Tap-to-Pair: Associating Wireless Devices with Synchronous Tapping

TENGXIANG ZHANG, XIN YI*, RUOLIN WANG, YUNTAO WANG, CHUN YU, YIQIN LU, and YUANCHUN SHI, Tsinghua University, China

Ad-hoc wireless device pairing enables impromptu interactions in smart spaces, such as resource sharing and remote control. The pairing experience is mainly determined by the device association process, during which users express their pairing intentions between the advertising device and the scanning device. Currently, most wireless devices are associated by selecting the advertiser's name from a list displayed on the scanner's screen, which becomes less efficient and often misplaced as the number of wireless devices increases. In this paper, we propose Tap-to-Pair, a spontaneous device association mechanism that initiates pairing from advertising devices without hardware or firmware modifications. Tapping an area near the advertising device's antenna can change its signal strength. Users can then associate two devices by synchronizing taps on the advertising device with the blinking pattern displayed by the scanning device. By leveraging the wireless transceiver for sensing, Tap-to-Pair does not require additional resources from advertising devices and needs only a binary display (e.g. LED) on scanning devices. We conducted a user study to test users' synchronous tapping ability and demonstrated that Tap-to-Pair can reliably detect users' taps. We ran simulations to optimize parameters for the synchronization recognition algorithm and provide pattern design guidelines. We used a second user study to evaluate the on-chip performance of Tap-to-Pair. The results show that Tap-to-Pair can achieve an overall successful pairing rate of 93.7% with three scanning devices at different distances.

CCS Concepts: • **Human-centered computing → Interaction techniques**; **Ubiquitous and mobile computing systems and tools**;

Additional Key Words and Phrases: wireless device association, synchronization, temporal correlation, pattern design, Bluetooth

ACM Reference Format:
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, Article 201 (December 2018), 21 pages. https://doi.org/10.1145/3287079

1 INTRODUCTION

With the rise of the Internet of Things (IoT), there are more wireless devices in our daily lives than ever before. Ad-hoc pairing between such devices can enable more interaction possibilities in smart spaces. For example, by spontaneously pairing a smart ring with different computing devices, users can have an easier and more consistent input experience. Wireless devices can be categorized into two types: *scanning devices* (e.g. smartphones, laptops) and *advertising devices* (e.g. Bluetooth speakers, WiFi routers) (Figure 1). Advertising devices (advertisers)

*This is the corresponding author

Authors' address: Tengxiang Zhang, ztx16@mails.tsinghua.edu.cn; Xin Yi, yixin@mail.tsinghua.edu.cn; Ruolin Wang; Yuntao Wang; Chun Yu; Yiqin Lu; Yuanchun Shi, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Global Innovation eXchange Institute, Tsinghua University, 30 Shuangqing Road, Beijing, 100084, China.

