Facilitating Temporal Synchronous Target Selection through User Behavior Modeling

TENGXIANG ZHANG, Tsinghua University and Chinese Academy of Sciences
XIN YI*, Tsinghua University
RUOLIN WANG, Tsinghua University and UCLA
JIAYUAN GAO, YUNTAO WANG, and CHUN YU, Tsinghua University
SIMIN LI, Beihang University
YUANCHUN SHI, Tsinghua University

Temporal synchronous target selection is an association-free selection technique: users select a target by generating signals (e.g., finger taps and hand claps) in sync with its unique temporal pattern. However, classical pattern set design and input recognition algorithm of such techniques did not leverage users' behavioral information, which limits their robustness to imprecise inputs. In this paper, we improve these two key components by modeling users' interaction behavior. In the first user study, we asked users to tap a finger in sync with blinking patterns with various period and delay, and modeled their finger tapping ability using Gaussian distribution. Based on the results, we generated pattern sets for up to 22 targets that minimized the possibility of confusion due to imprecise inputs. In the second user study, we validated that the optimized pattern sets could reduce error rate from 23% to 7% for the classical Correlation recognizer. We also tested a novel Bayesian, which achieved higher selection accuracy than the Correlation recognizer when the input sequence is short. The informal evaluation results show that the selection technique can be effectively scaled to different modalities and sensing techniques.

CCS Concepts: • **Human-centered computing → Interface design prototyping**; *User models*; *Gestural input*.

Additional Key Words and Phrases: target selection, synchronous tapping, Bayesian prediction, pattern generation

ACM Reference Format:
Tengxiang Zhang, Xin Yi, Ruolin Wang, Jiayuan Gao, Yuntao Wang, Chun Yu, Simin Li, and Yuanchun Shi. 2019. Facilitating Temporal Synchronous Target Selection through User Behavior Modeling. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 159 (December 2019), 24 pages. https://doi.org/10.1145/3369839

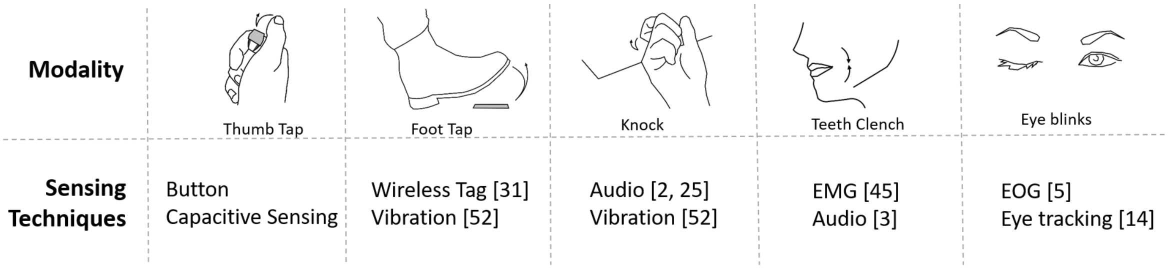
*This is the corresponding author.

Authors' addresses: Tengxiang Zhang, Tsinghua University, Institute of Computing Technology, Chinese Academy of Sciences, Kexueyuan South Road, Beijing, 100084, ztxseuthu@gmail.com; Xin Yi, Key Laboratory of Pervasive Computing, Ministry of Education, Beijing Key Lab of Networked Multimedia, Beijing National Research Center for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, hhdxs2@163.com; Ruolin Wang, Tsinghua University, UCLA; Jiayuan Gao; Yuntao Wang; Chun Yu, Global Innovation eXchange Institute, Key Laboratory of Pervasive Computing, Ministry of Education, Beijing Key Lab of Networked Multimedia, Department of Computer Science and Technology, Tsinghua University, Shuangqing Road, Beijing, 100081; Simin Li, Department of Electrical and Computing Engineering, Beihang University, Beijing, 100081; Yuanchun Shi, Tsinghua University, Shuangqing Road, Beijing, 100081.

2474-9567/2019/12-ART159 $15.00
https://doi.org/10.1145/3369839

1 INTRODUCTION

With the increasing popularity of ubiquitous computing and Internet of Things (IoT), more and more devices and daily objects are becoming interactive. However, as one of the most basic interaction tasks, target selection can be challenging on these new interfaces. There are three reasons: 1) cross-device interaction for a large number of devices calls for association-free target selection techniques [6, 10, 23]. It is not practical to have a designated controller for each individual device or require users to associate with the devices each time before usage, especially when there is a large number of devices; 2) the interaction expressivity (e.g., audio, gesture) and form factor (e.g. button, touchscreen) of the interfaces can be vastly different from each other. Users are not able to transfer their interaction experience across interfaces. Therefore, providing a consistent and spontaneous interaction experience across different interfaces is of significance; 3) The sensing capability of the new interfaces can be very limited (e.g. BitID [49] only senses binary inputs), so traditional target selection techniques are not applicable on such interfaces.



Modality	Thumb Tap	Foot Tap	Knock	Teeth Clench	Eye blinks
Sensing Techniques	Button Capacitive Sensing	Wireless Tag [31] Vibration [52]	Audio [2, 25] Vibration [52]	EMG [45] Audio [3]	EOG [5] Eye tracking [14]

Fig. 1. Users can tap thumb or foot, knock on a table, clench/click teeth, or blink eye to select objects.

Aimed at these challenges, temporal synchronous target selection [6, 27, 34, 50] has been proposed by researchers to enable association-free target selection on devices with different interaction interfaces. Instead of browsing and selecting the target device from a list on a screen, users can generate temporal synchronized signals with a temporal pattern (e.g., blinking) to select the corresponding target. This kind of technique has three advantages: 1) It does not require device association as long as the pattern for each target is unique, which can save the total interaction time; 2) Temporal signal can be generated on multi-modality interfaces, which enables subtle and accessible selection experience, so that users can choose the appropriate interface when in different scenarios. For example, users can tap fingers (touchscreen [6]), clap hands (audio [22]), tap foot (vibration[52]), contract muscles (EMG [3, 35, 45]), blink eye (EOG [5]), and even breath [16, 17] to sync with the target pattern. As the interaction paradigm-generating binary changing signals in sync with the target pattern-remains the same, users would be able to transfer the interaction experience across different interfaces; 3) The selection technique's extremely low requirement of sensing resources makes it compatible with a wide variety of both new and existing sensors. For example, such selection technique can be used on several recently proposed low-cost and easily-deployable touch [19, 49, 51] and audio [2, 25] input interfaces to provide a pervasive selection experience. It is especially suitable for resource-constrained devices with limited sensing capabilities due to cost, power, or size constrains (e.g. smart rings).

However, we see two major limitations in current temporal synchronous selection approaches: 1) there lacks a design guideline for the patterns of different targets. Consequently, the patterns displayed are not necessarily optimized for the selection task, which may lead to sub-optimal performance due to mutual confusion among "similar" patterns. It is not even trivial to define the similarity metric between patterns, which is decided by not only the patterns *per se*, but also the users' interaction ability; 2) As we will show in this paper, the imprecise inputs during the beginning of synchronizations can deteriorate the performance of the widely used Correlation recognizer, especially when there is fewer (e.g. $n < 7$) selective targets (i.e. shorter input sequences). In this paper, we address the two challenges through systematically modeling users' synchronization interaction behavior: 1) To address the first challenge, we proposed a model-based pattern set optimization procedure to reduce recognition confusions, thus support more selective targets; 2) To address the second challenge, we proposed a model-based Bayesian recognizer that is more robust to imprecise inputs compared with the Correlation recognizer.

In this paper, we validate our model-based temporal selection technique on widely-used finger tap input interfaces with a distal movie poster selection task. Specifically, we first extracted *Period* and *Lag* as the two major features for defining the design space of blinking patterns. We then modeled users' finger tapping ability when syncing with different patterns Study 1, and found that both input period and delay followed a Gaussian distribution. Based on the results, we optimized the pattern set generation process by minimizing the confusion probability among different patterns caused by imprecise inputs. Study 2 has two goals: 1) Validate the improvement of the optimized pattern set; 2) Evaluate the differences between the Bayesian and the Correlation recognizer. We found that users can select from up to 21 targets with an accuracy higher than 90%. Compared with using a period-increasing pattern set, the optimized pattern set reduces selection error rate from 23% to 7% when using the Correlation recognizer, and from 14% to 6% when using the Bayesian recognizer. Our proposed Bayesian recognizer also performs better than the classical correlation-based recognizer when the input sequence is short. In the end, we discussed the selection technique's scalability across modalities (thumb tap, foot tap, and teeth clench) and sensing techniques (pressure, capacitive, and wireless signal strength based sensing) by informal evaluations. The selection accuracies when using the Bayesian recognizer and pressure sensor are above 90% for all three modalities even with non-modality specific behavior models, demonstrating potential to leverage one behavior model across modalities.

The contribution of this paper is four-folded:

(1) Theoretically, we formally explored the design space of blinking patterns in terms of *Period* and *Lag*, which complements the design rationale of temporal patterns, and introduced the Bayesian probability to quantify the similarity among patterns.
(2) Empirically, we provide models of users' synchronous tapping behavior. Based on the results, we optimized the pattern set generation process by presenting an optimized pattern set for up to 22 targets that minimized the confusion caused by imprecise inputs. We also validated that the optimized pattern set can improve the selection accuracy for both Correlation and Bayesian recognizers.
(3) Technically, we proposed a Bayesian recognizer which achieves higher performance than the classical Correlation recognizer when the input sequence is short.
(4) In application, we explored the model-based selection technique's scalability to different modalities and sensing techniques through informal evaluations. The results implied that users can transfer their temporal synchronous selection experience across modalities and our technique can work with different sensors.

2 BACKGROUND AND RELATED WORK

In this section, we first look at current widely used *target selection techniques* in a smart space; then we introduce both spatial and temporal *synchronous* target selection techniques; at last, we examine the *patterns* used by different temporal synchronous selection techniques. We also map different techniques onto a target selection design space for a clear comparison.

2.1 Common Target Selection Techniques

Currently, users can select a target on a touch screen from a list or a collection of APPs. The cost and size of current touch screens are high though, which makes them inappropriate for ubiquitous deployment. The screen wakeup process (e.g. unlocking smart phones) makes the selection method unspontaneous. Users need to learn and remember different selection operations on different platforms. Switching APPs is also shown to introduce significant overhead in both cognition and efficiency [26]. Users can also select a target with speech for a natural and spontaneous experience. Speech based selection is not subtle though, which limits its application scenarios (e.g. not suitable in libraries). Microphones and advanced speech recognition algorithm are required for such method.

