

Facilitating Temporal Synchronous Target Selection through User Behavior Modeling

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 input noise. 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 input noise. In the second user study, we validated that the optimal pattern sets could reduce error rate from 23% to 7% for the classical correlation-based recognizer. We also tested a novel Bayesian algorithm for input recognition, which achieved higher selection accuracy than the correlation-based algorithm with short input sequence. We also conducted an informal evaluation to explore users' selection performance when using thumb and feet to perform taps with appropriate sensing techniques. Results implied that users could still select targets effectively across modalities with different sensors.

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:

.0000. Facilitating Temporal Synchronous Target Selection through User Behavior Modeling. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 0, 0, Article 00 (0000), 19 pages. <https://doi.org/0000001.0000001>

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 two reasons: 1) cross-device interaction for a large number of devices calls for association-free target selection techniques [5, 7, 17]. 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; 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 interaction experience across different interfaces is of significance.

Aimed at these challenges, temporal synchronous target selection [5, 20, 23] 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 in multi-modality interfaces, which enables subtle and

Author's address:

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 ACM 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.

© 0000 Association for Computing Machinery.

2474-9567/0000/0-ART00 \$15.00

<https://doi.org/0000001.0000001>

00:2 •

Modality	Thumb Tap	Foot Tap	Knock	Teeth Clench	Eye blinks
Sensing Techniques	Button Capacitive Sensing	Wireless Tag [31] Vibration [34]	Audio [2, 18] Vibration [34]	EMG [3] Audio [23]	EOG [4] Eye tracking [5]

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

accessible selection experience, so that users can choose the appropriate interface when in different scenarios. For example, users can tap fingers (touchscreen [5]), clap hands (audio [16]), tap foot (vibration[34]), contract muscles (EMG [3, 24]), blink eye (EOG [4]), and even breath [11, 12] 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 easily; 3) The selection technique's low requirement of sensing resources makes it compatible with a wide range of both new and existing sensors. For example, several recent research enables low-cost and easily deployable touch interfaces [13, 31, 33], which can provide a pervasive and consistent selection experience. Users can also adapt to the current sensing infrastructure to select objects. For example, the already existing microphones in smartphones, smart speakers, and laptops[2, 18] can also be leveraged to detect synchronous sounds generated by users (e.g., hand clap, finger snap, knocking).

However, we see two major limitations in current temporal synchronous selection approaches: 1) there lacks a principle guideline for designing 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, user input noise will harm the performance of the widely used correlation recognizer when the input sequence is short, which is particularly important when there is fewer selective targets (e.g. $n < 7$).

In this paper, we improved the state-of-the-art of temporal synchronous selection techniques, and investigated the potential of such techniques through systematically modeling users' interaction behavior. We chose blinking patterns as a typical form of temporal patterns, and extracted *Period* and *Lag* as the two major features for defining the design space of patterns. In our first user study, we modeled users' finger tapping ability when syncing with different patterns, and found that both input period and delay followed a Gaussian distribution. Based on the results, we then optimized the pattern set generation process by minimizing the confusion probability among different patterns caused by noisy input. To validate the optimized pattern set, we conducted a second user study, and found users can select from up to 21 targets with an accuracy higher than 90%. Compared with using a random pattern set, the optimal 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 performs better than the classical correlation-based recognizer when the input sequence is short. Finally, we explored the performance of this selection technique in two more interfaces with different modalities (thumb and feet tap) and sensing techniques (capacitive sensing and RFID signal strength monitoring) through an informal user study. We observed that users can easily and effectively transfer their selection experience across different interfaces.

The contribution of this paper is four-folded:

- 95 (1) Theoretically, we formally explored the design space of blinking patterns in terms of *Period* and *Lag*, which
 96 complements the design rationale of temporal patterns, and introduced the Bayesian probability to quantify
 97 the similarity among patterns.
 98 (2) Empirically, we provide models of users' synchronous tapping behavior. Based on the results, we optimized
 99 the pattern set generation process by presenting an optimized pattern set for up to 22 targets that minimized
 100 the confusion caused by input noise. We also validated that the optimal pattern set can improve the selection
 101 accuracy for both Correlation and Bayesian recognizers.
 102 (3) Technically, we proposed a Bayesian recognizer which achieves higher performance than the classical
 103 Correlation recognizer when the input sequence is short and noisy.
 104 (4) In application, we implemented corresponding sensing techniques for two more modalities-thumb and feet
 105 tap. The results implied that users can transfer their temporal synchronous selection experience across
 106 modalities and select the target with different signals as inputs.

108 2 BACKGROUND AND RELATED WORK

109 Currently, users in a smart space use designated controllers or smartphones for target selection. However, the
 110 increasing number of smart devices makes it unpractical to have a controller for each device. Switching APPs is
 111 also shown to introduce significant overhead in both cognition and efficiency [19]. Thus, an association-free and
 112 spontaneous target selection technique is required for a better selection experience in smart spaces.
 113

114 2.1 Synchronous Target Selection Techniques

115 Recently, spatial synchronous target selection [25] has gained traction: users sync with the movement trace of the
 116 target on a screen to select it. Pursuit [26] and Orbit [10] show that users can select a moving target by syncing
 117 their eye gazes with it on distal displays or smart watches. The same mechanism is also shown to work for head,
 118 hand, and finger gestures [6, 8, 9, 30]. However, the eye tracking apparatus or cameras required is usually large,
 119 expensive and power demanding. The moving target can also cause visual distractions, especially when there are
 120 many targets on the screen. The spatial movements performed can also cause physical fatigue.
 121

122 Temporal synchronous selection technique is a sensorimotor synchronization task [22], which selects a target
 123 by syncing with its unique temporal patterns. RhythmLink [20] selects the target device by tapping rhythms of
 124 a designated song. SynchroWatch[23] selects a target on a smartwatch screen by syncing their thumbs with a
 125 blinking overlay on the target. Compared with spatial synchronization, temporal synchronization works with
 126 binary signals, which are typically easier for users to generate. The sensing and computing resources required is
 127 also less than those of the spatial synchronization techniques. Thus, temporal synchronous selection techniques
 128 can work on interfaces with different sizes, cost, and modalities, which enables a more consistent selection
 129 experience across different interfaces.

