begins to follow a target matters for the interaction—*when ideally it should not*: a target that is just entering the edge of the square will be harder to select than one that is about to hit the corner. This might affect the interaction by delaying the selection decision. If we are able to quantify how distinct any given window in a trajectory is to the others, we can have an initial estimate of the quality of



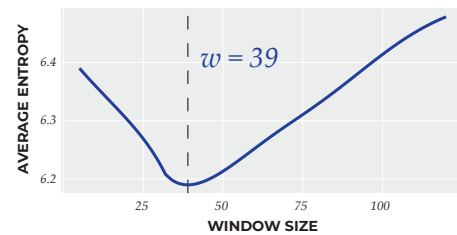
**Figure 6: The window size affects the overall entropy of the system. Different windows have different amounts of information, leading to varying levels of entropy. The x-axis represents the index of the window. Increasing the window size does not necessarily lead to a smaller entropy.**

that trajectory for a motion correlation interface. The key idea here is that motion correlation interfaces must not only consider the targets trajectories (i.e. space + time), but also the path itself (i.e. space alone).

As such, we propose the use of entropy for understanding how distinguishable each window along a path is from the rest. For each window, we compute the similarity between that window and every other window and normalise these values to obtain probabilities. We then compute the entropy for that window. This measures how similar this window is to all others—a low value of entropy means that there are few windows that capture most of the high similarity values, which is what we want. We finally compute the entropy for all windows of the trajectory to see how it fluctuates as the target moves. The idea here is that by looking at the path itself, comparing the possible windows sampled from it to each other (as opposed to comparing them to windows sampled from the input), we can estimate the upper bound of the confidence that the system can have at a given point in time (as the path itself would correspond to perfect motion matching with no noise).

To demonstrate this technique, we generated a square trajectory with 120 points. We then computed the entropy for all possible windows along the trajectory, varying the window size—this is equivalent to having 120 targets moving along a square while the user follows one of them with no distortion. We used the Pearson’s correlation of the most dissimilar axis [10, 32] as the similarity measure, and we repeated this process for windows ranging from 5 to 120 samples. In total, we computed  $116 \text{ (window sizes)} \times 120 \text{ (windows on the path)}$  values of entropy, calculated based on 120 comparisons between windows for each entropy value.

Figures 6 and 7 show how the entropy varies along the square for different window sizes. In Figure 6-A, the window sizes are shorter than the size of the edge, so plateaus of high entropy alternate with entropy drops at the windows that include the corners, with the lowest values at the windows centred at each corner. Because the x-axis refers to the first element of the window, as we increase the window sizes, the curves shift to the left. Also, as we increase the window size, the plateaus become shorter until they disappear once the window size becomes larger than the edge (Figure 6-B). The overall entropy hits a minimum when it is approximately one third of the perimeter of the square and begins to increase again. As it



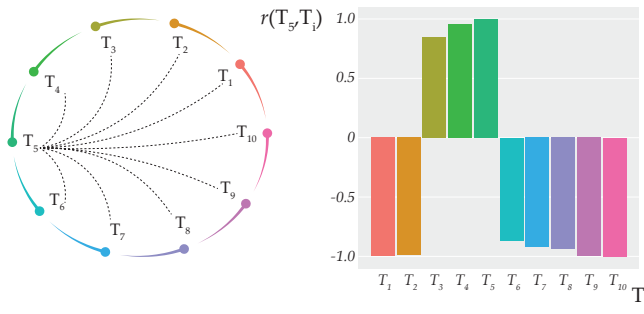
**Figure 7: Average entropy across all windows as a function of the window size, for a square with 120 samples. A window size of approximately 1/3 of the perimeter minimises the entropy.**

increases, it becomes sinusoidal (Figure 6-C) with higher baselines and smaller amplitudes until it becomes a flat horizontal line when the window size is 100% of the perimeter (Figure 6-D).