One of the most widely used selection mechanism is pointing one device toward the target device. For example, a TV remote controller emits directive Infrared (IR) beam that can be only be received by the target device pointed. Researchers also used laser and IR transceivers to improve selection experience on handheld [30] and head mounted [47] devices. For devices equipped with speakers and microphones, Doppler Effect of audio signals can be leveraged to recognize pointing gestures [4, 37]. Users can also select a target by pointing the camera toward the target and take a picture [9, 13]. However, such pointing based selection mechanisms require extra (e.g. IR and laser transceivers) or advanced (e.g. microphones, speakers, and cameras) hardware resources that can be large, expensive, and power demanding. The selection process is also not subtle. Aside from pointing with hand or fingers, users can also point their gazes toward the target and then dwell for selection [28]. Gaze pointing is subtle and efficient [36]. The estimation of gaze direction can be difficult though [46], which limits the number of targets supported.

2.2 Synchronous Target Selection Techniques

Recently, spatial synchronous target selection [38] has gained traction for its spontaneous and natural selection experience: users sync with the movement trace of the target on a distal screen to select it. Pursuit [40] and Orbits [14] show that users can select a moving target by syncing eye gazes with it on distal displays or smart watches. Users can also synchronize their head, hand, and finger movements for target selection. The movements of different body parts can be recorded by eye tracking apparatus [14], cameras [8, 11, 12], or IMUs [15, 31, 39, 48]. Spatial synchronous selection techniques differentiate targets by their movement traces instead of positions, which alleviates the 'Midas Touch' issue of pointing. Thus it is especially suitable for target selection on one distal screen. The display of moving paths usually requires a screen though, which may be too large or expensive for certain devices.

Temporal synchronous selection technique is a sensorimotor synchronization task [33], which selects a target by syncing with its unique temporal patterns. RhythmLink [27] selects the target device by tapping rhythms of a designated song. SynchroWatch[34] selects a target on a smartwatch screen by syncing their thumbs with a blinking overlay on the target. Tap-to-Pair[50] enables users to tap on a device to synchronize its wireless signal strength to the blinking pattern of the target device for association purpose. Temporal synchronous selection techniques only require a binary sensor on the initiating device, and a binary display (e.g. LED) associated with the target. The extremely low resource requirements makes the selection technique applicable on interaction interfaces with different sizes, costs, and power constrains. It even works with recently proposed low-cost passive backscatter binary sensors [19, 49] for target selection. Users can also generate binary signals in a wide variety of modalities (e.g. teeth clench, breath) for a pervasive selection experience. In Table 1 we roughly divide the above discussed techniques into six categories and compare them in five aspects:user effort to learn the technique, user effort to select a target, sensing and display resources (e.g. hardware, computing), subtleness and privacy level, and the number of targets it can support.

Table 1. Comparison of Example Target Selection Techniques

Selection Technique	Learning Effort	Selection Effort	Resource Requirements	Subtleness &Privacy	Targets Supported
TouchScreen	Medium	Low	High	High	High
Speech	Low	Low	High	Low	High
Hand Pointing [13, 30, 37]	Low	Low	High	Medium	High
Gaze Pointing [28, 46]	Low	Low	High	High	Low
Spatial Sync [12, 14, 31, 40]	Low	Medium	Medium	Medium	Medium
Temporal Sync [34, 44, 50]	Low	Medium	Low	High	Medium

2.3 Temporal Patterns Design

The number of targets that current temporal synchronous techniques support is limited though. A principled pattern design guideline could reduce recognition confusion between targets, thus support selection from more targets. One way to design temporal patterns is to refer to existing music pieces. TapSongs[43] and RhythmLink[27] uses rhythms from songs as the temporal sequence. The method does not require any pattern indications, since it fully relies on users' memory of song rhythms. However, there are a limited number of songs that users are familiar with. Also, different users tend to know different songs. So, design patterns based on existing song rhythms does not scale well for a large number of users and targets.

Instead of using existing song rhythms, Resonant Bits [7] relies on the micro-movement of hands to sync with an oscillating pendulum on the screen for interactions. Such a pattern takes a large space on the screen though. So the number of targets it can support is limited. Ghomi et.al.[20] defined their own "rhythmic pattern" as a sequence of beats composed of three types of taps and two types of breaks. The patterns used in SEQUENCE[6] are coded as 8 bits binary sequences. Tap-to-Pair[50] uses cyclic coded binary sequences. The coded patterns can support selection from more targets. However, such patterns usually take more time for users to learn and select, which might deteriorate user experience. Also, coded patterns can have irregular breaks, which is difficult for users to sync with[21]. Such pattern design methods are highly dependent on the coding schemes used and lack a quantitative metric to guide the pattern set design process.

SynchroWatch[34] and Seesaw[44] select from two targets by using two color blocks that blink at the same period but opposite phases. A wrist gesture is designed to iterate through different target pairs to select from more than two targets. However, it can be physically demanding to switch pairs when there are many targets. The gesture also makes it difficult to apply such pattern design and association method to other interfaces.

In this paper, we propose to use non-coded periodic patterns with different periods and initial lags. Compared with patterns differentiated only in the phase dimension, the Period-Lag 2D design space supports selection from more targets. The Bayesian probability of correct selections are calculated based on user tapping behavior models on such design space, which can be used to design pattern sets with minimized confusion, thus support more targets. The behavior model based design method also enables the usage of the Bayesian recognizer, which can reduce selection errors caused by imprecise inputs at the beginning of syncing when users are still 'finding the rhythm'.

3 DESIGNING TEMPORAL PATTERNS

A core component for temporal synchronous target selection techniques is the design of patterns for each target. The patterns should be distinguishable from each other and easy for users to follow. To this end, the temporal characteristics of the patterns should be carefully designed and optimized. In this section, we first define the design space of temporal patterns used in this paper, then introduce the optimized pattern sets generation process.

3.1 Design Space of A Single Pattern

Ideally, a temporal pattern with any waveform can be used as a pattern for selection. In practice, however, most researchers used periodic patterns [6, 34, 50]. Compared with non-periodic patterns, this would allow the users to remember the pattern and plan their syncing behavior in advance. Rather than following an unpredictable pattern, learning the pattern can result in better user experience with less fatigue and shorter selection time. Therefore, in this paper, we focus on designing periodic patterns. Furthermore, we assume that the pattern is represented with binary data streams, so that it can be displayed on interfaces with different output capabilities. We describe the periodic pattern used in this paper as "blinking pattern" henceforth.

As a binary sequence, the pattern can be either monotone or coded. One widely known example of coded pattern is Morse Code. Tap-to-Pair [50] also used coded non-cyclic patterns, which is essentially pulse modulation with the same frequency but different duty cycles. However, coded pattern sets lack a quantitative optimization parameter. Also, they require longer observation time, which may lead to longer selection time especially when the pattern set is large.

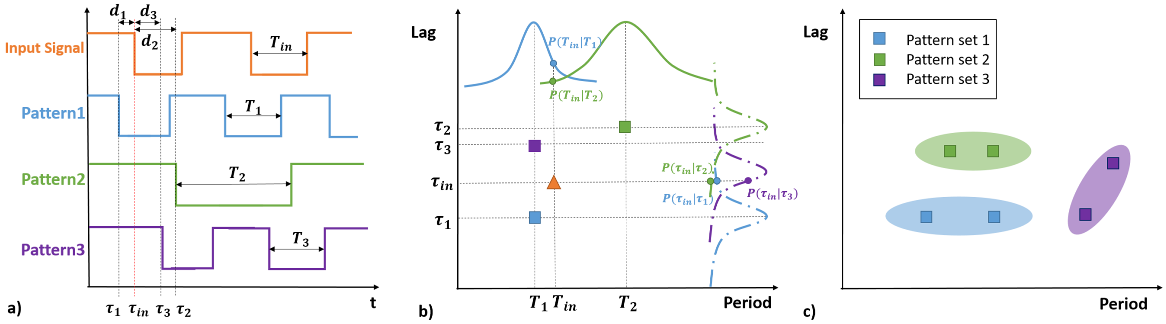


Fig. 2. T , τ, and d for the input signal and three patterns are illustrated on the time domain (a) and mapped to the T-τ plane (b). Three pattern sets are shown in (c), each with two patterns ($n = 2$).

In this paper, we used monotone blinking patterns with two features: *Period* (T) and *Lag* (τ). The 2D design space enables an easier and straightforward modeling of syncing behaviors on each dimension. A design space with higher dimension could support more targets, but they could also require extensive modeling efforts. For example, if a code dimension is added, we might need to model syncing behaviors for each code separately, which could be impractical for a complicated coding scheme. *Period* is defined as **half** of the repetition duration, which quantifies the frequency of the repetition; *Lag* quantifies the initial time lag after a preset start time. We also define *edge* as the state changing moment in a pattern or input signal. Now, the design space of blinking patterns can be described on a T-τ two dimensional plane, where each pattern can be represented by a dot at coordinate (T, τ). Figure 2a shows an example of the processed input signal and three patterns in the time domain. Pattern 1 and 3 have the same period (T_1) but different initial lags with respect to the start time ($t = 0$), while Pattern 2 has a different period (T_2). Their corresponding locations on the T-τ plane is shown in Figure 2b.

We define the asynchrony of the input signal with the closest edge of a pattern [33] as *delay* (d) (see Figure 2a). For the k^{th} pattern, $d_{in,k} = \tau_{in} - \tau_k$. It is obvious that for any period T, d lies within $[-0.5T, 0.5T)$. Negative d

indicates that the phase of the input leads ahead of the target pattern, and positive d indicates the phase of the input lags behind the target pattern.

3.2 Evaluation Metric of a Pattern Set

We define the collection of patterns corresponding to all selective targets as a *pattern set*. There are many possible patterns sets within the design spaces. Within each pattern set (patterns visible to the user at the same time), similar patterns will be difficult for both users and recognizers to differentiate, thus causing confusion (erroneous selection). For example, there are three pattern sets with two patterns in each set ($n = 2$) in Figure 2c. Pattern set 1 (Pset1) is likely to have a lower confusion rate than Pset2 since the two patterns in Pset1 are further apart than those in Pset2 on the period dimension. However, it is not as straightforward to decide whether Pset1 would have a lower confusion rate than that of Pset3, since the patterns in Pset3 differ in both T and τ dimensions. So an evaluation metric is necessary for pattern set design and optimization, which can be calculated based on selection probabilities given an input pattern.