130 2.2 Temporal Patterns for Target Selection

131 One way to design temporal patterns is to refer to existing music pieces. TapSongs[28] and RhythmLink[20] uses
 132 rhythms from songs as the temporal sequence. The method does not require any pattern indications, since it fully
 133 relies on users' memory of song rhythms. However, there are a limited number of songs that users are familiar
 134 with. Also, different users tend to know different songs. So, design patterns based on existing song rhythms do
 135 not scale well for a large number of users and targets.
 136

137 Instead of using existing song rhythms, Ghomi et.al.[14] defined their own "rhythmic pattern" as a sequence
 138 of beats composed of three types of taps and two types of breaks. The patterns used in SEQUENCE[5] are coded
 139 as 8 bits binary sequences. Tap-to-Pair[32] uses cyclic coded binary sequences. The coded patterns can support
 140 selection from more targets. However, such patterns usually take more time for users to learn and select, which
 141

00:4 •

142 might deteriorate user experience. Also, coded patterns can have irregular breaks, which is difficult for users
143 to sync with[15]. Such pattern design methods are highly dependent on the coding schemes used and lack a
144 quantitative metric to guide the pattern set design process.

145 SynchroWatch[23] and Seesaw[29] select from two targets by using two color blocks that blink at the same
146 period but opposite phases. A wrist gesture is designed to iterate through different target pairs to select from
147 more than two targets. However, it can be physically demanding to switch pairs when there are many targets.
148 The gesture also makes it difficult to apply such pattern design and association method to other interfaces.

149 In this paper, we propose to use non-coded periodic patterns with different periods and initial lags. Compared
150 with patterns differentiated only in the phase dimension, the Period-Lag 2D design space supports selection from
151 more targets. The Bayesian probability of correct selections are calculated based on user tapping behavior models
152 on such design space, which can be used to design pattern sets with minimized confusion.

155 3 DESIGNING TEMPORAL PATTERNS

156 A core component for temporal synchronous target selection techniques is the design of patterns for each target.
157 The patterns should be distinguishable from each other and easy for users to follow. To this end, the temporal
158 characteristics of the patterns should be carefully designed and optimized. In this section, we first define the
159 design space of temporal patterns used in this paper, then introduce the optimized pattern sets generation process.
160

162 3.1 Design Space of A Single Pattern

163 Ideally, a temporal pattern with any waveform can be used as a pattern for selection. In practice, however, most
164 researchers used periodic patterns [5, 23, 32]. Compared with non-periodic patterns, this would allow the users
165 to remember the pattern and plan their syncing behavior in advance. Rather than following an unpredictable
166 pattern, learning the pattern can result in better user experience with less fatigue and shorter selection time.
167 Therefore, in this paper, we focus on designing periodic patterns. Furthermore, we assume that the pattern is
168 represented with binary data streams, so that it can be displayed on interfaces with different output capabilities.
169 We describe the periodic pattern used in this paper as “blinking pattern” henceforth.
170

171 As a binary sequence, the pattern can be either monotone or coded. One widely known example of coded
172 pattern is Morse Code. Tap-to-Pair [32] also used coded non-cyclic patterns, which is essentially pulse modulation
173 with the same frequency but different duty cycles. However, coded pattern sets lack a quantitative optimization
174 parameter. Also, they require longer observation time, which may lead to longer selection time especially when
the pattern set is large.
175

176 In this paper, we used monotone blinking patterns with two features: *Period* (T) and *Lag* (τ). *Period* is defined
177 as **half** of the repetition duration, which quantifies the frequency of the repetition; *Lag* quantifies the initial time
178 lag after a preset start time. We also define *edge* as the state changing moment in a pattern or input signal. Now,
179 the design space of blinking patterns can be described on a T- τ two dimensional plane, where each pattern can
180 be represented by a dot at coordinate (T, τ) . Figure 2a shows an example of the processed input signal and three
181 patterns in the time domain. Pattern 1 and 3 have the same period (T_1) but different initial lags with respect to
182 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.
183

184 We define the asynchrony of the input signal with the closest edge of a pattern [22] as *delay* (d) (see Figure 2a).
185 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
186 indicates that the phase of the input leads ahead of the target pattern, and positive d indicates the phase of the
187 input lags behind the target pattern.
188

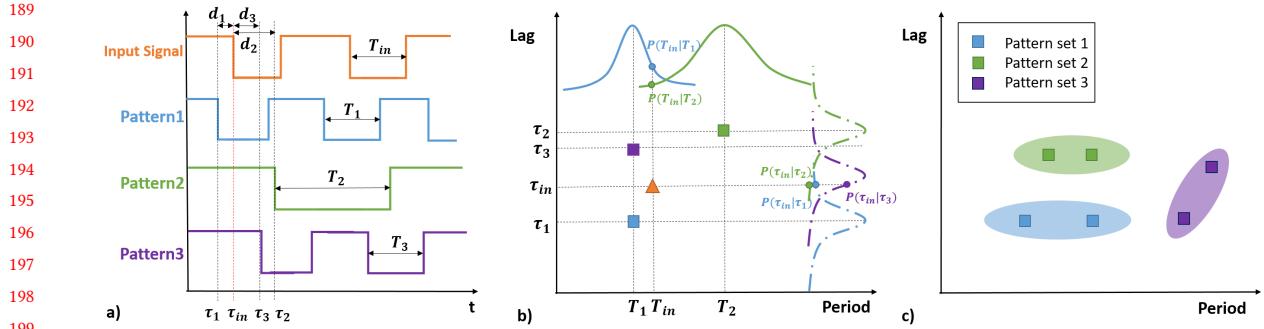


Fig. 2. T , τ , and d for the input signal and three patterns are illustrated on the time domain (a) and mapped to the T - τ plane (b). Three pattern sets are shown in (c), each with two patterns ($n = 2$).

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 pattern sets within the design spaces. 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.

To evaluate performance of an arbitrary pattern set, we calculate the confusion matrix by making each pattern as input while others as selection targets. To quantify the probability of confusion, 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. According to Bayesian theory and total probability formula, and assuming independent distributions of period and delay, the calculation is:

$$P(i|input = j) = P(T_i, \tau_i | T_{input}, \tau_{input}) \quad (1)$$

$$= P(T_i | T_{input}) \times P(\tau_i | \tau_{input}) \\ = \frac{P(T_{input} | T_i)P(T_i)}{P(T_{input})} \times \frac{P(\tau_{input} | \tau_i)P(\tau_i)}{P(\tau_{input})} \quad (2)$$

$$= \frac{P(T_{input} | T_i)P(T_i)}{\sum_{k=1}^n P(T_{input} | T_k)P(T_k)} \times \frac{P(\tau_{input} | \tau_i)P(\tau_i)}{\sum_{k=1}^n P(\tau_{input} | \tau_k)P(\tau_k)} \quad (3)$$

where n is the number of patterns in the set, $T_{input} = T_j$ and $\tau_{input} = \tau_j$. 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.