2474-9567/2018/12-ART201 $15.00
https://doi.org/10.1145/3287079

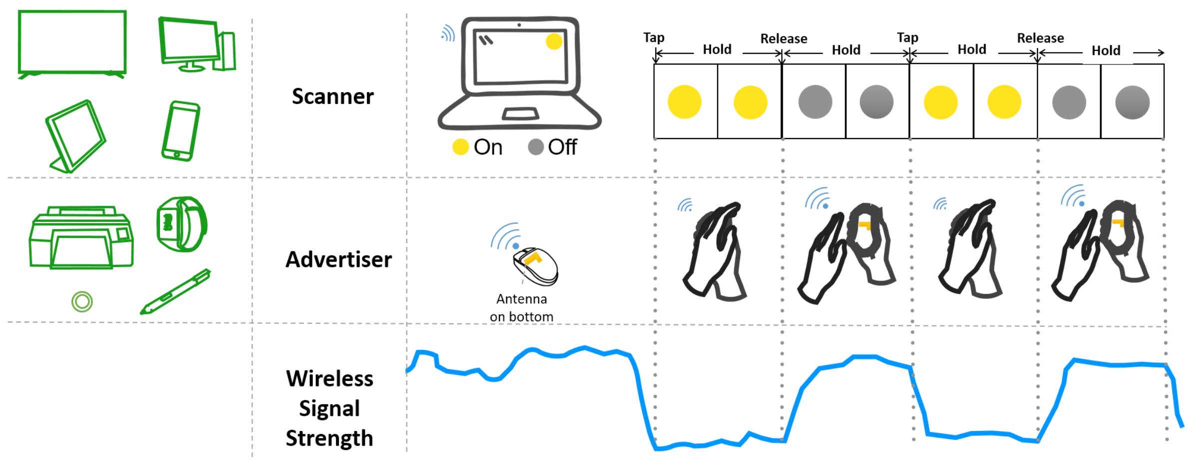

Fig. 1. Initiate pairing from a wireless mouse with a laptop using Tap-to-Pair. Top row: example scanners, and a pattern displayed on a laptop and the mapped behaviors; Middle row: example advertisers, and user taps on a mouse in synchronization with the pattern; Bottom row: The mouse's advertised wireless signal strength received by the laptop.

advertise their identities over the air, while scanning devices (scanners) scan for such advertisements and initiate pairing by sending connection requests to advertisers. Before scanners send the requests, however, users must manually associate the two devices (referred to as *device association* [7]), which directly impacts the pairing experience. Currently, the most popular device association mechanism is selecting the advertiser's name from a list on the scanner's screen.

Unfortunately, it is difficult to locate the target advertiser as the list becomes exceedingly crowded in environments with a large number of wireless devices. Moreover, this current method only allows users to initiate association from scanners, while most prefer to use small and light devices whenever possible [7], which are usually the advertisers. It is also more convenient to initiate association from advertisers that are constantly paired with different scanners. For example, when switching a Bluetooth keyboard between PCs, it is easier to initiate the association from the keyboard than from each PC separately. The increasing number of wireless devices can make it difficult for users to find the target advertiser buried in a long list. Previous research has enabled association from advertisers by using special hardware like Infrared (IR) transceivers [14, 35, 40]. However, such methods are not suitable for advertisers that are small, cost sensitive, or power constrained (e.g. mouse, stylus). Thus, it is important to have a scheme that can enable spontaneous device association from advertisers with minimal resource requirements.

In this paper, we propose *Tap-to-Pair*, a spontaneous device association mechanism that enables users to initiate association using off-the-shelf advertising devices without hardware or firmware modifications. Tapping an area near the advertiser's antenna creates a change in its transmitted signal strength, which can then be detected by the scanner. This allows users to initiate association by matching the rhythm of their taps on the advertiser with a blinking pattern (e.g. LED) displayed on the scanner. For example, the user can tap on a mouse to create signal drops in synchronization with a blinking dot on a laptop's screen (Figure 1). The scanner first acts as the recognizer to record the correlation between the two signals, then sends a connection request to the corresponding advertiser.

Tap-to-Pair has two major advantages: 1) The resource requirements of implementing Tap-to-Pair are minimal. It leverages the built-in wireless transmitter on advertisers to create signal changes. On scanners, Tap-to-Pair only requires a binary display and a recognizer. The almost ubiquitous LEDs present on wireless devices can act as binary displays. Meanwhile, the lightweight computation required by Tap-to-Pair allows the recognizer to be implemented on the scanner's preexisting wireless receiver. 2) Instead of associating from scanners with advanced interfaces, Tap-to-Pair enables users to initiate association from advertisers with various size, cost, and power constraints, including smart rings, universal remotes, smart tables, and even smart papers.