Figure 7 shows how the average entropy across the whole trajectory varies with the window size—each point in the curve is the average of each curve like the ones in Figure 6. In this case, the optimal window size would be 1/3 of the perimeter, but this will depend on the similarity measure and trajectory shape. Naturally, this window size is only optimal in terms of how well it distinguishes a window from the other windows in the shape, but in practice the optimal window will also depend on other factors, such as the interaction latency and robustness to noise. Nevertheless, this analysis can help us reduce the design space of possible window sizes to explore in empirical studies. Further, it allows us to quantify the effect of the starting point of the interaction on the expected performance of the system.

The previous analysis shows that the entropy for each window in the shape is affected by the amount of information in each segment of the shape. An analysis like this can show that a priori, the system will have a higher degree of confusion at the edges than at the corners, so the developer can choose to defer a decision until it reaches the corner or change the system parameters (e.g. adjusting the window size to always include a corner). Further, the variations in entropy are not simply due to the shape of the trajectory—the measure itself leads to variations in performance along the same





**Figure 8: Similarity between all combinations of target  $T_5$  and every possible target  $T_i$  within the same window. Even if the user managed to perfectly follow the target, they would still yield a high similarity with other targets.**

shape. This is best demonstrated by how the same pairs of input and target motion windows can yield different similarity values depending on the coordinate system (see a concrete example in the case study below).

### 6.1 Case Study: Analysing the similarity between windows in Orbits

In *Orbits*, because targets move around circular trajectories, no two windows will look exactly the same, regardless of the window size, avoiding the problem shown in Figure 5. However, depending on the point along the trajectory through which the target is passing and how far apart they are, they might nevertheless exhibit high correlation.

The easiest way to reveal this is to make pairwise comparisons between the target windows themselves (in contrast to the usual approach of comparing the targets’ and the user’s behaviours). Figure 8 shows an example of this type of analysis in the context of *Orbits*. Consider 10 targets moving around a circle with 160 samples. The figure shows a comparison between one arbitrary window ( $w = 15$ ) and all other windows using Vidal et al.’s method based on Pearson’s correlation [32]. By comparing the targets to each other, we analyse the system performance as if the user was able to perfectly match the desired target’s motion, giving us an upper bound on the system performance. In the case of the figure, the reference target is  $T_5$ . As the bar chart shows, the similarity between the target and itself is perfect ( $r(T_5, T_5) = 1.0$ )—as we should expect—but the similarity to other targets is also rather large:  $r(T_5, T_4) = 0.96$  and  $r(T_5, T_3) = 0.85$ . In practice, this means that *even if the user follows the motion of the target perfectly, we will still observe a high correlation with other targets*.

The choice of similarity measure also affects how the entropy of the design fluctuates along a path. To demonstrate this in the context of *Orbits*, we generated a circular trajectory (radius = 1) with 60 samples, from which we sampled a window with 20 samples. We then simulated user input by distorting this trajectory with a Gaussian noise (SD = .1). We then computed the similarity between the original window and its distorted version using 4 similarity metrics described in the literature [2, 7, 29, 32]. We then rotated

these windows (both the original and the distorted) by angles ranging from zero to 360 degrees, computing the similarity for each rotation.

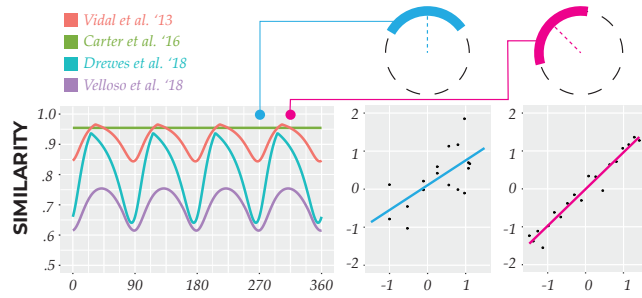
Figure 9-Left shows our results for four different similarity measures. Though the absolute value of the similarity alone is not so important in this case—what matters is how well they are able to discriminate between correct and incorrect targets—there should not be much fluctuation in this value. This is because the data being compared is *exactly the same*; the only difference is how the windows are orientated relative to the coordinate system. Notice that three of the measures are substantially affected by the orientation of the window. The plots in Figures 9-Centre and 9-Right help to illustrate why. In the case of the blue window, because there is little variance in the trajectory data in the Y axis the noise in the data ends up having a larger effect, leading to a lower correlation ( $r = 0.85$ ), as we can see in the middle plot. In the case of the pink window, both X and Y axes exhibit a more balanced variance, which in turn, leads to a more robust calculation of the correlation coefficient ( $r = 0.98$ ). The only measure that is robust to these changes in rotation is Carter et al.’s *Rotated Correlation*, which prior to calculating the correlation coefficient, rotates the window to better distribute the variance across the axes [2].