Based on the confusion metric, we then define the metric for evaluating the confusion of a whole pattern set as the estimated overall accuracy $\bar{\eta}$, given users generate "perfect" input. Therefore,

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

00:6 •

236 Intuitively, the optimal pattern set should be the one with the highest $\bar{\eta}$ value. We also set it to be the optimization
237 metric for our pattern set generation procedure.

238

239 4 STUDY 1: MODELING SYNCHRONOUS TAPPING BEHAVIOR

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

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

247

248 4.1 Participants and Apparatus

249 We recruited 12 right-handed students (9 males) from the local institution. Their ages range from 20 to 26 (Mean
250 = 22.6, SD = 2.11). Each was compensated 10 USD for their time.

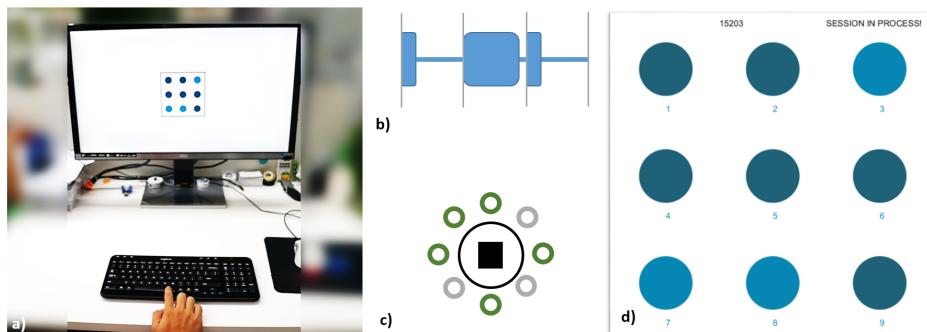
251 We used a Logitech K360 Bluetooth keyboard to collect finger tap data. The participants were asked to sit
252 at a desk, and synchronously tap on the SPACE key using their right index finger, which is found to have the
253 fastest cadence[1]. The blinking pattern was displayed on a 32-Inch monitor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

255

256 4.2 Experiment Design

257 We tested nine levels of periods within 300-700ms: 300ms, 350ms, 400ms, 450ms, 500ms, 550ms, 600ms, 650ms, and
258 700ms. The 300ms period is already close to human finger synchronization limits[21]. And according to our pilot
259 study, periods longer than 700ms could be too slow and lead to long selection time. Nine circles (3cm diameter on
260 screen) were arranged in a 3x3 layout on the monitor (see Figure 3a). Note that due to the time-homogeneity of
261 users' tapping behavior, τ would not affect users' tapping behavior in terms of T or d . Therefore, we set $\tau \equiv 0$
262 during this study. Distribution models for other τ values can be obtained by shifting the model here along the
263 time axis.

264



277 Fig. 3. a) Study 1 experiment setup; b-c) Pattern displays from previous research; d) Selection interface used in this study.
278

279

280 There are many ways to visualize the blinking patterns. For example, Ghomi et.al. visualize the patterns as
281 clipped sound waves[14] (Figure 3b), while SEQUENCE uses eight circles around the target[5] (Figure 3c). Users
282

283 can see the entire repetition of the pattern, which makes the selection process less mentally demanding. However,
 284 such visualizations occupy large amount of space on the screen and is not suitable for non-screen displays (e.g.
 285 LEDs). Alternatively, we used blinking grey translucent overlays on each circle to display the patterns (Figure 3d)
 286 in this study, which was proposed in [23]. Such pattern display mechanism can be scaled to both large screens
 287 and low-power LEDs, and is intuitive for users to sync with. The display arrangement also allows us to observe
 288 users' synchronization performance under visual distraction from other blinking targets.

289

290 4.3 Procedure

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

298

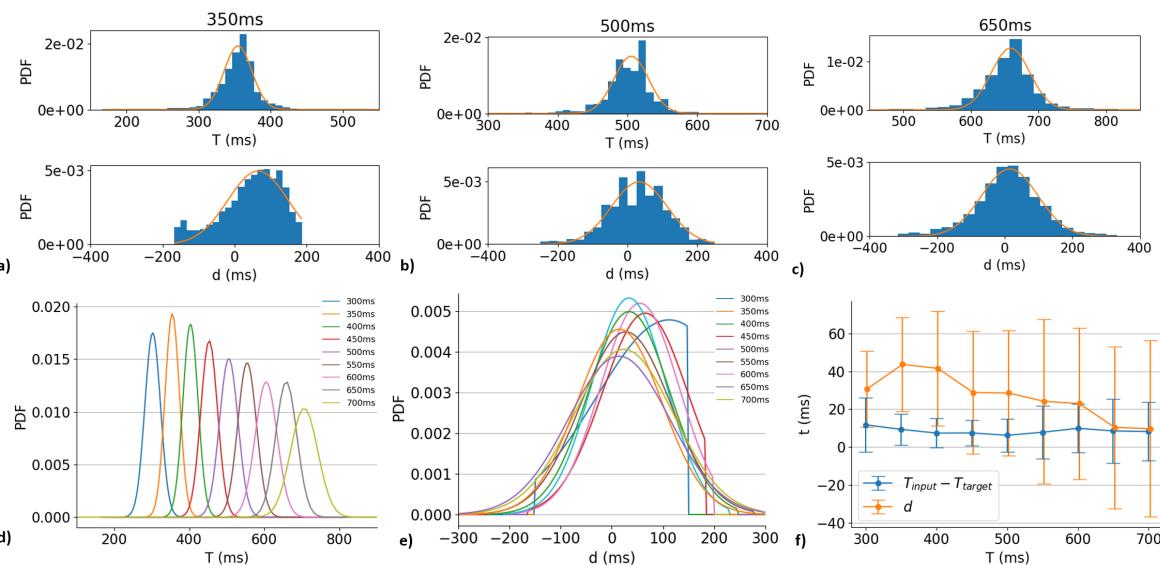
299 4.4 Results

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

303

304

305



324

325 Fig. 4. The histograms of measured distribution and the fitted Gaussian models of periods T and input delay d when the
 326 target blinks at 350ms (a), 500ms (b) and 650ms (c) are shown for illustration. Fitted Gaussian models of the measured tapping
 327 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
 328 mean difference between measured input and the target pattern is shown in (f), error bar indicate one standard deviation.