We built our prototype using Bluetooth devices since they are a widely deployed peer-to-peer protocol that supports impromptu connection and provides Received Signal Strength Indicator (RSSI) to measure signal amplitude changes. We first conducted a user study to test users' synchronous tapping performance for different blinking patterns.The results also showed that Tap-to-Pair and Inertial Measurement Unit (IMU) have similar tap synchronization sensing performance. We then ran a simulation using the collected data to determine the optimal parameters for our recognition algorithm. In the final evaluation study, Tap-to-Pair implemented on-chip was able to associate a Bluetooth mouse and a keyboard with three scanners, with a successful pairing rate of 93.7%. Finally, we offer four potential applications of Tap-to-Pair and discuss its scalability, design considerations, and limitations. Our contributions are threefold:

(1) We propose a novel device association mechanism that can initiate pairing from advertising devices by synchronous tapping, with no hardware or firmware modifications required on the advertiser.
(2) We conducted a user study to demonstrate Tap-to-Pair's sensing rationale and test users' synchronous tapping ability. We also provide suggestions for pattern designs based the results.
(3) We implemented Tap-to-Pair's algorithm on chip, validated its usability in terms of pairing time, false pairing rate and user satisfaction, and illustrate several applications to demonstrate its potential.

2 BACKGROUND AND RELATED WORKS

Tap-to-Pair is a temporal correlation based device association technique, which is inspired by previous works that leverage rhythmic input and correlated signals for device association and object selection. In this section, we first look at rhythmic-based input techniques. Then we explain Tap-to-Pair's differences with existing correlation-based techniques, and review other cross-device association mechanisms that are closely related to Tap-to-Pair. Finally, we highlight the differences between Tap-to-Pair and the aforementioned association techniques in terms of hardware and computing resource requirements in Table 1.

2.1 Rhythmic Input

Rhythmic input is related to Tap-to-Pair in that they both involve rhythmic tapping. Users can select a menu item by clicking a mouse when it is highlighted following a rhythmic pattern[21]. They can also use rhythmic input for text entry [33]. Rhythmic input is especially suitable on I/O resource constrained devices. For example, users can authenticate on an earphone controller [39], interact with small screens like smartwatches [24], and play games[41] using rhythmic tapping. RhythmLink shows that users can associate smart devices with a smartphone by tapping a piece of the song rhythm [18].

However, the rhythmic input based device association mechanisms assume that users have prior knowledge of the target device (i.e. the device's rhythm), which may not be true, especially when there are a larger number of devices [13]. Tap-to-Pair does not require users to remember any pattern or rhythm beforehand. Also, Tap-to-Pair provides a more general binary sensing approach for wireless devices without requiring any additional sensors.

2.2 Correlation-based Techniques

Correlation-based techniques match two similar signals for interaction purposes (e.g., object selection, authentication, and cross-device association). For example, when users bump [16] or shake two devices simultaneously [22], the IMU signals on both devices become highly correlated and can be used for association purposes. Users can also pair with a display by shaking a device according to the shake and pause sequence shown on the display [26]. Aside from cross-device association, the IMU signal of a device can be correlated with image data for user association [38] or with touchscreen data for touch event association [31]. However, such techniques only work for IMU-equipped devices that are small and light enough for users to move freely.

Recent research has explored various motion correlation [34] techniques to select objects by correlating the user's movements with moving objects on a distal display. The spatial correlation is calculated between the moving path of the user's gaze [11, 37], hands [5, 36], head and hand-held object [8, 9] and that of the displayed objects. The object is selected when the correlation coefficient between the two paths is high. In comparison, temporal correlation-based techniques require less computational resources by correlating the time series of one dimensional signals. For example, SynchroWatch [30] selects blinking objects on a smart watch by correlating a blinking pattern with magnetic field changes (users are required to wear a magnet on the thumb to generate the magnetic fields).

Tap-to-Pair is a temporal correlation-based device association technique. It leverages the built-in wireless transceiver for sensing purposes, without requiring additional hardware to be equipped on the devices (e.g. IMU), on users' bodies (e.g. magnet ring), or in the surrounding environment (e.g. camera). Also, Tap-to-Pair does not require the user to move the devices, which enables it to work with both small and large devices as long as their RSSIs can be effectively changed by tapping.

2.3 Other Cross-device Association Mechanisms

Aside from the correlation-based techniques, there are also many association mechanisms based on other interaction techniques (e.g., gesture recognition, image recognition, rhythmic input, and proximity detection).