## 7 DECIDING WHEN TO MAKE A SELECTION

In the previous section, we saw that current approaches are limited in how they handle multiple targets yielding high similarities, often just selecting the most similar one. Ideally, we should be able to quantify this uncertainty and make decisions about whether and which target to select based on our level of confidence considering the interface as a whole. Here, we draw directly from Williamson and Murray-Smith, who used a similarity metric to compute a weight for each potential target, normalised them into probabilities and computed the entropy of the system to decide whether to make a selection [22, 35]. For us, entropy is interesting because it quantifies how concentrated the probabilities are around any of the targets.

The entropy value can give us a hint as to the number of states between which system is undecided: if the entropy  $H(X) = \log_2 K$ , we are uncertain between approximately  $K$  states. As we gather more evidence that a certain target  $x_i$  is being followed, we increase its probability  $p(x_i)$  and reduce the probabilities of the other states  $\{x_j | j \neq i\}$ . As  $p(x_i)$  reaches one (and the others approach zero), the uncertainty is reduced and the entropy tends to zero. Therefore, this allows us to handle the case where multiple targets are above the threshold. As a few of them yield high likelihoods, the others will yield low ones. Looking at the probabilities individually would lead us to select one or all targets; looking at the entropy, we learn that the system is not confident enough in its estimation, so we might wait until more data is available before making a selection. In practice, we make a selection when the entropy of the system falls below a certain empirically determined threshold. Other simpler, but related approaches, such as the Gini coefficient or the ratio between the largest and second largest likelihoods can also be informative in this regard.

Figure 10 shows an example of how this works in practice in the case of three targets moving with constant speed around a

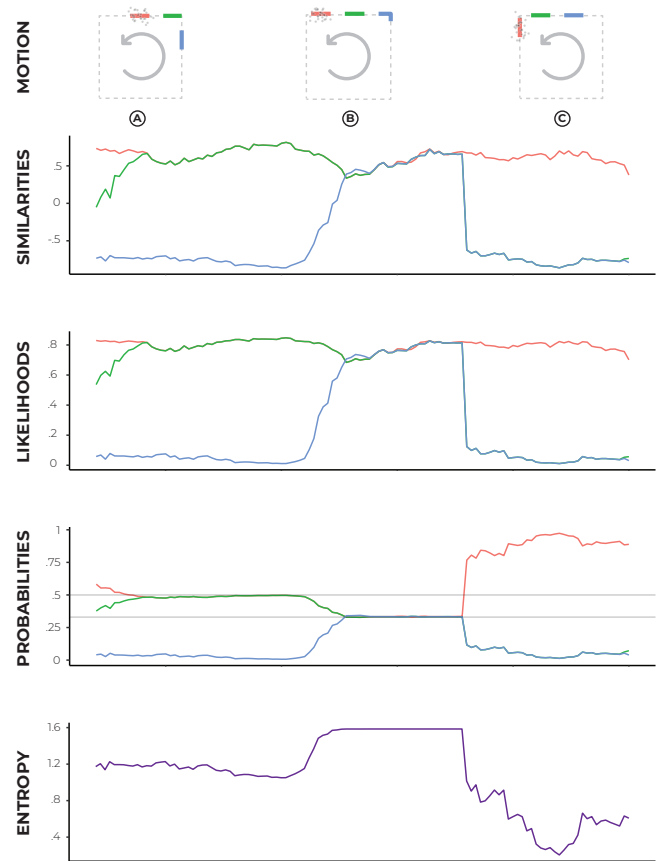


**Figure 9: Left: The effect of the coordinate system on the output of different measures. Centre and Right: Correlation between the Y coordinate of the reference window and the noisy window for windows centred at North and North-west points of the circle. When the variance in the data is more evenly distributed along each axis (pink), we observe a higher correlation than when it is concentrated in one of the axes (blue).**