329

As expected, for each of the target periods, the measured T and d both roughly follow a univariate Gaussian distribution. For illustration, Figure 4a-4c showed the histograms of measured T and d for 350ms, 500ms and 650ms respectively. 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 find a consistent trend. RM-ANOVA confirmed that target period has significant effects on input delays ($F_{8,88} = 3.87, p < .001$), but not on input period error ($F_{8,88} = 0.57, p = .80$).

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[22], which we believe will not significantly affect our results.

4.5 Generating Optimal Pattern Sets

To determine the optimal 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).

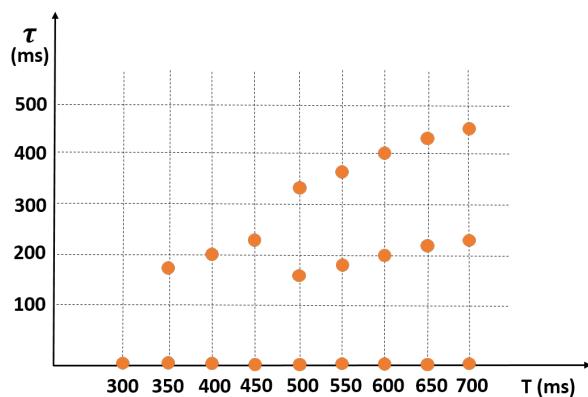


Fig. 5. Candidate patterns on the $T - \tau$ plane.

As described in previous sections, among all possible combinations of the patterns, the optimal 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 optimal 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.

377 The results are shown in Table 1. In total, we selected 21 pattern sets from 4,194,281 possible pattern sets, with
 378 sizes $2 \leq n \leq 22$. In general, the optimal pattern set with size n is the superset of the pattern set with size $n - 1$.
 379 This facilitates users' selection behavior transfer for different numbers of targets. By looking into the composition
 380 of the optimal pattern sets, we found that when expanding a pattern set, new patterns with period different than
 381 existing patterns are more likely to be selected. As n further increases (e.g. > 9), patterns with the same period
 382 but different τ are added, starting from longer periods to shorter periods.
 383

384 Table 1. Optimized pattern sets with sizes $2 \leq n \leq 22$ and the corresponding optimized window size and threshold for
 385 Bayesian (win_{baye}, TH_{baye}) and correlation-based recognizers (win_R, TH_R).
 386

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

410

411

412 5 INPUT RECOGNIZER IMPLEMENTATION

413 So far, we have optimized the pattern set design process of temporal synchronous selection techniques. In this
 414 section, we describe our implementation of the input recognizer, which is also important for target selection
 415 performance. Specifically, we implemented two kinds of recognizers: correlation-based recognizer and Bayesian
 416 recognizer. Correlation-based recognizer (referred to as *Correlation recognizer* henceforth) is the most widely-
 417 used recognizer in state-of-the-art research of spatial [25] and temporal [23, 32] synchronous object selection
 418 techniques. It works by calculating the Pearson correlation coefficients between the signal induced by user
 419 movement and the displayed blinking pattern within a time *window*. An object is selected when the coefficient
 420 between the signal and its corresponding blinking pattern is greater than a specific *threshold*.

421 However, due to input noise, the performance of correlation recognizer may degrade when the input signal-
 422 to-noise ratio is low. This usually happens for short input sequence, during which users tend to make more
 423

00:10 •

424 input errors (noises) while trying to “find the rhythm”. Accordingly, the user behavior model in Study 1 can
425 potentially improve this situation by calculating the Bayesian probability. Therefore, we also implemented a
426 Bayesian recognizer.

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

431

$$P(i|input) = P(T_i, \tau_i | T_{input}, \tau_{input}) \quad (5)$$

433

$$= P(T_i | T_{input}) \times P(\tau_i | \tau_{input}) \\ = \frac{P(T_{input} | T_i)P(T_i)}{P(T_{input})} \times \frac{P(\tau_{input} | \tau_i)P(\tau_i)}{P(\tau_{input})} \quad (6)$$

436

$$\propto P(T_{input} | T_i)P(T_i) \times P(\tau_{input} | \tau_i)P(\tau_i) \quad (7)$$

438

439 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
440 using the Gaussian model from Study 1 (see Figure 4).

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

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

451

452 5.1 Recognizer Parameter Optimization

453 As described above, the values of window size (*win*) and threshold (*TH*) need to be optimized so that the recognizer
454 can achieve high accuracy and low selection time. A small *win* or a low *TH* would cause the recognizer to be too
455 sensitive, while a big *win* or a high *TH* will lead to longer selection time. To determine the optimal value of *win*
456 and *TH*, for each n , we ran simulations for both recognizers using all the data collected in Study 1. We tested
457 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
458 process is:

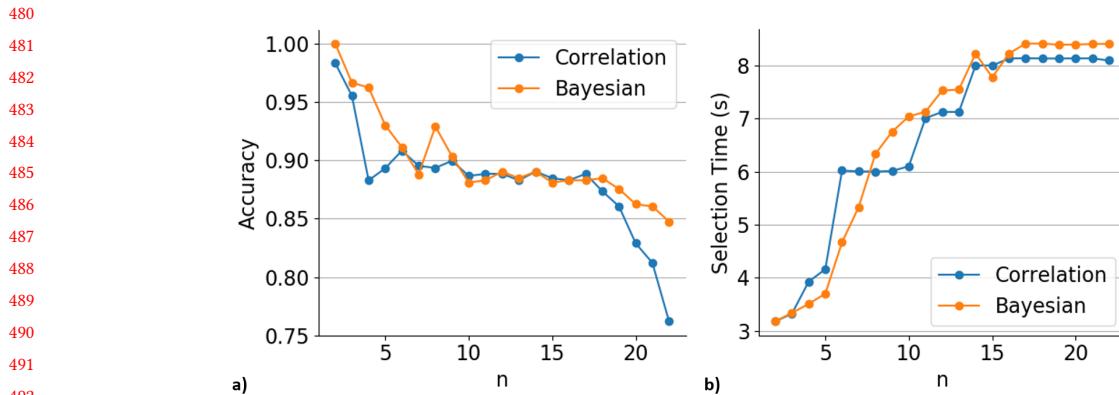
459

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

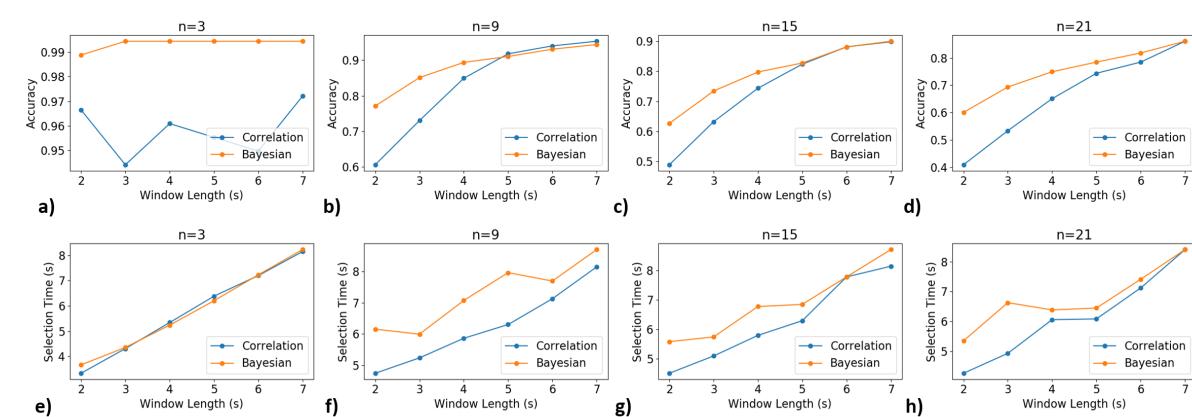
470

471 Table 1 showed the results of the optimal recognizer parameters for each n . As expected, the optimal win value
 472 both increased with n for correlation recognizer and Bayesian recognizer. On the other hand, optimal TH value
 473 did not yield a consistent trend with n for either recognizer.

474 Figure 6 showed the simulated accuracy and selection time with optimal win and TH for each n . Generally,
 475 Bayesian Recognizer showed advantage over Correlation recognizer in terms of both accuracy and selection time
 476 for small number of targets ($n \leq 5$). The performance of both recognizers became similar when n increased. Note
 477 that optimal win value for small n is very small (see Table 1), this complies with our assumption that Bayesian
 478 recognizer is more robust to scenarios with low input signal-to-noise ratio (e.g., data sequence is short).
 479



490 Fig. 6. The simulated accuracy (a) and selection time (b) with optimized win and TH for each n using data collected in Study
 491 1.



513 Fig. 7. Calculated upper bound of accuracy (a-d) and the corresponding selection time (e-h) of different window lengths for
 514 $n = 3, 9, 15, 21$ by running Correlation (blue) and Bayesian (orange) recognizers on the optimized pattern sets.

518 5.2 Accuracy Upper Bound

519 To look deeper into the characteristics of correlation recognizer and Bayesian recognizer, we compared the upper
 520 bound of the accuracy of both recognizers for different level of n . The upper bound accuracy was calculated by
 521 fixing win , then chose the highest accuracy yielded by all TH values. Figure 7 showed the result for $n = 3, 5, 9, 12$,
 522 with the corresponding mean selection time. Similar with previous results (see Figure 6), for $n = 3$, the Bayesian
 523 recognizer achieves higher accuracy than the correlation recognizer, while keeping competitive selection time.
 524 For $n \geq 9$, the Bayesian recognizer still seems to yield higher accuracy than the correlation recognizer, especially
 525 when win was small (i.e., data sequence is short). However, the selection time of Bayesian recognizer was also
 526 longer than the correlation recognizer, suggesting a speed-accuracy tradeoff. The performance of Correlation
 527 recognizer improves as the input sequence gets longer and the input signal-to-noise ratio increases.
 528

529 6 STUDY2: EVALUATING OPTIMAL PATTERN SETS AND RECOGNIZERS

530 So far, we have improved the two key components of temporal synchronous selection technique: pattern set
 531 design and recognizer. In this section, we conducted another user study to verify the performances of the proposed
 532 pattern sets and recognizers. We designed the tasks to mimic a realistic smart TV scenario, where participants
 533 were asked to tap a single key to select from a list of displayed movie posters. We tested two recognizers (Bayesian
 534 and Correlation), two types of pattern sets (optimized, period-increasing), and four sizes of pattern sets (3, 9, 15,
 535 21).

536 6.1 Participants and Apparatus

537 We recruited 14 right-handed participants (9 males) from the local institution that did not take part in the previous
 538 study. Their ages range from 19 to 30 (Mean = 22.9, SD = 2.95). Each participant was compensated 15 USD. We
 539 used the same 32-inch monitor as in Study 1 to display the experiment interface. A Microsoft Designer Bluetooth
 540 Keyboard was used to detect tapping input. The selection interface was implemented using Python Tkinter.
 541

542 6.2 Experiment Design and Procedure

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

553 Upon arrival, the participant first completed a demographic questionnaire. We then explained the task and
 554 asked the participant to warm up for two minutes. The participant then sat 1.5m away from the monitor, and
 555 completed 8 sessions of target selection tasks. In each session, they complete 16 blocks of tasks, each corresponds
 556 to a $\{pattern\ set, recognizer, n\}$ combination. The order of the combinations were randomized. In each block, the
 557 participant were asked to select the target poster by pressing the SPACE key on the keyboard in sync with the
 558 corresponding blinking square. Each block is finished after 15 seconds or when a poster (whether correct or not)
 559 is selected. The participant then pressed ENTER to enter the next block. A 30-second and 1-minute break was
 560 enforced in the middle of a session and between sessions, respectively. At the end of the study, we conducted
 561

562

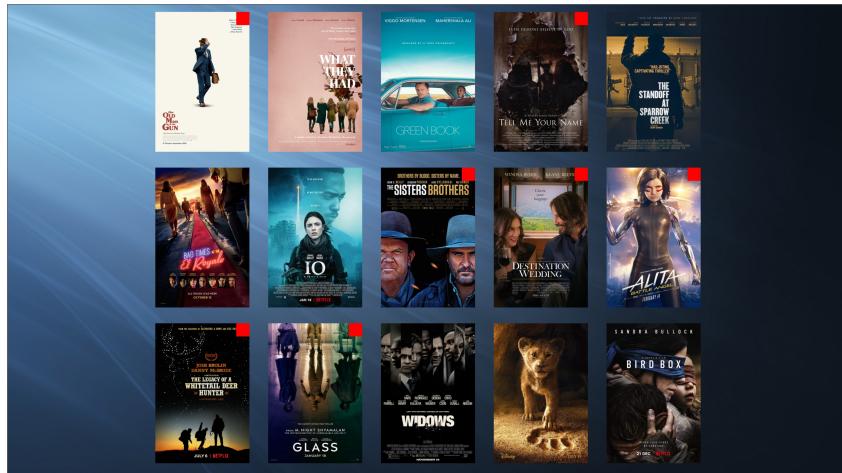


Fig. 8. Selection interface for $n = 15$. The target poster is the first poster in the first row.