A selection probability confusion matrix can be calculated by making each pattern as input while others as selection targets. We calculate item (i, j) in the matrix as the probability of predicting target as the i^{th} pattern given the input signal identical to the j^{th} pattern. Assuming independent distributions of period and delay, we calculate the probabilities according to the Bayesian theorem. We also use total probability formula in the calculation for T and τ respectively, assuming users always intend to select a pattern with a certain period or initial lag.

$$\begin{aligned} P(i|input = j) &= P(T_i, \tau_i | T_{input}, \tau_{input}) && (1) \\ &= P(T_i|T_{input}) \times P(\tau_i|\tau_{input}) \\ &= \frac{P(T_{input}|T_i)P(T_i)}{P(T_{input})} \times \frac{P(\tau_{input}|\tau_i)P(\tau_i)}{P(\tau_{input})} && (2) \\ &= \frac{P(T_{input}|T_i)P(T_i)}{\sum_{k=1}^{n_T} P(T_{input}|T_k)P(T_k)} \times \frac{P(\tau_{input}|\tau_i)P(\tau_i)}{\sum_{k=1}^{m_i} P(\tau_{input}|\tau_k)P(\tau_k)} && (3) \end{aligned}$$

where $T_{input} = T_j$ and $\tau_{input} = \tau_j$. n_T is the number of patterns with unique periods in a set, and m_i is the number of patterns with the same period T_i but different initial lags. In real use, $P(T_i)$ and $P(\tau_i)$ is the prior probability of selecting different targets. In trivial cases, we can assume that users have the same chance to select all patterns, then $P(T_i)$ and $P(\tau_i)$ can be calculated based on the distribution of patterns in the set. $P(T_j|T_i)$ and $P(\tau_j|\tau_i)$ can be modeled using a distribution function, respectively. Both distributions reflect users' syncing ability with respect to the target pattern, and can be measured through user studies.

A probability model based metric can then be calculated as the optimization goal. One intuitive metric is the overall estimation accuracy $\bar{\eta}$,

$$\bar{\eta} = \frac{1}{n} \sum_{i=1}^{n} P(i|input = i) \tag{4}$$

Intuitively, the optimal pattern set should be the one with the highest $\bar{\eta}$ value. The procedure requires enumerating through all possible pattern sets though, which can be time consuming for pattern sets with large n.

Alternatively, we can maximize the information gain $IG(I|J)$ for a pattern set given the "perfect" input,

$$IG(I|J) = H(I) - H(I|J) = -\sum_{i=1}^{n} P(i) \log_2 P(i) + \sum_{i=1}^{n} P(i) \sum_{j=1}^{n} P(i|j) \log_2 P(i|j) \tag{5}$$

where J is the set of input patterns, I is the set of estimated patterns, i, j is the i^{th} and j^{th} pattern respectively, $P(i) = 1/n$ (assuming users have the same chance to select each pattern). The more information is gained from the input (the less uncertainty remained) the better. The submodular property of entropy [29] enables a faster optimization procedure that can approach the optimal solution within a given error [24].

4 STUDY 1: MODELING SYNCHRONOUS TAPPING BEHAVIOR

In this section, we conduct a user study to gather users' behavior data when syncing with different blinking patterns. We are interested in two aspects: 1) the distribution of periods and delays of user input when syncing with different periods; 2) the effect of the different patterns on users' behavior.

Although temporal synchronous selection techniques can be applied to a number of scenarios, traversing a large number of different interfaces is not practical. Therefore, in this study, we chose finger tapping as a typical means for input. In practical scenarios, finger tapping is convenient and common for users, making it suitable for selecting targets on a number of devices (e.g., smart watch, smart phone, and smart TV).

4.1 Participants and Apparatus

We recruited 12 right-handed students (9 males) from the local institution. Their ages range from 20 to 26 (Mean = 22.6, SD = 2.11). Each was compensated 10 USD for their time. All participants knew the detailed study procedure and consented to complete the whole study.

We used a Logitech K360 Bluetooth keyboard to collect finger tap data. The participants were asked to sit at a desk, and synchronously tap on the SPACE key using their right index finger, which is found to have the fastest cadence[1]. The blinking pattern was displayed on a 32-Inch monitor that was placed 30cm away from the participant.

4.2 Experiment Design

We tested nine levels of periods within 300-700ms: 300ms, 350ms, 400ms, 450ms, 500ms, 550ms, 600ms, 650ms, and 700ms. The 300ms period is already close to human finger synchronization limits[32]. And according to our pilot study, periods longer than 700ms could be too slow and lead to long selection time. Nine circles (3cm diameter on screen) were arranged in a 3x3 layout on the monitor (see Figure 3a). Note that due to the time-homogeneity of users' tapping behavior, τ would not affect users' tapping behavior in terms of T or d. Therefore, we set $\tau \equiv 0$ during this study. Distribution models for other τ values can be obtained by shifting the model here along the time axis.

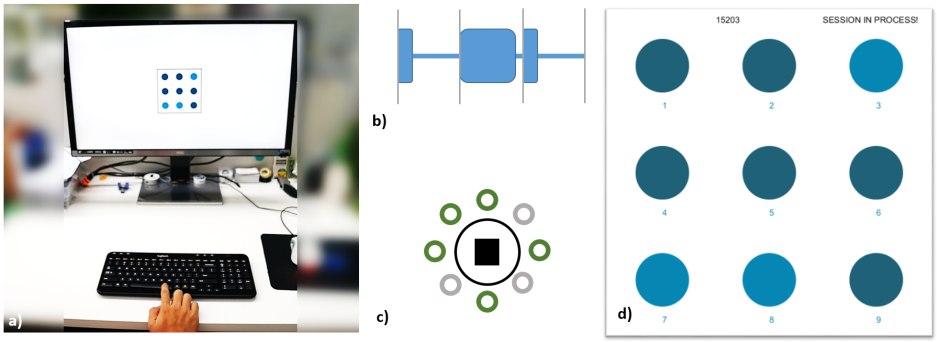


Fig. 3. a) Study 1 experiment setup; b-c) Pattern displays from previous research; d) Selection interface used in this study.

There are many ways to visualize the blinking patterns. For example, Ghomi et.al. visualize the patterns as clipped sound waves[20] (Figure 3b), while SEQUENCE uses eight circles around the target[6] (Figure 3c). Users can see the entire repetition of the pattern, which makes the selection process less mentally demanding. However, such visualizations occupy large amount of space on the screen and is not suitable for non-screen displays (e.g. LEDs). Alternatively, we used blinking grey translucent overlays on each circle to display the patterns (Figure 3d) in this study, which was proposed in [34]. Such pattern display mechanism can be scaled to both large screens and low-power LEDs, and is intuitive for users to sync with. The display arrangement also allows us to observe users' synchronization performance under visual distraction from other blinking targets.

4.3 Procedure

Upon arrival, the participant first completed a demographic questionnaire. We then explained in detail how to sync with the blinking target: keep pressing the SPACE key when the target circle is in one state (dimmed or lighted) and release when the state changes. After two minutes of warm up, the participant completed five sessions of selection tasks. In each session, the participant synced with the nine circles in a sequential order, For each target, the participant tapped for 20s then rested for 20s. The arrangement order of targets are randomized in each session, and a two-minute break was enforced between sessions. The participants completed the experiment within 40 minutes.

4.4 Results

A total of $20\,seconds \times 9\,targets \times 5\,sessions \times 12\,participants = 10,800s$ (540 target selection trials) synchronous finger tapping data was collected. For each edge in the input signal, we calculated the input delay d with all patterns in the pattern set. T was set to the average of the previous two periods for smoother data.

According to the central-limit theorem, for each of the target periods, both the measured T and d was likely to follow a univariate Gaussian distribution. For illustration, Figure 4a-c showed the histograms of measured T and d for 350ms, 500ms and 650ms respectively. Although they did not pass Shapiro-Wilk tests (all $p < .003$), the histograms fitted well with the Gaussian distribution curves. Figure 4f showed the mean difference between measured input and the target pattern. For all the nine T levels, participants tended to yield an input signal with T and d greater than the target pattern. Generally, the input T is very close to, but slightly greater than the target T, with a difference between 6.2 and 11.8ms. In comparison, the lag between users' input and the target pattern was greater, between 9.7 and 43.7ms. Interestingly, except for 300ms (30.8ms), the difference in d decreased monotonously with increasing T, suggesting that users tended to yield greater lag for patterns with shorter period. While for difference between input period and target period, we did not found a consistent trend.

We use Friedman test and Wilcoxon signed-rank test for significance test and Kendall's W coefficient of concordance to assess agreements. Friedman test results showed that target period does not have significant effects on input period error ($\chi^2(8) = 6.84, p = .55, W = .07$), but has significant effects on input delays ($\chi^2(8) = 24.6, p < .005, W = .26$). Post-hoc pair-wise Wilcoxon signed-rank test results show that out of the 21 tests that has at least one $periods < 450ms$, 12 are significant; none of the 15 tests with both $periods \geq 450ms$ are significant though (see Appendix A). The results show that small periods have a larger impact on syncing input delays.

As described in previous section, our Bayesian probability calculation assumes measured T and d are independent from each other. To verify this, we calculated the linear correlation coefficient (R^2) between measured period and delay for each participant. The mean R^2 value was 0.03 with the highest value being 0.17, which indicated the two parameters are not linearly dependent. The dependency should be small if there is any[33], which we believe will not significantly affect our results.

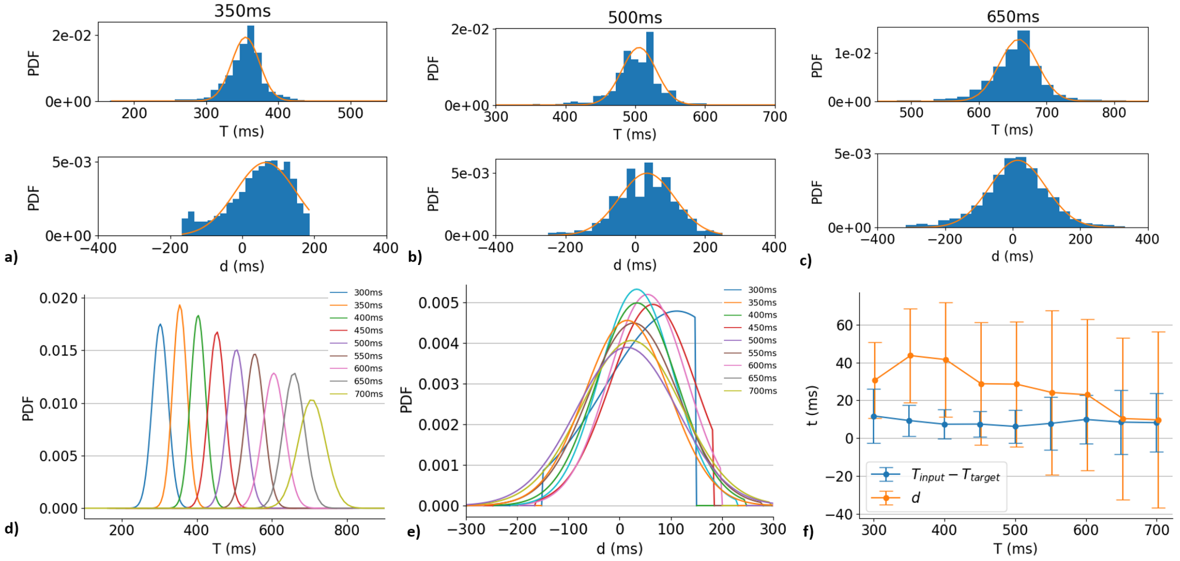


Fig. 4. The histograms of measured distribution and the fitted Gaussian models of periods T and input delay d when the target blinks at 350ms (a), 500ms (b) and 650ms (c) are shown for illustration. Fitted Gaussian models of the measured tapping periods (d) and delays (e) for each target period are also shown. Note that the delay for period T is cut off at $[-T/2, T/2]$. The mean difference between measured input and the target pattern is shown in (f), error bar indicate one standard deviation.