square path, but slightly offset. The user data is shown in grey and corresponds to a noisy version of the red trajectory. We computed the similarity between the user data and each target trajectory using *Rotated Correlation*. We then computed the likelihoods using the corresponding probability density function shown in Figure 4. Though it is similar in shape to the similarity curve, it normalises the values and pushes middling values towards the extremes of the scale. The likelihoods are then normalised into probabilities, from which we compute the overall system entropy. Initially, both the red and the green targets yield high similarities and a deterministic system would struggle to decide which to select. When the blue target reaches the edge, the uncertainty in the system further increases. This process is reflected in the entropy curve, which increases when the blue target reaches the edge, but drops when the red target reaches the side edge and its trajectory becomes substantially different to the others. Note that the likelihoods shown in the plots are instantaneous (i.e. they only consider the data in the corresponding window, ignoring previous data). The system can be further enhanced by incorporating a dynamic model that considers the previous likelihoods. This way, the probability of the blue target at point B would not be as large as the probabilities of the other targets and the overall entropy would be lower.

### 7.1 Case Study: Analysing the entropy in Orbits

To understand the theoretical limits of a circular trajectory design for motion correlation, we first consider the problem of discovering the maximum number of targets we could add to our interface under the assumption that the user is able to perfectly mimic the motion of the target. Given the sampling rate of the eye tracker (30Hz), we can generate hypothetical perfect gaze trajectories with  $30 \times 360/Speed$  samples—leaving us with trajectories of 45, 60, and 90 samples. For each of these three trajectories, we computed the similarity between all pairs of possible windows and averaged them as per the number of samples between them. This step is necessary because as we saw in Figures 8 and 9 the correlation values vary depending on the orientation of the window relative to



**Figure 10: Stepping through a probabilistic calculation: Three targets move along a square path as the user noisily matches the motion of the red target. We compare the input signal to the trajectories of the targets, calculate the likelihoods using probability density functions, and normalise them to calculate the probabilities and the system entropy. Initially, two of the targets exhibit the same relative trajectory (A), yielding high similarities for both. When the third target reaches the edge (B), the entropy of the system increases. When the red target reaches the side edge of the square (C), its trajectory is sufficiently distinct from the others, leading to a drop in the entropy, which signals to the system that a selection can be made.**

the coordinate system. Considering that each of these samples could be a potential target leaves us with the problem of distinguishing which of these  $NSamples$  targets the user is trying to follow. As such, the initial entropy of the system is  $\log_2(NSamples)$ .

Using the same approach used by Esteves et al., we thresholded the similarity values at  $\lambda = 0.8$ . We assigned a likelihood of 1 to all windows above this threshold and zero otherwise. We then normalised the likelihoods to obtain the probabilities and the entropy. Table 1 shows our results. The table only considers targets moving in the same direction. Because in *Orbits* the authors also included targets moving in the opposite direction, we can double the number of targets. We are able to make this assumption because if the

target is moving in the opposite direction, at least one of the coordinate axes will be inversely correlated, ensuring that the overall correlation is negative. From the table, we see that there are on average approximately 10-16% of windows with a similarity above the threshold depending on the overall trajectory size. In other words, there will be confusion between a target and any target within the 36-22.5 degree arc. This means that we need at least this distance between the targets, in order to minimise uncertainty. Therefore, we can calculate the maximum number of targets supported by a given design by dividing 360 by this distance, and doubling it if we also consider targets moving in the opposite direction along the same path.

| N  | $R > \lambda$ | Proportion | Entropy | Max Targets        |
|----|---------------|------------|---------|--------------------|
| 45 | 7             | 0.156      | 2.81    | $2 \times 7 = 14$  |
| 60 | 9             | 0.148      | 3.17    | $2 \times 7 = 14$  |
| 90 | 9             | 0.099      | 3.17    | $2 \times 11 = 22$ |