Pointing one device at another is an efficient gesture to associate two devices. One way to detect pointing is to use directive transmitters like Infrared [14, 40] or laser [2, 25]. The targeted device receives the directive signal, and the transmitting and receiving devices are then associated. Microphones and speakers [1, 27, 32] can detect the movements involved during pointing gestures based on Doppler effect. Computer vision techniques can also recognize the user's pointing direction without equipping the devices with additional hardware [4, 12]. Even though pointing can be more efficient than continuous tapping, the required hardware and computation resources make it difficult to deploy ubiquitously. In comparison, Tap-to-Pair requires no extra hardware on the initiating device, and only a binary display on the target device. Tap-to-Pair's correlation algorithm also requires minimal computing resource and can be run on-chip. Tap-to-Pair can even work with occlusions between devices since it uses radio frequency signals.

If the initiating device has a camera, users can scan the tag on the target device [23, 28] to pair. Users can even snap a picture of the target device and use computer vision techniques for device recognition [6, 10]. However, this technique also require a considerable amount of computation for image processing and object recognition. Moreover, cameras are usually expensive and power demanding for commercial products.

Several US patents [15, 19, 20] pair the scanner with the advertiser that has the largest RSSI, which they assume is the closest one. However, the absolute value of RSSI is very noisy and can be affected by many factors (e.g. transmit power) aside from distance, which can lead to high error rates. Such methods also do not support remote pairing. Tap-to-Pair, on the other hand, relies on the relative change of RSSI signals, which is more robust and enables association with distal target devices.

Table 1. Additional resources requirements of different wireless device association methods

	Initiating Device	*Target Device*	*Other requirements*
IR/Laser [2, 14, 25, 35]	IR/laser transmitter	IR/laser receiver	None
Acoustic Gesture [1, 27, 32]	Speaker	Microphone	None
Vision Gesture [4, 12]	None	None	Kinect and cloud services
Synchronous Gestures [16, 22]	IMU	IMU	None
Tagging system [23, 28]	Camera	Tags	None
Snapping pictures[6, 10]	Camera	None	Cloud services
Rhythmic Taps [18, 39]	Binary Sensor	Binary Sensor	None
Tap-to-Pair	None	Binary display	None

In Table 1, we summarize the resource requirements of major device association techniques. The minimal hardware and computing resources required by Tap-to-Pair make it suitable to a wide range of wireless devices with different shapes, weights, and cost constraints.

3 TAP-TO-PAIR

Tap-to-Pair correlates the tap-induced signal strength changes with the displayed blinking patterns to recognize the user's pairing intentions. In this section, we explain in detail Tap-to-Pair's sensing mechanism, pattern representation and mapping, and the synchronization recognition algorithm.

3.1 Sensing Mechanism

The "hand effect" is well known among consumer electronics antenna designers: the performance of antennas can be greatly impacted by nearby hands. Hands placed in close proximity to the antenna may block wireless signals and detune the matched impedance of the antennas [17]. Tap-to-Pair, however, takes advantage of the signal changes caused by the "hand effect". A hand tapping near the antenna area will significantly decrease the transmitting signal strength, which can be restored by moving the hand away. Scanning devices can then detect users' tapping on the advertising device by monitoring its signal strength. A *tap* event can be detected by a large signal drop, a *release* event by a corresponding signal increase, and a *hold* event by the relatively small signal variations after a tap or release event. The signal drops must be larger than the signal fluctuations for reliable recognition. Figure 2 illustrates that RSSI drops significantly after a tap event, which is much larger than its normal variations.