4.5 Generating Optimized Pattern Sets

To determine the optimized pattern set with a specified size, ideally, we should exhaustively search from all possible combinations of patterns in the design space. Considering this impractical, we resorted to discretize the design space to limit the amount of possible patterns: we only chose patterns with the 9 levels of period as in this study. For each period, we chose as many patterns as possible, while ensuring that they are at least separated by 2σ on the τ dimension. σ is the parameter from the fitted d Gaussian distribution (see Figure 4e). The 2σ threshold is empirically decided through pilot study, so that patterns with different initial lags can be effectively differentiated by the users. A total of 22 candidate patterns are then chosen to generate pattern sets (Figure 5).

As described in previous sections, among all possible combinations of the patterns, the optimized pattern set is the one with the largest $\bar{\eta}$. Therefore, for each specific pattern set size (n), we calculated $\bar{\eta}$ for all C_{22}^{n} possible pattern sets to find the optimized one. If multiple pattern sets yield the same $\bar{\eta}$ value, we chose the one with shortest averaged periods. This is based on the finding from the interview that when input accuracy is similar, 10/12 participants preferred shorter periods to longer periods to save time and effort.

We then enumerate through all 4,194,281 possible pattern sets with sizes $2 \leq n \leq 22$, and calculated the two proposed metrics $\bar{\eta}$ and IG for each pattern set. The pattern sets optimized based on each metric are then generated for comparison. Out of the 21 pairs of optimized pattern sets, 9 pairs are identical. Difference of $\bar{\eta}$ averaged across n between the two pattern sets is only 0.003 (SD = 0.004), implying that the expected evaluation results (e.g, input accuracy) on the two types of pattern sets can be very close. So we use the $\bar{\eta}$-optimized pattern sets to evaluate the performance of our target selection technique in this paper, and leave the implementation of faster entropy-based optimization algorithm for future work.

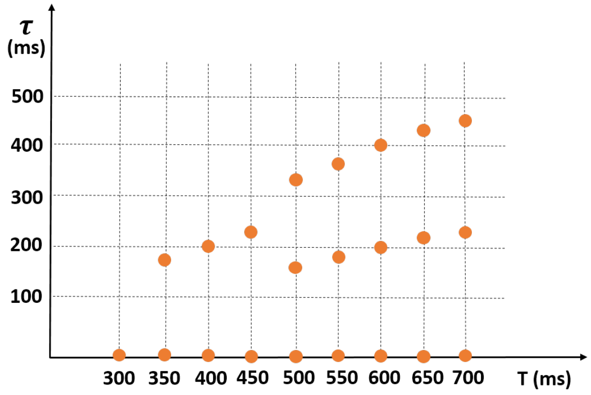


Fig. 5. Candidate patterns on the $T - \tau$ plane.

The results are shown in Table 2. In general, the optimized pattern set with size n is the superset of the pattern set with size $n - 1$. This facilitates users' selection behavior transfer for different numbers of targets. By looking into the composition of the optimized pattern sets, we found that when expanding a pattern set, new patterns with period different than existing patterns are more likely to be selected. As n further increases (e.g. > 9), patterns with the same period but different τ are added, starting from longer periods to shorter periods.

5 INPUT RECOGNIZER IMPLEMENTATION

So far, we have optimized the pattern set design process of temporal synchronous selection techniques. In this section, we describe our implementation of the input recognizer, which is also important for target selection performance. Specifically, we implemented two kinds of recognizers: correlation-based recognizer and Bayesian recognizer. Correlation-based recognizer (referred to as *Correlation recognizer* henceforth) is the most widely-used recognizer in state-of-the-art research of spatial [38] and temporal [34, 50] synchronous object selection techniques. It works by calculating the Pearson correlation coefficients between the signal induced by user movement and the displayed blinking pattern within a time *window*. An object is selected when the coefficient between the signal and its corresponding blinking pattern is greater than a specific *threshold*.

However, the performance of correlation recognizer may degrade due to imprecise inputs. This usually happens for short input sequence, during which users tend to make more input errors while trying to “find the rhythm". Accordingly, the user behavior model in Study 1 can potentially improve this situation by calculating the Bayesian probability. Therefore, we also implemented a Bayesian recognizer.

Instead of the Pearson correlation coefficient, the Bayesian recognizer calculates the probability of each blinking pattern being the target given the input signal within a time *window*. An object is selected when the probability of a target is greater than a *threshold*. Similar to Equation (3), the probability of the i^{th} pattern being the target can be calculated as:

$$\begin{aligned} P(i|input) &= P(T_i, \tau_i | T_{input}, \tau_{input}) && (6) \\ &= P(T_i|T_{input}) \times P(\tau_i|\tau_{input}) \\ &= \frac{P(T_{input}|T_i)P(T_i)}{P(T_{input})} \times \frac{P(\tau_{input}|\tau_i)P(\tau_i)}{P(\tau_{input})} && (7) \\ &\propto P(T_{input}|T_i)P(T_i) \times P(\tau_{input}|\tau_i)P(\tau_i) && (8) \end{aligned}$$

Table 2. Optimized pattern sets with sizes $2 \leq n \leq 22$ and the corresponding optimized window size and threshold for Bayesian (win_{baye}, TH_{baye}) and correlation-based recognizers (win_R, TH_R).

n	Pattern set (T, τ)	win_{baye}	TH_{baye}	win_R	TH_R
2	(300,0),(550,0)	2s	0.1	2s	0.3
3	(300,0),(450,0),(650,0)	2s	0.6	2s	0.4
4	(300,0),(400,0),(550,0),(700,0)	2s	0.6	2s	0.6
5	(300,0),(400,0),(500,0),(600,0),(700,0)	3s	0.3	2s	0.7
6	Pset5,(350,0)	2s	0.7	5s	0.2
7	Pset6,(450,0)	4s	0.4	5s	0.2
8	Pset7,(550,0)	4s	0.5	5s	0.2
9	Pset8,(650,0)	5s	0.4	5s	0.2
10	Pset9,(700,467)	5s	0.4	5s	0.3
11	Pset10,(650,433)	5s	0.4	6s	0.2
12	Pset11,(600,400)	6s	0.3	6s	0.3
13	Pset12,(700,233)	6s	0.3	6s	0.3
14	Pset13,(550,367)	7s	0.1	7s	0.2
15	Pset14,(650,216)	6s	0.3	7s	0.2
16	Pset14,(450,225),(500,333)	7s	0.1	7s	0.3
17	Pset15,(500,333),(600,200)	7s	0.2	7s	0.3
18	Pset17,(450,225)	7s	0.2	7s	0.3
19	Pset18,(550,183)	7s	0.2	7s	0.3
20	Pset19,(400,200)	7s	0.2	7s	0.3
21	Pset20,(500,167)	7s	0.2	7s	0.3
22	Pset21,(350,175)	7s	0.2	7s	0.3

where T_{input} and τ_{input} can be measured from the input signal, $P(T_{input}|T_i)$ and $P(\tau_{input}|\tau_i)$ can be calculated using the Gaussian model from Study 1 (see Figure 4).

Now, we can see that to fully implement the recognition algorithms, the value of *window size* and *threshold* should be carefully determined. Moreover, the optimized value of these parameters should be dependent of *n*. Therefore, we will conduct an optimization process to determine the values of these parameters.

With the increasing popularity of machine learning algorithms, it is also attractive to train such a recognizer with collected input data. In practice, however, machine learning approaches (e.g., LSTM and SVM) would require extensive training for a great number of parameters to achieve optimal performances. Moreover, unlike the two models above whose parameters have been determined, the parameters of the machine learning model and *window size*, *threshold* is coupled, and can only be trained at the same time. This demands a huge amount of training data, and requires considerable computation resources. Therefore, we defer this to future work.

5.1 Recognizer Parameter Optimization

As described above, the values of window size (*win*) and threshold (*TH*) need to be optimized so that the recognizer can achieve high accuracy and low selection time. A small *win* or a low *TH* would cause the recognizer to be too sensitive, while a big *win* or a high *TH* will lead to longer selection time. To determine the optimal value of *win* and *TH*, for each *n*, we ran simulations for both recognizers using all the data collected in Study 1. We tested sliding windows with 6 levels of length (2s to 7s) and 8 levels of threshold (0.1 to 0.8). For each n, the optimization process is:

(1) Calculate the mean accuracy and selection time for all taping data for 48 different recognizers, corresponding to 48 combinations of win and TH.
(2) Discard recognizers whose accuracy was too low (< 88% according to pilot study). For all the remaining recognizers, sort them in ascending order by accuracy. Designers can set different accuracy thresholds for a speed-accuracy tradeoff.
(3) Assigning the recognizer with the lowest accuracy (thus shortest selection time) as the optimal recognizer. If there exists another recognizer, whose selection time is slightly longer, but accuracy was far higher, then replace the optimal recognizer to the new one. Loop until the result do not change. The threshold ratio $\Delta_{acc}/\Delta_{time}$ was empirically set to 0.15. Designers can adjust the threshold ratio for finer tradeoff of selection speed and accuracy.

Table 2 showed the results of the optimized recognizer parameters for each n. As expected, the optimized win value both increased with n for correlation recognizer and Bayesian recognizer. On the other hand, optimized TH value did not yield a consistent trend with n for either recognizer.