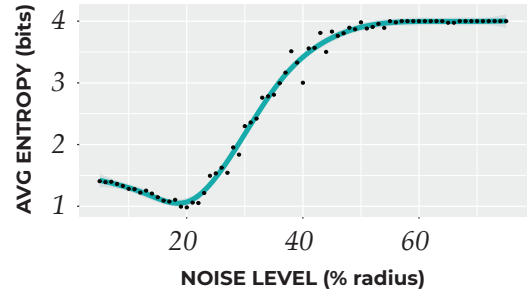
**Table 1: Simulation results for the design of *Orbits*: For each trajectory size, we show the number of windows above the threshold  $\lambda$ , the corresponding proportion of the number of windows, the entropy of the system, and the maximum number of targets that could be placed along this trajectory in order to obtain zero entropy (we multiply this last number by 2 to consider targets moving in the opposite direction).**

These results suggest that assuming that the user is able to follow the targets perfectly and that half of the targets will move in the opposite direction to the rest, we can add up to 14 targets with the *Slow* and *Medium* speeds, and up to 22 with the *Fast* speed. Esteves et al.’s empirical results found acceptable error rates for up to 8 targets, but not for 16 targets, which aligns with our theoretical predictions.

However, despite our predictions that the *Fast* condition could support up to 22 targets, their study found that even 16 were too many in this condition. This is because at high speeds the eyes can no longer maintain a smooth pursuit, engaging instead in a series of saccades. This highlights the importance of user testing, as our theoretical results can only provide an upper limit for the algorithm performance, but do not say much about human capabilities without further data. Nevertheless, the advantage of a probabilistic approach is that it enables us to incorporate motor models as priors, which is a rich direction for future work.

## 7.2 Effect of Noise on Entropy

To better understand how a design will handle noisy data, we can repeat the tests above under different noise conditions. As an example, consider Figure 11. To generate this plot, we simulated 16 targets moving around a circle at 180 degrees/s (the *Medium* speed tested in *Orbits*). We then simulated user input by adding Gaussian noise to the trajectory data, varying the noise between 5-75% of the radius of the circle. We repeated each simulation 30 times and computed the average entropy across all simulations for each level of noise. The smooth curve in the figure was computed with a cubic regression spline on the data.



**Figure 11: Effect of noise on the entropy of the system, considering 16 targets moving with the Medium speed tested in *Orbits*, with a window size of 30 samples.**

The higher the noise level, the lower the correlation with the trajectory data. At small noise levels, this reduces the similarity with incorrect targets, leaving fewer targets above the threshold, and therefore reducing the uncertainty. At approximately 20%, however, the uncertainty begins to increase, culminating at an entropy of 4, which is the maximum for 16 targets— $\log_2(16) = 4$ . This is not a universal rule, and these results should be computed for each specific combination of variables, but one can use a curve like this to predict the uncertainty in the system given a sensor’s noise characteristics. Conversely, given the sensor’s noise characteristics, we can specify the minimum radius of the targets to achieve a desired entropy level.

## 8 DISCUSSION

One main contribution of this paper is the re-framing of previous works on motion correlation-based interaction techniques under a probabilistic framework. Along this process, we have shown that individual design decisions in a motion correlation interface, such as which window size or similarity measure to use, should not be made in isolation as they influence each other. Computational techniques drawn from probability and information theory are particularly useful in helping us to quantify these decisions. A probabilistic approach offers a common language that allow us to consolidate the existing knowledge within motion correlation and also to relate it to other types of interaction techniques. In particular, it provides us with a notation to formalise the principles behind motion correlation interfaces.

Another contribution is the description of practical examples that demonstrates that a probabilistic approach can help us analyse motion correlation designs and understand why they work (or do not). Next we summarise other relevant outcomes spread along this paper as a recommendation list for future work.