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 $4n \times 2 \text{ recognizers} \times 2 \text{ pattern sets} \times 8 \text{ sessions} \times 14 \text{ participants} = 1,792$ selection tasks were conducted. We ran RM-ANOVA on parametric data and tested them against a 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 on period-increasing patterns (Bayesian ($F_{1,13} = 11.4, p < .01$), Correlation ($F_{1,13} = 41.6, p < .001$)). The Bayesian recognizer yield similar performance as the correlation recognizer on the optimized pattern sets (93.8% vs. 93.3%). However, on non-optimal pattern sets, Bayesian recognizer achieved higher accuracy than the correlation recognizer (85.9% vs. 77.2%). Note that on non-optimal pattern sets, input signal for different targets are more alike, yielding a lower relative signal-to-noise level. Therefore, this again confirmed the advantage of Bayesian recognizer for predicting noisy input.

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 ($F_{3,39} = 0.77, p = .52$) or recognizer ($F_{1,13} = 0.04, p = .84$) 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 input noises in short input sequence (only a few taps), while Correlation recognizer is more vulnerable to the input noises. In terms of selection time, no significant effect of recognizer ($F_{1,13} = 0.38, p = .54$) was found (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 Figure 10, the overall ratings showed that all participants were

00:14 •

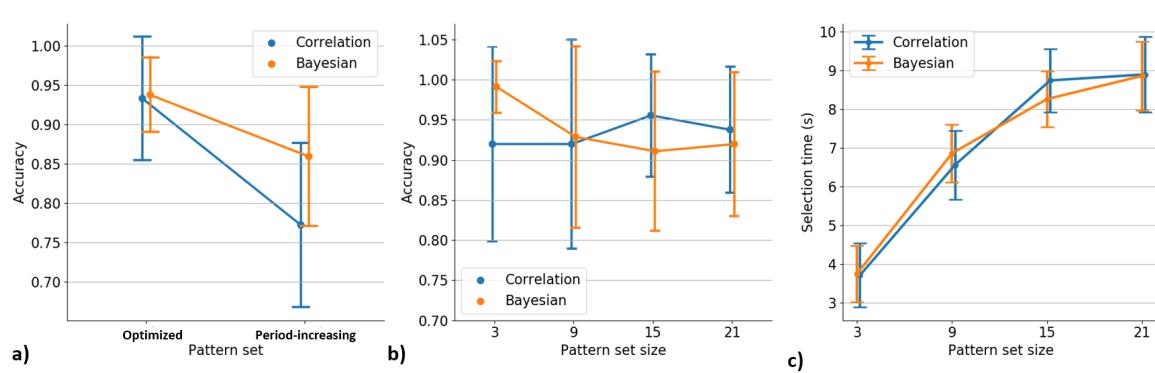


Fig. 9. (a) Accuracy of different recognizers on different pattern sets, merged different n ; (b) accuracy of different recognizers on different n and optimal pattern set; (c) selection time of different recognizers on different n and optimal pattern set. Error bar indicate one standard deviation.

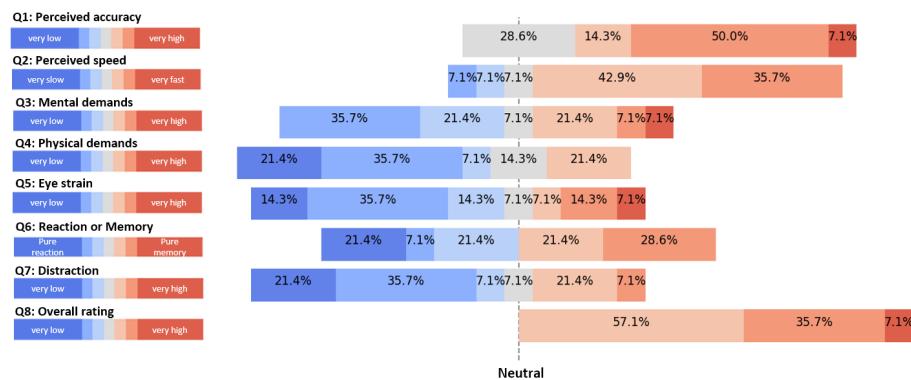


Fig. 10. Subjective ratings of the selection technique.

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 STUDY3: EXPLORING MORE MODALITIES

One of the major advantages of temporal synchronization selection technique is the easiness to be applied to different interfaces with different modalities and sensing techniques. To verify this, we conducted an informal study that involves two more modalities using different sensing techniques: 1. *Thumb Tap*: Thumb taps on a ring

worn on the index finger; 2. *Foot Tap*: Foot taps on the floor. The modalities cover use cases when one hand or both hands are busy, and provide low-effort, subtle, and accessible object selection experience.

661 662 7.1 Participants, Apparatus and Procedure

663 We recruited three male participants with ages 23, 23, and 24 respectively from the local institution. They all
664 have participated in Study 2. Each participant was compensated 10 USD. We used the same display monitor and
665 experiment interface as in Study 2. The only difference is the sensing technique:

666 **Thumb Tap** We built a capacitive sensing ring by placing an off-the-shelf capacitive sensing module into a
667 3D printed case. The module outputs high voltage only when being touched. The analog output signal was
668 monitored by an Arduino UNO and reported to a PC via a serial port. Users can then tap their thumbs on
669 the ring to select objects.

670 **Foot Tap** We placed an Alien AZ-9654 Ultra High Frequency (UHF) RFID tag on the floor carpet, and
671 monitored its Received Signal Strength Indicator (RSSI) using an Impinj R420 reader. The reader then
672 reported the RSSI of the tag to PC via TCP connection. Rhythmic foot taps can introduce periodic signal
673 changes due to the signal blockages and antenna mismatch [32], which can then be used for temporal
674 synchronization based object selection.