Figure 6 showed the simulated accuracy and selection time with optimized win and TH for each n. Generally, Bayesian Recognizer showed advantage over Correlation recognizer in terms of both accuracy and selection time for small number of targets ($n \leq 5$). The performance of both recognizers became similar when n increased. Note that optimized win value for small n is very small (see Table 2), this complies with our assumption that Bayesian recognizer is more robust to scenarios with more imprecise inputs (e.g., data sequence is short).

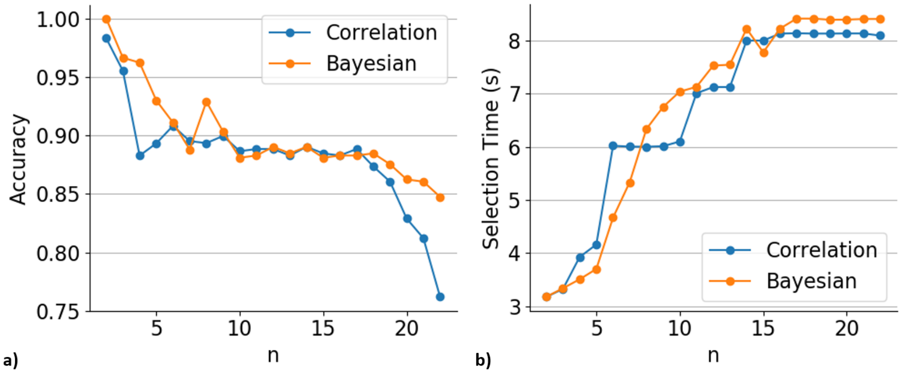


Fig. 6. The simulated accuracy (a) and selection time (b) with optimized win and TH for each n using data collected in Study 1.

5.2 Accuracy Upper Bound

To look deeper into the characteristics of correlation recognizer and Bayesian recognizer, we compared the upper bound of the accuracy of both recognizers for different level of n. The upper bound accuracy was calculated by fixing win, then chose the highest accuracy yielded by all TH values. Figure 7 showed the result for $n = 3, 5, 9, 12$, with the corresponding mean selection time. Similar with previous results (see Figure 6), for $n = 3$, the Bayesian recognizer achieves higher accuracy than the correlation recognizer, while keeping competitive selection time. For $n \geq 9$, the Bayesian recognizer still seems to yield higher accuracy than the correlation recognizer, especially

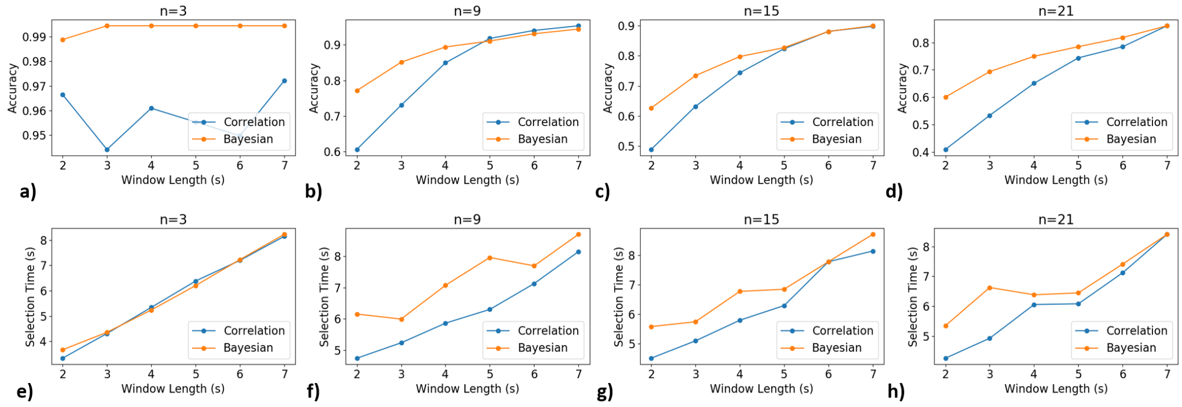


Fig. 7. Calculated upper bound of accuracy (a-d) and the corresponding selection time (e-h) of different window lengths for $n = 3, 9, 15, 21$ by running Correlation (blue) and Bayesian (orange) recognizers on the optimized pattern sets.

when *win* was small (i.e., data sequence is short). However, the selection time of Bayesian recognizer was also longer than the correlation recognizer, suggesting a speed-accuracy tradeoff. The performance of Correlation recognizer improves as the input sequence gets longer and the inputs become more precise.

6 STUDY2: EVALUATING OPTIMIZED PATTERN SETS AND RECOGNIZERS

So far, we have proposed methods to improve two key components of temporal synchronous selection technique: pattern set design and recognizer. In this section, we conducted another user study to evaluate the performances of the optimized pattern sets and Bayesian recognizer. Specifically, we want to answer three questions: 1) Can the behavior models be generalized among different users? 2) How much does the optimized pattern improve the selection performance? 3) How the performances of the Bayesian and the Correlation recognizers differ when selecting from different numbers of targets? To answer the above questions, we designed the tasks to mimic a realistic smart TV scenario, where participants were asked to tap a single key to select from a list of displayed movie posters. We tested two recognizers (Bayesian and Correlation), two types of pattern sets (optimized, period-increasing), and four sizes of pattern sets (3, 9, 15, 21).

6.1 Participants and Apparatus

We recruited 14 right-handed participants (9 males) from the local institution that did not take part in the previous study. Their ages range from 19 to 30 (Mean = 22.9, SD = 2.95). Each participant was compensated 15 USD. All participants knew the detailed study procedure and consented to complete the whole study. We used the same 32-inch monitor as in Study 1 to display the experiment interface. A Microsoft Designer Bluetooth Keyboard was used to detect tapping input. The selection interface was implemented using Python Tkinter.

6.2 Experiment Design and Procedure

The experiment interfaces showed different numbers of movie posters (n = 3, 9, 15, and 21). For each poster, a $7.5mm \times 7.5mm$ blinking red square on its upper-right corner displayed the blinking pattern (Figure 8). The participants were asked to select a specific target from all the candidates by syncing to the corresponding pattern. To reduce the visual search time, we always chose the only poster with white background as the target for all tasks. The position and blinking pattern of the target poster was randomized for each task. We evaluated two types of pattern sets for each n: 1) Optimized pattern set as shown in Table 2; 2) Pattern set generated by starting from small periods to larger periods (period-increasing), which is expected to yield high confusion. For example, the pattern set for $n = 3$ of period-increasing pattern set is {(300, 0), (350, 0), (350, 175)}. During the study, we used two kinds of recognizer (Bayesian and Correlation) with optimized parameters (see Figure 4 and Table 2) to predict users' input.

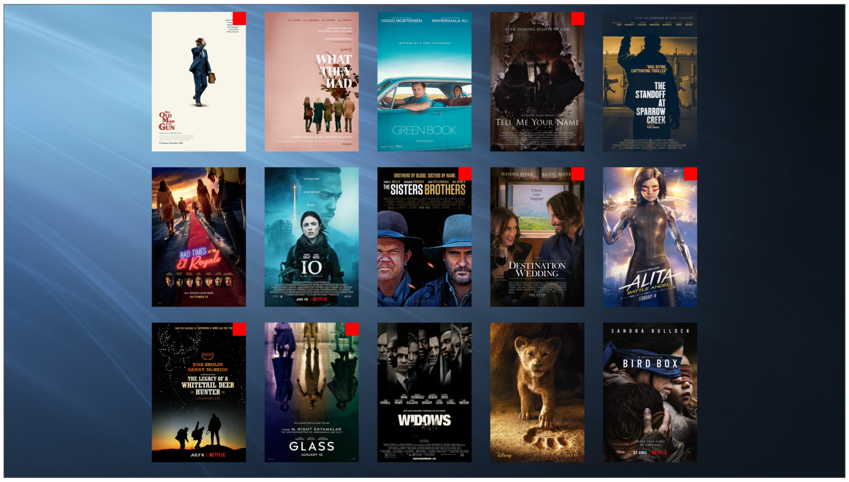


Fig. 8. Selection interface for $n = 15$. The target poster is the first poster in the first row.

Upon arrival, the participant first completed a demographic questionnaire. We then explained the task and asked the participant to warm up for two minutes. The participant then sat 1.5m away from the monitor, and completed 8 sessions of target selection tasks. In each session, they complete 16 blocks of tasks, each corresponds to a $\{pattern\,set, recognizer, n\}$ combination. The order of the combinations were randomized. In each block, the participant were asked to select the target poster by pressing the SPACE key on the keyboard in sync with the corresponding blinking square. Each block is finished after 15 seconds or when a poster (whether correct or not) is selected. The participant then pressed *ENTER* to enter the next block. A 30-second and 1-minute break was enforced in the middle of a session and between sessions, respectively. At the end of the study, we conducted an interview to collect subjective ratings using a 7-point Likert scale. The experiment took around one hour to complete.

6.3 Results and Discussion

A total of $4\,n \times 2\,recognizers \times 2\,pattern\,sets \times 8\,sessions \times 14\,participants = 1,792$ selection tasks were conducted. We report Greenhous-Geisser corrected degree of freedom and p-value if the Mauchly test shows violation of sphericity assumption for parametric tests (0.05 significance level).

To compare the performance of different pattern sets, we merged different n, and calculated the mean accuracy of different recognizers (see Figure 9a). As expected, for both recognizers, the selection accuracy on optimized pattern sets are significantly higher than that on period-increasing patterns (Paired t-test ($t(13) = 3.38, p < .005, Cohen's\,d = 1.11$) for Bayesian recognizer, ($t(13) = 6.45, p < .001, Cohen's\,d = 1.70$) for Correlation recognizer). The Bayesian recognizer yield similar performance as the correlation recognizer on the optimized pattern sets (93.8% vs. 93.3%). However, on non-optimized pattern sets, Bayesian recognizer achieved higher accuracy than the correlation recognizer (85.9% vs. 77.2%). Note that on non-optimized pattern sets, input signal for different targets are more alike. Therefore, this again confirmed the advantage of Bayesian recognizer for predicting imprecise input.