### 8.1 Lessons Learned

**Probabilistic modelling extends previous approaches, while still being compatible with them.** Deterministic approaches are simply particular cases where the probabilities are 1 and 0. By explicitly modelling uncertainty in a system we can yield valuable design insights, such as quantifying how distinguishable each segment

of the path is, understanding the limitations of different similarity measures, and enabling the integration of different sources of information.

**Interpret the similarity measure as an estimate of mutual information.** Because selection by motion correlation can be considered as transmission of information about the target trajectory, we argued that measuring motion similarities is related to estimating mutual information. As a consequence, a good similarity measure must not only yield high values when user and target motions are similar, but also yield low values in every other case.

**Make explicit the similarity measure assumptions:** Different similarity measures perform differently depending on the assumptions behind the design. By making these assumptions explicit (for example, invariance to scale, rotation, and/or translation), we can make more informed comparisons between measures. As such, when proposing novel similarity measures, future works should clearly label which transformations are necessary to unambiguously transform the trajectory data into the gaze data.

**Use softer functions instead of step functions.** Hard thresholds make comparisons unstable when the results are near the threshold and are more vulnerable to confusion when multiple targets yield similarities above the threshold. Using softer thresholds mitigates this problem. Even better, we can incorporate our knowledge of the problem domain by using the probability density functions obtained by a mix of simulations and data collections (see Figure 4).

**Consider the similarity values together instead of independently.** Similarity values only consider each possible state individually, but high correlations with incorrect targets can still be consistent with perfectly following the correct target (see Figure 8). Calculating the entropy of the probabilities of different states enables us to quantify the ambiguity of the inference and make more informed decisions about how to proceed.

**Understand how the entropy of the design behaves.** The shape of the trajectory, the number of samples, the window size, and the similarity measures all influence each other. Computing the inherent entropy of each window in relation to other windows helps us quantify the uncertainty of the design prior to even collecting user data and identify points of risk (see Figure 6).

**Simulation is not an excuse for not collecting user data.** Though simulations can be very valuable in explicating design decisions, they do not replace user testing. However, as the community makes progress in modelling user behaviour from a computational perspective, we can expect to see more accurate models that will further empower our own simulations.

## 9 CONCLUSION

This paper frames interaction techniques based on motion correlation as a probabilistic reasoning problem. We argued that previous techniques can be formulated as such, and demonstrated the kinds of analysis that this formalism enables. In particular, we demonstrated analyses of similarity measures in terms of their assumptions about user and interface data, of their behaviour along different trajectory shapes, and of their value as a measure of mutual information. We also made a case for computing conditional probabilities for each potential state and for making decisions based on entropy

values rather than on independent similarities. Finally, we argued for the use of entropy as a tool for understanding the information content of a given design.

We highlight a few practical pieces of advice for developers that stem from our analysis: (1) use the advice in section 4 to select an appropriate similarity metric, (2) collect null data to derive probability density functions for the priors (e.g. Figure 4), (3) select an appropriate window size and path shape based on the analysis of the entropy of the shape, (4) use entropy or other measures of uncertainty to decide whether to make a selection. All of these are ideas that can be readily incorporated in the development of systems like the ones described in the literature. We encourage future work to frame their results in probabilistic terms in order to better contextualise their contributions and more consistently measure progress. To facilitate this, we suggest a series of lessons learned that can be taken forward in future work.

## ACKNOWLEDGMENTS

This work was partially funded by a FAPESP-University of Melbourne SPRINT Grant (Project Number: 2016/10148-3). Eduardo Velloso is the recipient of an Australian Research Council Discovery Early Career Award (Project Number: DE180100315) funded by the Australian Government, and Carlos Morimoto is the recipient of FAPESP grants no. 2016/10148-3 and 2017/50121-0.