675 We used the Correlation recognizer in this study, as it does not require model training, and can achieve similar
676 performance as the Bayesian recognizer for great n . And we used the optimal pattern set with $n = 3, 5, 9, 21$ to
677 better compare the results with those from Study 2. For each modality, the participants were asked to complete 5
678 sessions of selection task (order balanced by Latin Square), with 12 selection tasks in each session, three for each
679 n . The participants shared their experience of using different modalities after the experiment. The experiment
680 took about 30 minutes to complete for each participant.



693 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
694 selection accuracy for different modalities with respect to target pattern period using Correlation recognizer is also shown
695 (c). We added the index data from Study 2 for comparison

696 697 7.2 Results and Analysis

698 The overall accuracy across different T was 91% for thumb tap and 81% for foot tap. Compared with results in
699 Study 2 (see Figure 9b), the index finger and thumb finger tapping yield similar selection accuracies, both higher
700 than foot tapping. We found that selection error in foot tap mainly happened on patterns with small periods
701 (i.e. blinks fast) (Figure 11c). In general, the foot tapping accuracy becomes close to those of finger tapping for
702 larger periods with an outlier at $T = 500\text{ms}$. We found that 80% of the selection errors at $T = 500\text{ms}$ period are
703 incorrect recognition on the τ dimension (recognized as the pattern with correct T but incorrect τ), which might
704 be caused by the low signal-to-noise ratio of the UHF RFID tag. The index finger tapping accuracy is higher than
705 the foot tapping accuracy, which is consistent with previous studies [32].

⁷⁰⁶ due to the fact that patterns with $T = 500ms$ has the smallest separation on the τ dimension (167ms). The results
⁷⁰⁷ implied that the foot has a worse synchronization ability in both T and τ dimensions than the index and thumb
⁷⁰⁸ finger. The observation was also confirmed in the interview, when all three participants mentioned that the fast
⁷⁰⁹ patterns are too fast to follow by tapping foot.

⁷¹⁰ 8 DISCUSSION

⁷¹¹ 8.1 Correlation Recognizer vs Bayesian Recognizer

⁷¹² Aside from the performance difference observed in Study 2, the Correlation and Bayesian recognizer are different
⁷¹³ in many other aspects.

⁷¹⁴ **Signal Type** Correlation recognizer requires signals similar to Unipolar Non-Return-to-Zero (NRZ) code ¹,
⁷¹⁵ which maps to current sensing *status*. For example, the RFID RSSI and capacitive sensing signal in Study 3
⁷¹⁶ are both NRZ signals. Bayesian recognizer, however, requires signals similar to Unipolar Return-to-Zero
⁷¹⁷ (RZ) code ², which maps to sensing status *changes*. One example is the audio signal pulses generated by
⁷¹⁸ clapping hands periodically. The two types of signals can be converted to each other during preprocessing.
⁷¹⁹ Edge or peak detection is usually used to convert NRZ signals to RZ signals, while the RZ signals can be
⁷²⁰ easily converted to NRZ signals by switching state at each pulse.

⁷²¹ **Data Requirements** Bayesian recognizer can only be used with behavior models, while it is not necessary
⁷²² to collect any data in advance when using Correlation recognizer.

⁷²³ **Resource Requirements** The Bayesian recognizer demands higher computational power and more storage
⁷²⁴ capacity for the behavior models. The computation and storage resources demands for Correlation recog-
⁷²⁵ nizer is lower, which makes it especially suitable for embedded real-time system with limited resources [32].

⁷²⁶ **Robustness to Input Noise** Bayesian recognizer is more robust to input noise, especially when the input
⁷²⁷ sequence is short. Correlation recognizer is vulnerable to input noises with short input sequence. For longer
⁷²⁸ input sequence (larger window size), the signal-to-noise ratio converges and the two recognizers have
⁷²⁹ similar performances.

⁷³⁰ In general, Bayesian recognizer performs better for single modality selection tasks with few targets (e.g. $n < 5$),
⁷³¹ while Correlation recognizer is better suited for multi-modality selection tasks with more selective targets.

⁷³² 8.2 Multi-modality Selection and Sensor Fusion

⁷³³ The results of Study3 imply that the optimal patterns for different modalities can be different. However, it is
⁷³⁴ difficult to understand the motor synchronization abilities of all human body parts. One solution is to use a sensor
⁷³⁵ hub to process signals collected by different types of sensors (e.g., video, audio, touch) so that users can choose
⁷³⁶ the modality that provides the most *confident* and *comfortable* selection experience. For example, P2 in Study 3
⁷³⁷ mentioned that he felt “more certain” when using foot tapping for slow blink patterns and finger tapping for fast
⁷³⁸ blink patterns. This suggests that users can use foot to sync with slower patterns and fingers to sync with faster
⁷³⁹ patterns.

⁷⁴⁰ For multi-modality selection, a recognizer can be implemented on the sensor hub to match all currently active
⁷⁴¹ patterns with different signal streams, and select the target when the matching result for any signal is higher than
⁷⁴² the threshold. Signals from different sensors can also be cross-checked to improve recognition performance[27].
⁷⁴³ For example, knocking on a table can generate periodic sounds and vibrations at the same time, which can both
⁷⁴⁴ be sensed by a smart phone placed on the table. The audio and IMU data collected on the phone can be analyzed
⁷⁴⁵ together to improve the recognition accuracy.

⁷⁴⁶ ¹<https://en.wikipedia.org/wiki/Non-return-to-zero>

⁷⁴⁷ ²<https://en.wikipedia.org/wiki/Return-to-zero>

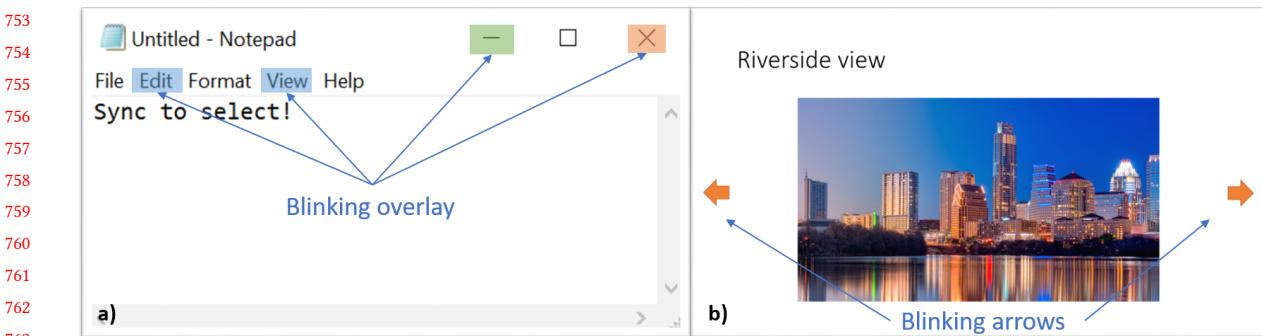


Fig. 12. Blinking half-transparent overlays in Notepad (a) and blinking arrows on slides for navigation (b).