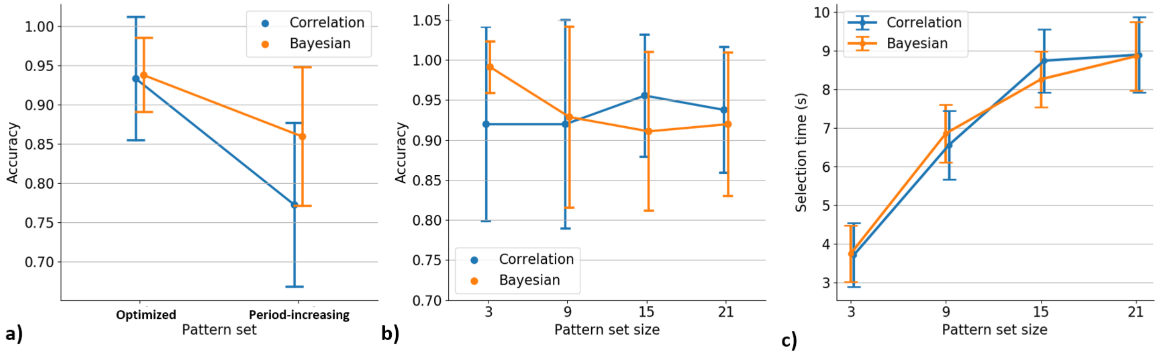


Fig. 9. (a) Accuracy of different recognizers on different pattern sets, merged different n; (b) accuracy of different recognizers on different n and optimized pattern set; (c) selection time of different recognizers on different n and optimized pattern set. Error bar indicate one standard deviation.

We then analyzed the selection time and accuracy of different recognizers with respect to n on the optimized pattern set. No significant effect of n ($F_{2.34,30.5} = 0.77, p = .49, \eta_p^2 = .06$) or recognizer ($F_{1,13} = 0.04, p = .84, \eta_p^2 = .003$) was found on selection accuracy. When $n = 3$, the Bayesian recognizer showed higher accuracy than the correlation recognizer. However, for greater n, it showed a trend to fall below the latter. Again, the results imply that Bayesian recognizer can better deal with imprecise inputs in short input sequence (only a few taps), while Correlation recognizer is more vulnerable to imprecise inputs. We average selection time across n and no significant effect of recognizer ($t(13) = -0.62, p = .54, Cohen's\, d = .10$) was found on the optimized pattern set(see Figure 9c).

Generally, the selection performance in this study was quite similar with the simulation results (see Figure 6), but with a slightly higher accuracy and a slightly longer selecting time. We speculate this was due to the change of users' speed-accuracy tradeoff in a realistic setting with selection feedback, compared with in the Wizard-of-Oz setting in Study 1.

The goal of the interview is to gather participants' feedback towards temporal synchronous selection techniques based on their experience in this user study. In Figure10, the overall ratings showed that all participants were positive that such technique is suitable as a multi-modal association-free selection technique. Most of them felt the selection technique was accurate (10/14 ratings > 4) and fast (11/14 ratings > 4) with low mentally (8/14 ratings < 4) and physically (9/14 ratings < 4) demands. Only four participants experienced different levels of eye fatigue, which was acceptable since the experiment lasted around one hour. Surprisingly, 9/14 participants felt the blinking patterns only introduced low distractions during selection. This may due to the carefully designing of the small blinking square, which also mimics their experience in daily life (e.g. video recording indicator).

7 EXPLORING MORE MODALITIES AND SENSING TECHNIQUES

We conducted pilot studies to help understand our technique's generalizability across modalities and sensing techniques: 1) The optimized patterns are generated using the finger tap synchronous model, which might cause suboptimal performance for other modalities. We first built the syncing behavior models for three more modalities-*Thumb Tap*, *Foot Tap*, and *Teeth Clench* with pressure sensors. The modalities cover use cases when

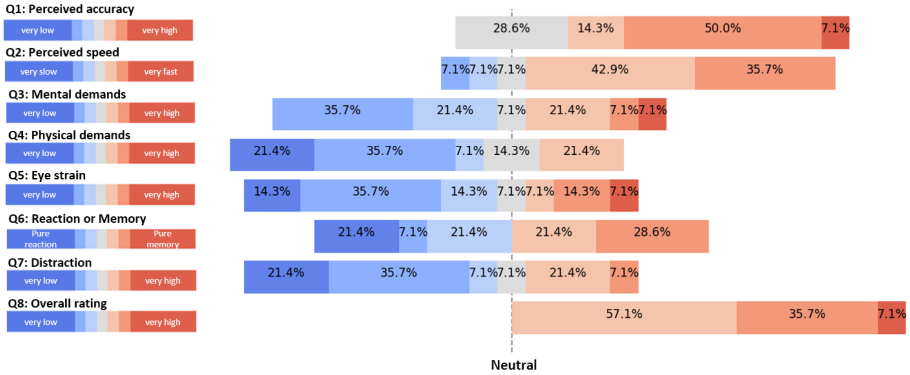


Fig. 10. Subjective ratings of the selection technique.

one hand or both hands are busy, and provide low-effort, subtle, and accessible target selection experience. Then we compare the Bayesian recognizer's performance on optimal pattern sets using the modality-specific model with that of using the index finger tapping model built in Study 1; 2) The keyboard inputs used in the studies have very low signal noises. To understand our technique's performance with different signal noise levels, we implemented two sensing techniques-capacitive sensing (medium signal noise) and RFID RSSI based sensing (large signal noise) for the thumb and the foot tap input interfaces separately. Their performance is tested using a Correlation recognizer to avoid impacts caused by different behavior models. All participants knew the study procedure and consented to complete the whole study.

7.1 Modality Generalization Pilot Study

We collected syncing data under different modalities from four participants (2 males) with ages ranging from 20 to 29. We used FSR402 pressure sensor[1] to detect syncing behaviors in all modalities. For thumb tapping modality, the sensor is placed on a ring worn on the index finger; for the foot tapping modality, the sensor is placed on the floor; for the teeth clench modality, the sensor is wrapped with food grade silicone sheets and placed between the upper and lower front teeth [45]. Similar to Study 1, we asked each participant to sync with nine patterns blinking at periods from 300ms to 700ms with a 50ms interval. For each modality, a total of five sessions were conducted. Within each session, the participant synced for 10 seconds with each blinking pattern. We then built syncing behavior models for each modality with the same method used in Study 1.

We then tested the Bayesian recognizer in a second pilot study with four participants (2 males, 1 participated the previous study) with ages ranging from 21 to 30. For each modality, a total of six sessions were conducted with two blocks in each session. Within each session, the participant tried to select the target movie poster four times from $n = 3$ and $n = 15$ posters (two times for each n) using the same index finger tapping model in one block, then repeat the procedure using the modality-specific model for the other block (order balanced).

Using the same index finger tapping model, the selection accuracy averaged across all modalities is 0.92 (SD = 0.08) when $n = 3$ and 0.89 (SD = 0.06) when $n = 15$. As expected, in general the measured selection accuracies are higher when using modality-specific models (Table 3). It is especially true for shorter input sequences (small n). When using non-modality-specific models, the overall selection accuracy drops from 0.98 to 0.92 when $n = 3$,

[1] http://www.trossenrobotics.com/productdocs/2010-10-26-DataSheet-FSR402-Layout2.pdf

while the performance is similar when $n = 15$ (0.88 vs 0.89). This aligns with our previous observation that the behavior model has more impact on selection accuracies when the input sequence is short.

Table 3. Selection accuracy averaged across n (SD in parenthesis)

Model	Thumb	Foot	Teeth
Index Finger Tapping	0.91 (0.08)	0.91 (0.04)	0.91 (0.06)
Modality-specific	0.97 (0.04)	0.93 (0.07)	0.88 (0.10)

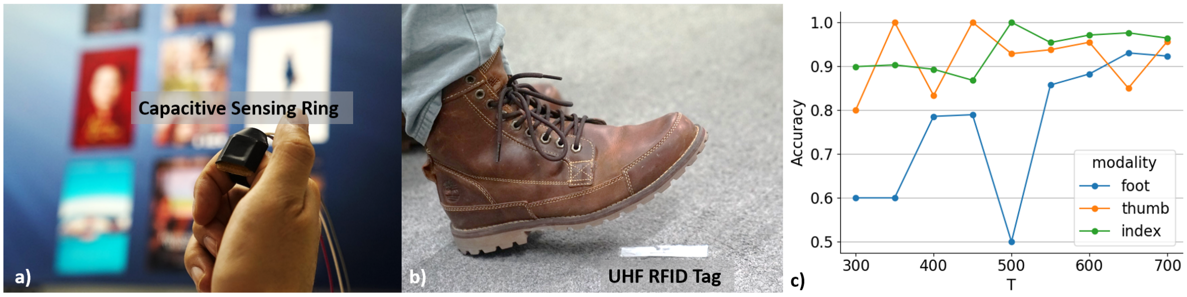


Fig. 11. The participant can tap the thumb on a ring (a) or tap a foot on a UHF RFID tag (b) to select on-screen objects. The selection accuracy for different modalities with respect to target pattern period using Correlation recognizer is also shown (c). We added the index data from Study 2 for comparison

7.2 Sensing techniques Generalization Pilot Study

To explore the selection technique's performance using different sensing techniques, we conducted another pilot study that uses capacitive sensing for thumb tapping and RFID RSSI based sensing for foot tapping. We recruited three male participants with ages 23, 23, and 24 respectively from the local institution. They all have participated in Study 2. Each participant was compensated 10 USD. We used the same display monitor and experiment interface as in Study 2. The only difference is the sensing technique:

Thumb Tap We built a capacitive sensing ring by placing an off-the-shelf capacitive sensing module into a 3D printed case. The module outputs high voltage only when being touched. The analog output signal was monitored by an Arduino UNO and reported to a PC via a serial port. Users can then tap their thumbs on the ring to select objects. The signal noise is medium due to the capacitive sensing nature and the sensor location.

Foot Tap We placed an Alien AZ-9654 Ultra High Frequency (UHF) RFID tag on the floor carpet, and monitored its Received Signal Strength Indicator (RSSI) using an Impinj R420 reader. The reader then reported the RSSI of the tag to PC via TCP connection. Rhythmic foot taps can introduce periodic RSSI changes due to the signal blockages and antenna mismatch [50], which can then be used for temporal synchronization based target selection. RFID RSSI is a noisy signal that have large fluctuation even in a static environment [41].

We used the optimized pattern set with $n = 3, 5, 9, 21$ to better compare the results with those from Study 2. For each modality, the participants were asked to complete 5 sessions of selection task (order balanced by Latin

Square), with 12 selection tasks in each session, three for each n. The participants shared their experience of using different modalities after the experiment. The experiment took about 30 minutes to complete for each participant.

The overall accuracy across different T was 91% for thumb tap and 81% for foot tap. Compared with results in Study 2 (see Figure 9b), the index finger and thumb finger tapping yield similar selection accuracies, both higher than foot tapping. The selection error in foot tap mainly happened on patterns with small periods (i.e. fast blinks) (Figure 11c), which could be too fast for foot to sync with. All three participants mentioned that the fast patterns are too fast to follow by tapping foot in the interview, which supported our speculation. When the target blinking period is large (e.g. >600m), foot tapping (RSSI based sensing) achieved a selection accuracy higher than 90%, which is comparable to index finger (keyboard based sensing) and thumb tapping (capacitive sensing). The results imply that the our technique can perform well at different signal noise levels. One possible reason is that the optimized pattern sets enable separation between patterns even with large noises.