## REFERENCES

- [1] Pradipta Biswas, Gokcen Aslan Aydemir, Pat Langdon, and Simon Godsill. 2013. Intent recognition using neural networks and Kalman filters. In *International Workshop on Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer, Maribor, Slovenia, 112–123. [https://doi.org/10.1007/978-3-642-39146-0\\_11](https://doi.org/10.1007/978-3-642-39146-0_11)
- [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* (Santa Clara, California, USA) (*CHI '16*). ACM, New York, NY, USA, 3415–3427. <https://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* (Heidelberg, Germany) (*UbiComp '16*). ACM, New York, NY, USA, 298–303. <https://doi.org/10.1145/2971648.2971714>
- [4] 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* (Qu&#233;bec City, QC, Canada) (*UIST '17*). ACM, New York, NY, USA, 179–192. <https://doi.org/10.1145/3126594.3126626>
- [5] Timothy F. Cootes and Christopher J. Taylor. 2004. Statistical models of appearance for computer vision.
- [6] 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* (Launceston, Tasmania, Australia) (*OzCHI '16*). ACM, New York, NY, USA, 371–380. <https://doi.org/10.1145/3010915.3010917>
- [7] Heiko Drewes, Mohamed Khamis, and Florian Alt. 2018. Smooth Pursuit Target Speeds and Trajectories. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia* (Cairo, Egypt) (*MUM 2018*). ACM, New York, NY, USA, 139–146. <https://doi.org/10.1145/3282894.3282913>
- [8] Heiko Drewes, Mohamed Khamis, and Florian Alt. 2019. DialPlates: Enabling Pursuits-Based User Interfaces with Large Target Numbers. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia* (Pisa, Italy) (*MUM '19*). Association for Computing Machinery, New York, NY, USA, Article 10, 10 pages. <https://doi.org/10.1145/3365610.3365626>
- [9] Ian L. Dryden and Kanti V. Mardia. 1998. *Statistical Shape Analysis*. John Wiley & Sons, New York, NY.
- [10] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. 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 (Osaka, Japan) (*UbiComp/ISWC '15 Adjunct*). ACM, New York, NY, USA, 419–422. <https://doi.org/10.1145/2800835.2800942>
- [11] 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* (Daegu, Kyungpook, Republic of Korea) (*UIST '15*). ACM, New York, NY, USA, 457–466. <https://doi.org/10.1145/2807442.2807499>
- [12] 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* (Qu&#233;bec City, QC, Canada) (*UIST '17*). ACM, New York, NY, USA, 167–178. <https://doi.org/10.1145/3126594.3126616>
- [13] 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* (Boston, MA, USA) (*CHI '09*). ACM, New York, NY, USA, 289–298. <https://doi.org/10.1145/1518701.1518748>
- [14] Argenis Ramirez Gomez and Hans Gellersen. 2018. Smooth-i: Smart Recalibration Using Smooth Pursuit Eye Movements. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) (*ETRA '18*). ACM, New York, NY, USA, Article 10, 5 pages. <https://doi.org/10.1145/3204493.3204585>
- [15] Laura D. Goodwin and Nancy L. Leech. 2006. Understanding Correlation: Factors That Affect the Size of  $r$ . *The Journal of Experimental Education* 74, 3 (2006), 251–266. <https://doi.org/10.3200/JEXE.74.3.249-266>
- [16] Kasper Hornbæk and Antti Oulasvirta. 2017. What Is Interaction?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI '17*). ACM, New York, NY, USA, 5040–5052. <https://doi.org/10.1145/3025453.3025765>
- [17] Mohamed Khamis, Carl Oechsner, Florian Alt, and Andreas Bulling. 2018. VR-pursuits: Interaction in Virtual Reality Using Smooth Pursuit Eye Movements. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (Castiglione della Pescaia, Grosseto, Italy) (*AVI '18*). ACM, New York, NY, USA, Article 18, 8 pages. <https://doi.org/10.1145/3206505.3206522>
- [18] 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* (Heidelberg, Germany) (*UbiComp '16*). ACM, New York, NY, USA, 274–285. <https://doi.org/10.1145/2971648.2971679>
- [19] Eric Langford, Neil Schwertman, and Margaret Owens. 2001. Is the Property of Being Positively Correlated Transitive? *The American Statistician* 55, 4 (2001), 322–325. <https://doi.org/10.1198/000313001753272286> arXiv:<https://doi.org/10.1198/000313001753272286>