8.3 Pattern Display and Feedback

For screen displayed objects, blinking UI elements can be automatically generated and associated with the target in different forms. For example, patterns can be half-transparent color blocks overlaid on the selection target for less blockage. Different colors be assigned to commands with different hierarchies. The patterns can also be blinking shapes that are suitable for the current context (see Figure 12). For example, when in the slides presentation mode, blinking arrows can be automatically generated to navigate through the slides. It is also possible to adjust the pattern sets based on the usage frequencies of different objects to achieve a more accurate and faster selection experience.

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[29] of the items can change according to the correlation value. A progress bar can also show the selection probability for each on-screen object. Even though the visual feedback can help improve the selection performance, it will also introduce extra visual distractions. We leave the feedback design for future research.

9 LIMITATIONS AND FUTURE WORK

Both Study 2 and Study 3 were done in a controlled lab environment. There could be more noise in a more realistic setting, which will impact the selection performance. On one hand, the noises can be easily filtered out using a band-pass filter thanks to the proposed design space; on the other hand, users can choose the modality that is most appropriate in the current environment. For example, users can choose to tap fingers instead of clap hands when there is loud background noise. In the future, we plan to conduct a field study to evaluate the selection technique's performance in the wild.

The pattern set generation algorithm and the Bayesian recognizer we proposed require user behavior models. The results of Study 2 show that the model can be generalized between users. The results of Study 3 imply that the patterns designed for index finger tapping are better suited for thumb tapping than to foot tapping though. We plan to explore methods to improve the generalizability of the user synchronous models in the future.

Selection time greatly impacts the user experience. On one hand, we believe more practices can reduce the time spent on pattern observation. On the other hand, user input behavior can be collected during usage to build a personalized model, which could help reduce the selection time.

800 10 CONCLUSION

801 In this paper, we proposed an approach to design patterns for temporal synchronous target selection techniques.
 802 We defined the design space on a Period-Lag 2D plane, and conducted a study to model synchronous finger
 803 tapping behaviors. With the model, we used Bayesian algorithm to generate pattern sets that minimize the
 804 selection confusion. We ran simulations and found that Bayesian recognizer out-performed the classical correlation
 805 recognizer for small size pattern sets. The evaluation results showed that the selection accuracy with the optimized
 806 pattern sets are higher than the period-increasing pattern sets by 16% using Correlation recognizer and 8% using
 807 Bayesian recognizer. The results of the informal multi-modality evaluation using different sensing techniques
 808 implied that users can effectively transfer their selection experience across modalities enabled by different sensors.
 809 We believe user modeling can facilitate the application of temporal synchronous target selection techniques on
 810 more interfaces as an association-free multi-modal object selection technique.

812 REFERENCES

- 813 [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>
- 814 [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>
- 815 [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*. ACM, New York, NY, USA, 158–169. <https://doi.org/10.1145/2935334.2935389>
- 816 [4] 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>
- 817 [5] 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>
- 818 [6] 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*.
 ACM, New York, NY, USA, 3415–3427. <https://doi.org/10.1145/2858036.2858284>
- 819 [7] 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>
- 820 [8] 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*. ACM, New York, NY, USA, 298–303. <https://doi.org/10.1145/2971648.2971714>
- 821 [9] 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*. ACM, New York, NY, USA, 179–192.
<https://doi.org/10.1145/3126594.3126626>
- 822 [10] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbita: Gaze Interaction for Smart Watches Using Smooth
 Pursuit Eye Movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, New York,
 NY, USA, 457–466. <https://doi.org/10.1145/2807442.2807499>
- 823 [11] 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*. ACM, New York, NY, USA, 157–160. <https://doi.org/10.1145/2371664.2371697>
- 824 [12] 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*.
 ACM, New York, NY, USA, 217–218. <https://doi.org/10.1145/2384916.2384961>
- 825 [13] Chuhuan Gao, Yilong Li, and Xinyu Zhang. 2018. LiveTag: Sensing Human-Object Interaction through Passive Chipless WiFi Tags.
<https://doi.org/10.1145/2207676.2208579>
- 826 [14] 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*. ACM, New York, NY, USA, 1253–1262.
<https://doi.org/10.1145/2207676.2208579>

- [15] 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>
- [16] 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*. ACM, New York, NY, USA, 3175–3180. <https://doi.org/10.1145/1520340.1520452>
- [17] 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>
- [18] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicouistics: Plug-and-Play Acoustic Activity Recognition. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 213–224. <https://doi.org/10.1145/3242587.3242609>
- [19] 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*. ACM, New York, NY, USA, 291–294. <https://doi.org/10.1145/2371574.2371617>
- [20] 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*. ACM, New York, NY, USA, 263–272. <https://doi.org/10.1145/2047196.2047231>
- [21] 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>
- [22] 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>
- [23] Gabriel Reyes, Jason Wu, Nikita Juneja, Maxim Goldshteyn, 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>
- [24] Duk Shin, Atsushi Katayama, Kyungsik 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*. Springer-Verlag, Berlin, Heidelberg, 1233–1242. https://doi.org/10.1007/11941354_128
- [25] Eduardo Veloso, 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>
- [26] 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*. ACM, New York, NY, USA, 439–448. <https://doi.org/10.1145/2493432.2493477>
- [27] 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>
- [28] 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*. ACM, New York, NY, USA, 93–96. <https://doi.org/10.1145/1622176.1622194>
- [29] 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. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM, 17–20. <https://doi.org/10.1145/3267242.3267251>
- [30] 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*. ACM, New York, NY, USA, 62–65. <https://doi.org/10.1145/3123021.3123041>
- [31] 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>
- [32] 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>
- [33] 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*. ACM, New York, NY, USA, 1–14. <https://doi.org/10.1145/3025453.3025842>
- [34] 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*. ACM, New York, NY, USA, 225–236. <https://doi.org/10.1145/3242587.3242608>