8 DISCUSSION

8.1 Correlation Recognizer vs Bayesian Recognizer

Aside from the performance difference observed in Study 2, the Correlation and Bayesian recognizer are different in many other aspects.

Signal Type Correlation recognizer requires signals similar to Unipolar Non-Return-to-Zero (NRZ) code [2], which maps to current sensing *status*. For example, the RFID RSSI and capacitive sensing signal are both NRZ signals. Bayesian recognizer, however, requires signals similar to Unipolar Return-to-Zero (RZ) code [3], which maps to sensing status *changes*. One example is the audio signal pulses generated by clapping hands periodically. The two types of signals can be converted to each other during preprocessing. Edge or peak detection is usually used to covert NRZ signals to RZ signals, while the RZ signals can be easily converted to NRZ signals by switching state at each pulse.

Data Requirements Bayesian recognizer can only be used with behavior models, while it is not necessary to collect any data in advance when using Correlation recognizer.

Resource Requirements The Bayesian recognizer demands higher computational power and more storage capacity for the behavior models. The computation and storage resources demands for Correlation recognizer is lower, which makes it especially suitable for embedded real-time system with limited resources [50].

Robustness to Imprecise Inputs Bayesian recognizer is more robust to imprecise inputs, especially when the input sequence is short. Correlation recognizer is vulnerable to imprecise inputs with short input sequence. The two recognizers have similar performances for longer input sequences (larger window sizes) since the inputs become more precise after the users “find the rhythm".

In general, Bayesian recognizer performs better for single modality selection tasks with few targets (e.g. $n < 5$), while Correlation recognizer is better suited for multi-modality selection tasks with more selective targets.

8.2 Supporting More Targets

The proposed 2D design space supports 22 unique patterns, which could be enough for many applications since the number of displays is usually limited within a user’s line of sights. A hierarchical blinking strategy can be employed to support more target devices. For example, one third of devices can blink with the same pattern first. The selected devices can then blink with unique patterns, which supports selection of 66 targets. The Bayesian recognizer will perform better with such strategy since there will be multiple short input sequences.

[2] https://en.wikipedia.org/wiki/Non-return-to-zero

[3] https://en.wikipedia.org/wiki/Return-to-zero

8.3 Sensor Fusion

A sensor hub can be used to process signals collected by different types of sensors (e.g., video, audio, touch) concurrently so that users can choose the modality that provides the most *confident* and *comfortable* selection experience, which could lead to more accurate syncing behaviors. For example, one participant in pilot studies stating that he felt "more certain" when using foot tapping for slow blink patterns and finger tapping for fast blink patterns. Signals from different sensors can also be cross-checked to improve recognition performance[42]. For example, knocking on a table can generate periodic sounds and vibrations at the same time, which can both be sensed by a smart phone placed on the table. The audio and IMU data can then be analyzed together to improve the recognition accuracy.

9 LIMITATIONS AND FUTURE WORK

In Section 7, we validated the potential of applying our algorithm to different modalities and sensing techniques using a pilot study. However, the relatively small sample size may limit the external validity of our results (e.g., users with different ages may yield different syncing ability). As the Bayesian decoder is a statistic model that models the intrinsic pattern in users' syncing behavior, we expect it to be generalizable to different groups of users by re-measuring the model parameters. We plan to further validate the performance of our algorithm with larger amount of users with more variations in demographics in future work.

Study 2 show that the models can be generalized between users roughly within the same age group. People at different age groups may require different syncing behavior models. For example, senior users may perform better when syncing with patterns with small periods. One potential solution is to extrapolate models of one age group to estimate behaviors of other age groups, which we leave for future work.

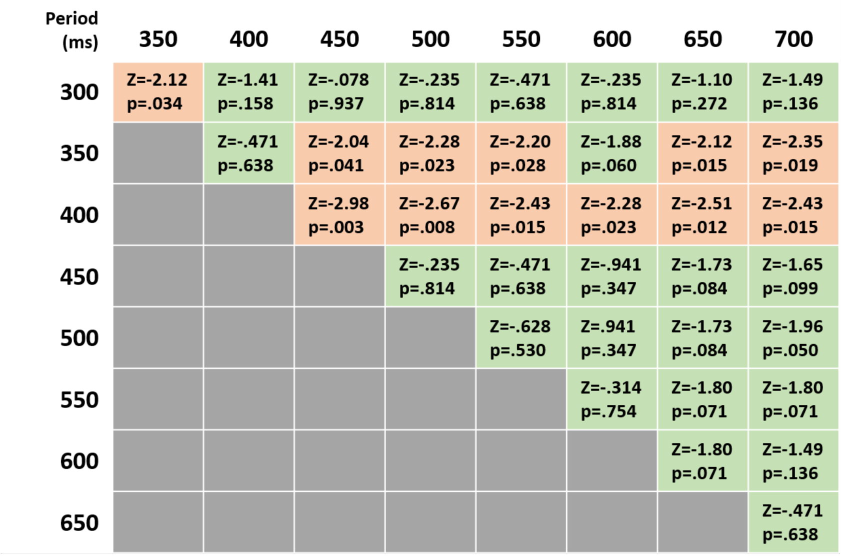
To maximize the internal validity of our results, the user studies in this paper were carried out in a controlled lab environment. In realistic settings, there would be more noise sources (e.g., electromagnetic interferences, user mobility and concurrent syncing interference), which may affect the selection performance of our technique. However, as we have demonstrated in Section 7, our proposed algorithm can potentially be applied to different modalities by re-measuring the model parameters. Another way to further improve the robustness of our algorithm is to add signal preprocessing components (e.g., band-pass filter). Finally, users could choose different interaction modalities in different scenarios to achieve optimal performance (e.g. tap fingers instead of clap hands when there is loud background music). We consider it worthwhile to further improve our algorithm, and validate its performance in a field study in the future.

Some participants in Study 2 mentioned that they might perform better with some feedback after each periodic movement. Visual feedback can be provided for screen displayed patterns. For example, the size or color[44] of the items can change according to the correlation value. A progress bar can also show the selection probability for each on-screen object. The visual feedback will also introduce extra visual distractions though, thus requiring careful design. We leave the feedback design for future research.

Selection time can impacts the user experience. Due to the low sensing resource requirements of detecting temporal synchronization, the selection time is generally longer than that of other techniques (e.g. pointing, spatial motion matching [18]). We believe the selection time can be reduced with more practices and personalized models.

10 CONCLUSION

In this paper, we proposed an approach to design patterns for temporal synchronous target selection techniques. We defined the design space on a Period-Lag 2D plane, and conducted a study to model synchronous finger tapping behaviors. With the model, we used Bayesian algorithm to generate pattern sets that minimize the selection confusion. We ran simulations and found that Bayesian recognizer out-performed the classical correlation



Period (ms)	350	400	450	500	550	600	650	700
300	Z=-2.12 p=.034	Z=-1.41 p=.158	Z=-.078 p=.937	Z=-.235 p=.814	Z=-.471 p=.638	Z=-.235 p=.814	Z=-1.10 p=.272	Z=-1.49 p=.136
350		Z=-.471 p=.638	Z=-2.04 p=.041	Z=-2.28 p=.023	Z=-2.20 p=.028	Z=-1.88 p=.060	Z=-2.12 p=.015	Z=-2.35 p=.019
400			Z=-2.98 p=.003	Z=-2.67 p=.008	Z=-2.43 p=.015	Z=-2.28 p=.023	Z=-2.51 p=.012	Z=-2.43 p=.015
450				Z=-.235 p=.814	Z=-.471 p=.638	Z=-.941 p=.347	Z=-1.73 p=.084	Z=-1.65 p=.099
500					Z=-.628 p=.530	Z=.941 p=.347	Z=-1.73 p=.084	Z=-1.96 p=.050
550						Z=-.314 p=.754	Z=-1.80 p=.071	Z=-1.80 p=.071
600							Z=-1.80 p=.071	Z=-1.49 p=.136
650								Z=-.471 p=.638

Fig. 12. Post-hoc test results. 12 tests are significant (red) and 24 tests are insignificant (green).

recognizer for small size pattern sets. The evaluation results showed that the selection accuracy with the optimized pattern sets are higher than the period-increasing pattern sets by 16% using Correlation recognizer and 8% using Bayesian recognizer. The results of the informal evaluations across more modalities and sensing techniques implied that users can effectively transfer their selection experience across different interfaces. We believe user modeling can facilitate the application of temporal synchronous target selection techniques on more interfaces as an association-free multi-modal object selection technique.

A POST-HOC RESULTS FOR TARGET PERIOD-INPUT DELAY TEST IN STUDY 1

We show the complete post-hoc paired Wilcoxon signed-rank test results in Figure 12.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their careful review comments. This work is supported by National Key R&D Plan of China (No. 2018YFB1005000, No. 2017YFB1002802), Natural Science Foundation of China (No. 61572471, No. 61672314, No. 61572276, No. 61521002 and No. 61902208), R&D Plan in Key Field of Guangdong Province (No. 2019B010109001), Chinese Academy of Sciences Research Equipment Development Project (No. YZ201527), Beijing Municipal Science&Technology Commission (No. Z171100000117017).