- [20] Wanyu Liu, Rafael Lucas D'Oliveira, Michel Beaudouin-Lafon, and Olivier Rioul. 2017. BIGNav: Bayesian Information Gain for Guiding Multiscale Navigation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI '17*). Association for Computing Machinery, New York, NY, USA, 5869–5880. <https://doi.org/10.1145/3025453.3025524>
- [21] Wanyu Liu, Olivier Rioul, Joanna McGrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2018. BIGFile: Bayesian Information Gain for Fast File Retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3173574.3173959>
- [22] David J.C. MacKay. 2003. *Information theory, inference and learning algorithms*. Cambridge university press, United Kingdom. <https://doi.org/10.2277/0521642981>
- [23] Antti Oulasvirta, Xiaojun Bi, and Andrew Howes. 2018. *Computational interaction*. Oxford University Press, United Kingdom. <https://doi.org/10.1093/oso/9780198799603.001.0001>
- [24] Ken Pfeuffer, Mélodie 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* (St. Andrews, Scotland, United Kingdom) (*UIST '13*). ACM, New York, NY, USA, 261–270. <https://doi.org/10.1145/2501988.2501998>
- [25] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2010. FingerCloud: Uncertainty and Autonomy Handover Incapacitive Sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) (*CHI '10*). Association for Computing Machinery, New York, NY, USA, 577–580. <https://doi.org/10.1145/1753326.1753412>
- [26] Julia Schwarz, Scott Hudson, Jennifer Mankoff, and Andrew D. Wilson. 2010. A Framework for Robust and Flexible Handling of Inputs with Uncertainty. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology* (UIST '10). Association for Computing Machinery, New York, NY, 47–56. <https://doi.org/10.1145/1866029.1866039>
- [27] Daming Shi, Steve R. Gunn, and Robert I. Dampier. 2003. Handwritten Chinese radical recognition using nonlinear active shape models. *IEEE transactions on pattern analysis and machine intelligence* 25, 2 (2003), 277–280. <https://doi.org/10.1109/TPAMI.2003.1177158>
- [28] 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>
- [29] 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 the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) (*ETRA '18*). ACM, New York, NY, USA, Article 25, 9 pages. <https://doi.org/10.1145/3204493.3204524>
- [30] 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* (Brisbane, QLD, Australia) (*DIS '16*). ACM, New York, NY, USA, 812–817. <https://doi.org/10.1145/2901790.2901867>
- [31] David Verweij, Augusto Esteves, Vassilis-Javed Khan, and Saskia Bakker. 2017. WaveTrace: Motion Matching Input Using Wrist-Worn Motion Sensors. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI EA '17*). ACM, New York, NY, USA, 2180–2186. <https://doi.org/10.1145/3027063.3053161>
- [32] 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* (Zurich, Switzerland) (*UbiComp '13*). ACM, New York, NY, USA, 439–448. <https://doi.org/10.1145/2493432.2493477>
- [33] Janett Walters-Williams and Yan Li. 2009. Estimation of Mutual Information: A Survey. In *Rough Sets and Knowledge Technology*, Peng Wen, Yuefeng Li, Lech Polkowski, Yiyu Yao, Shusaku Tsumoto, and Guoyin Wang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 389–396. [https://doi.org/10.1007/978-3-642-02962-2\\_49](https://doi.org/10.1007/978-3-642-02962-2_49)
- [34] John Williamson. 2006. *Continuous uncertain interaction*. Ph.D. Dissertation. University of Glasgow.
- [35] John Williamson and Roderick Murray-Smith. 2004. Pointing Without a Pointer. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria) (*CHI EA '04*). ACM, New York, NY, USA, 1407–1410. <https://doi.org/10.1145/985921.986076>
- [36] Cheng Zhang, Xiaoxuan Wang, Anandghan Waghmare, Sumeet Jain, Thomas Ploetz, Omer T. Inan, Thad E. Starner, and Gregory D. Abowd. 2017. FinGOrbits: Interaction with Wearables Using Synchronized Thumb Movements. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers* (Maui, Hawaii) (*ISWC '17*). ACM, New York, NY, USA, 62–65. <https://doi.org/10.1145/3123021.3123041>