REFERENCES

[1] Tomoko Aoki, Peter R. Francis, and Hiroshi Kinoshita. 2003. Differences in the abilities of individual fingers during the performance of fast, repetitive tapping movements. *Experimental Brain Research* 152, 2 (July 2003), 270–280. https://doi.org/10.1007/s00221-003-1552-z

[2] Nivedita Arora, Steven L. Zhang, Fereshteh Shahmiri, Diego Osorio, Yi-Cheng Wang, Mohit Gupta, Zhengjun Wang, Thad Starner, Zhong Lin Wang, and Gregory D. Abowd. 2018. SATURN: A Thin and Flexible Self-powered Microphone Leveraging Triboelectric Nanogenerator. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2 (July 2018), 60:1–60:28. https://doi.org/10.1145/3214263

[3] Daniel Ashbrook, Carlos Tejada, Dhwanit Mehta, Anthony Jiminez, Goudam Muralitharam, Sangeeta Gajendra, and Ross Tallents. 2016. Bitey: An Exploration of Tooth Click Gestures for Hands-free User Interface Control. In *Proceedings of the 18th International*

Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16). ACM, New York, NY, USA, 158–169. https://doi.org/10.1145/2935334.2935389

[4] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. 2013. DopLink: Using the Doppler Effect for Multi-device Interaction. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 583–586. https://doi.org/10.1145/2493432.2493515

[5] N. Barbara and T. A. Camilleri. 2016. Interfacing with a speller using EOG glasses. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 001069–001074. https://doi.org/10.1109/SMC.2016.7844384

[6] Alessio Bellino. 2018. SEQUENCE: a remote control technique to select objects by matching their rhythm. *Personal and Ubiquitous Computing* (March 2018). https://doi.org/10.1007/s00779-018-1129-2

[7] Peter Bennett, Stuart Nolan, Ved Uttamchandani, Michael Pages, Kirsten Cater, and Mike Fraser. 2015. Resonant Bits: Harmonic Interaction with Virtual Pendulums. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15)*. ACM, New York, NY, USA, 49–52. https://doi.org/10.1145/2677199.2680569

[8] 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

[9] Kaifei Chen, Jonathan Fürst, John Kolb, Hyung-Sin Kim, Xin Jin, David E. Culler, and Randy H. Katz. 2018. SnapLink: Fast and Accurate Vision-Based Appliance Control in Large Commercial Buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4 (Jan. 2018), 129:1–129:27. https://doi.org/10.1145/3161173

[10] 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 (Sept. 2017), 45:1–45:22. https://doi.org/10.1145/3130910

[11] 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

[12] 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

[13] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5909–5920. https://doi.org/10.1145/2858036.2858177

[14] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches Using Smooth Pursuit Eye Movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 457–466. https://doi.org/10.1145/2807442.2807499

[15] 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 event-place: Québec City, QC, Canada.

[16] Jackson Feijó Filho, Wilson Prata, and Thiago Valle. 2012. Breath Mobile: A Low-cost Software-based Breathing Controlled Mobile Phone Interface. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services Companion (MobileHCI '12)*. ACM, New York, NY, USA, 157–160. https://doi.org/10.1145/2371664.2371697

[17] Jackson Feijó Filho, Thiago Valle, and Wilson Prata. 2012. Breath Mobile: A Software-based Hands-free and Voice-free Breathing Controlled Mobile Phone Interface. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '12)*. ACM, New York, NY, USA, 217–218. https://doi.org/10.1145/2384916.2384961

[18] Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2016. Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2319–2331. https://doi.org/10.1145/2858036.2858308

[19] Chuhan Gao, Yilong Li, and Xinyu Zhang. 2018. LiveTag: Sensing Human-Object Interaction through Passive Chipless WiFi Tags. In *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI '18)*. Renton, WA, USA.

[20] Emilien Ghomi, Guillaume Faure, Stéphane Huot, Olivier Chapuis, and Michel Beaudouin-Lafon. 2012. Using Rhythmic Patterns As an Input Method. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1253–1262. https://doi.org/10.1145/2207676.2208579

[21] Jessica A. Grahn and Matthew Brett. 2007. Rhythm and Beat Perception in Motor Areas of the Brain. *Journal of Cognitive Neuroscience* 19, 5 (May 2007), 893–906. https://doi.org/10.1162/jocn.2007.19.5.893

[22] Antti Jylhä and Cumhur Erkut. 2009. A Hand Clap Interface for Sonic Interaction with the Computer. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 3175–3180. https://doi.org/10.1145/1520340.1520452

[23] Tiiu Koskela and Kaisa Väänänen-Vainio-Mattila. 2004. Evolution Towards Smart Home Environments: Empirical Evaluation of Three User Interfaces. *Personal Ubiquitous Comput.* 8, 3-4 (July 2004), 234–240. https://doi.org/10.1007/s00779-004-0283-x

[24] Andreas Krause and Carlos Guestrin. 2005. Near-optimal Nonmyopic Value of Information in Graphical Models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI'05)*. AUAI Press, Arlington, Virginia, United States, 324–331. http://dl.acm.org/citation.cfm?id=3020336.3020377 event-place: Edinburgh, Scotland.

[25] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. ACM, 213–224. https://doi.org/10.1145/3242587.3242609

[26] Luis Leiva, Matthias Böhmer, Sven Gehring, and Antonio Krüger. 2012. Back to the App: The Costs of Mobile Application Interruptions. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 291–294. https://doi.org/10.1145/2371574.2371617

[27] Felix Xiaozhu Lin, Daniel Ashbrook, and Sean White. 2011. RhythmLink: Securely Pairing I/O-constrained Devices by Tapping. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 263–272. https://doi.org/10.1145/2047196.2047231

[28] Carlos H. Morimoto and Marcio R. M. Mimica. 2005. Eye Gaze Tracking Techniques for Interactive Applications. *Comput. Vis. Image Underst.* 98, 1 (April 2005), 4–24. https://doi.org/10.1016/j.cviu.2004.07.010

[29] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming* 14, 1 (Dec. 1978), 265–294. https://doi.org/10.1007/BF01588971

[30] Shwetak N. Patel and Gregory D. Abowd. 2003. A 2-Way Laser-Assisted Selection Scheme for Handhelds in a Physical Environment. In *UbiComp 2003: Ubiquitous Computing (Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, 200–207. https://doi.org/10.1007/978-3-540-39653-6_16

[31] Filipe Quintal, Augusto Esteves, Fábio Caires, Vítor Baptista, and Pedro Mendes. 2019. Wattom: A Consumption and Grid Aware Smart Plug with Mid-air Controls. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '19)*. ACM, New York, NY, USA, 307–313. https://doi.org/10.1145/3294109.3295642 event-place: Tempe, Arizona, USA.

[32] Bruno H. Repp. 2003. Rate Limits in Sensorimotor Synchronization With Auditory and Visual Sequences: The Synchronization Threshold and the Benefits and Costs of Interval Subdivision. *Journal of Motor Behavior* 35, 4 (Dec. 2003), 355–370. https://doi.org/10.1080/00222890309603156

[33] Bruno H. Repp. 2005. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review* 12, 6 (Dec. 2005), 969–992. https://doi.org/10.3758/BF03206433

[34] Gabriel Reyes, Jason Wu, Nikita Juneja, Maxim Goldshtein, W. Keith Edwards, Gregory D. Abowd, and Thad Starner. 2018. SynchroWatch: One-Handed Synchronous Smartwatch Gestures Using Correlation and Magnetic Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4 (Jan. 2018), 158:1–158:26. https://doi.org/10.1145/3161162

[35] Duk Shin, Atsushi Katayama, Kyoungsik Kim, Hiroyuki Kambara, Makoto Sato, and Yasuharu Koike. 2006. Using a Myokinetic Synthesizer to Control of Virtual Instruments. In *Proceedings of the 16th International Conference on Advances in Artificial Reality and Tele-Existence (ICAT'06)*. Springer-Verlag, Berlin, Heidelberg, 1233–1242. https://doi.org/10.1007/11941354_128

[36] Linda E. Sibert and Robert J. K. Jacob. 2000. Evaluation of Eye Gaze Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 281–288. https://doi.org/10.1145/332040.332445 event-place: The Hague, The Netherlands.

[37] Zheng Sun, Aveek Purohit, Raja Bose, and Pei Zhang. 2013. Spartacus: Spatially-aware Interaction for Mobile Devices Through Energy-efficient Audio Sensing. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. ACM, New York, NY, USA, 263–276. https://doi.org/10.1145/2462456.2464437

[38] 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 (2017), 22:1–22:35. https://doi.org/10.1145/3064937

[39] David Verweij, Augusto Esteves, Saskia Bakker, and Vassilis-Javed Khan. 2019. Designing Motion Matching for Real-World Applications: Lessons from Realistic Deployments. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '19)*. ACM, New York, NY, USA, 645–656. https://doi.org/10.1145/3294109.3295628 event-place: Tempe, Arizona, USA.

[40] 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

[41] Ju Wang, Liqiong Chang, Omid Abari, and Srinivasan Keshav. 2019. Are RFID Sensing Systems Ready for the Real World?. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '19*. ACM Press, Seoul, Republic of Korea, 366–377. https://doi.org/10.1145/3307334.3326084

[42] Andrew D. Wilson and Hrvoje Benko. 2014. CrossMotion: Fusing Device and Image Motion for User Identification, Tracking and Device Association. ACM Press, 216–223. https://doi.org/10.1145/2663204.2663270

[43] Jacob Otto Wobbrock. 2009. TapSongs: Tapping Rhythm-based Passwords on a Single Binary Sensor. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 93–96. https://doi.org/10.1145/1622176.1622194

[44] Jason Wu, Cooper Colglazier, Adhithya Ravishankar, Yuyan Duan, Yuanbo Wang, Thomas Ploetz, and Thad Starner. 2018. Seesaw: rapid one-handed synchronous gesture interface for smartwatches. ACM, 17–20. https://doi.org/10.1145/3267242.3267251

[45] Xuhai Xu, Chun Yu, Anind K. Dey, and Jennifer Mankoff. 2019. Clench Interface: Novel Biting Input Techniques. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 275:1–275:12. https://doi.org/10.1145/3290605.3300505 event-place: Glasgow, Scotland Uk.

[46] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 246–253. https://doi.org/10.1145/302979.303053 event-place: Pittsburgh, Pennsylvania, USA.

[47] Ben Zhang, Yu-Hsiang Chen, Claire Tuna, Achal Dave, Yang Li, Edward Lee, and Björn Hartmann. 2014. HOBS: Head Orientation-based Selection in Physical Spaces. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction (SUI '14)*. ACM, New York, NY, USA, 17–25. https://doi.org/10.1145/2659766.2659773

[48] 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 (ISWC '17)*. ACM, New York, NY, USA, 62–65. https://doi.org/10.1145/3123021.3123041

[49] T. Zhang, N. Becker, Y. Wang, Y. Zhou, and Y. Shi. 2017. BitID: Easily Add Battery-Free Wireless Sensors to Everyday Objects. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. 1–8. https://doi.org/10.1109/SMARTCOMP.2017.7946990

[50] Tengxiang Zhang, Xin Yi, Ruolin Wang, Yuntao Wang, Chun Yu, Yiqin Lu, and Yuanchun Shi. 2018. Tap-to-Pair: Associating Wireless Devices with Synchronous Tapping. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4 (Dec. 2018), 201:1–201:21. https://doi.org/10.1145/3287079

[51] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3025453.3025842

[52] Yang Zhang, Gierad Laput, and Chris Harrison. 2018. Vibrosight: Long-Range Vibrometry for Smart Environment Sensing. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 225–236. https://doi.org/10.1145/3242587.3242608