3.2 Pattern Representation and Mapping

We use a simple binary code to represent patterns. The binary digit '1' means that the user's hand and the device are in contact, while '0' means there is no contact. We define a tap event as a '0' to '1' transition, a release event as '1' to '0' transition, and hold events in between. A blinking pattern can then be created by repeating a *code*. For example, the code "01" can be repeated to generate a pattern "010101...". We define *code-set* as the union of codes. Codes in a code-set must be acyclic, since it constantly repeats and users may start synchronizing at any moment. Below we show a code-set consisting of all codes with lengths less than 5.

$$\{01, 001, 011, 0001, 0011, 0111\}$$

We highlight two parameters that describe the feature of repeating patterns: *digit period*, which is the time duration of one digit; and *duty cycle*, which is the ratio between the number of '1's and total length of the code. When mapped to users' tapping behavior, a shorter digit period translates to faster taps, and higher duty cycle

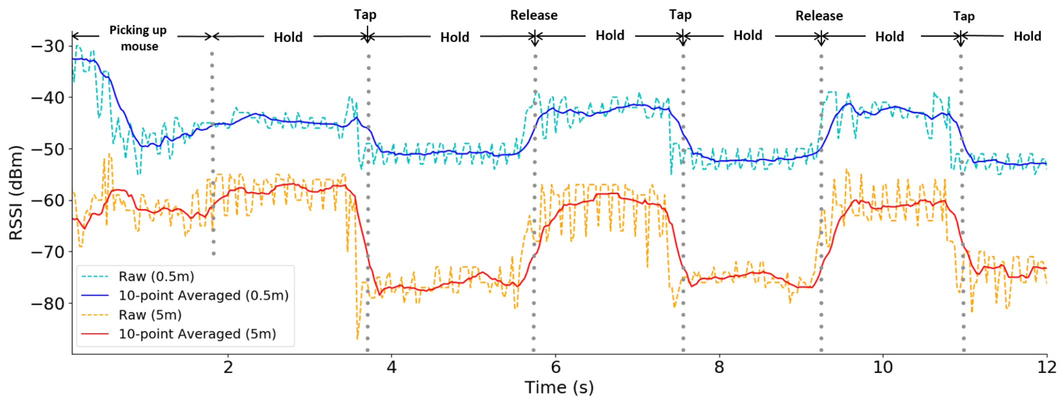

Fig. 2. Raw (dash line) and 10 point averaged (solid line) RSSIs for two scanning devices at 0.5m (blue) and 5m (red) away from the same advertising device. The timing of different events and contact status are marked at the top.

means longer contact between hands and the device. The codes can be single-tap (e.g. 00001) or double-taps (e.g. 01011). Different types of codes can yield different synchronization performance, which we will explore in our first study. The repetition of one code enables the user to anticipate the timing of the next movement. This anticipation is helpful for two reasons: 1) It helps users to start moving their hands in advance so that the contact can be made the instant the digit '1' is displayed; 2) It requires less mental effort after the user "finds the rhythm".

3.3 Synchronization Recognition Algorithm

Tap-to-Pair uses the Pearson correlation coefficient [3] to measure synchronization between the received RSSI and the displayed pattern. A *sliding window* is needed for real-time calculation of the coefficient ρ. The RSSI values for each advertiser (X) are correlated with the display sequence (Y) using Equation (1), in which μ is the mean, σ is the standard deviation, and $\mathbb{E}$ is the expectation. If ρ is larger than the preset threshold TH, the scanner will send a connection request to the corresponding advertiser. The algorithm can be implemented *on-chip* as it does not require heavy computation.

$$\rho_{X,Y} = \frac{\mathbb{E}\big[(X - \mu_X)(Y - \mu_Y)\big]}{\sigma_X \sigma_Y} \tag{1}$$

Synchronized fluctuations of RSSI that occur by chance can generate false positives. Thus, we only calculate correlation when the standard deviation of the RSSI sequence is larger than 2 (empirically decided). Also, no lag compensation is added to account for user's reaction time as in [30], as users may move in anticipation, resulting in both leading and lagging events. Also, while lag compensation may reduce pairing time, it may also increase false pairing rate when there are multiple scanners. The sliding window and threshold should be set to balance between the pairing time and the false pairing rate. A longer sliding window or a higher threshold reduces false positives but increases the pairing time.

4 USER STUDY 1

This study had three goals: 1) Validate that the RSSI signal can effectively communicate taps. To do this, we strapped an IMU onto the participant's wrist. The tap signal recovered from the IMU data served as the ground

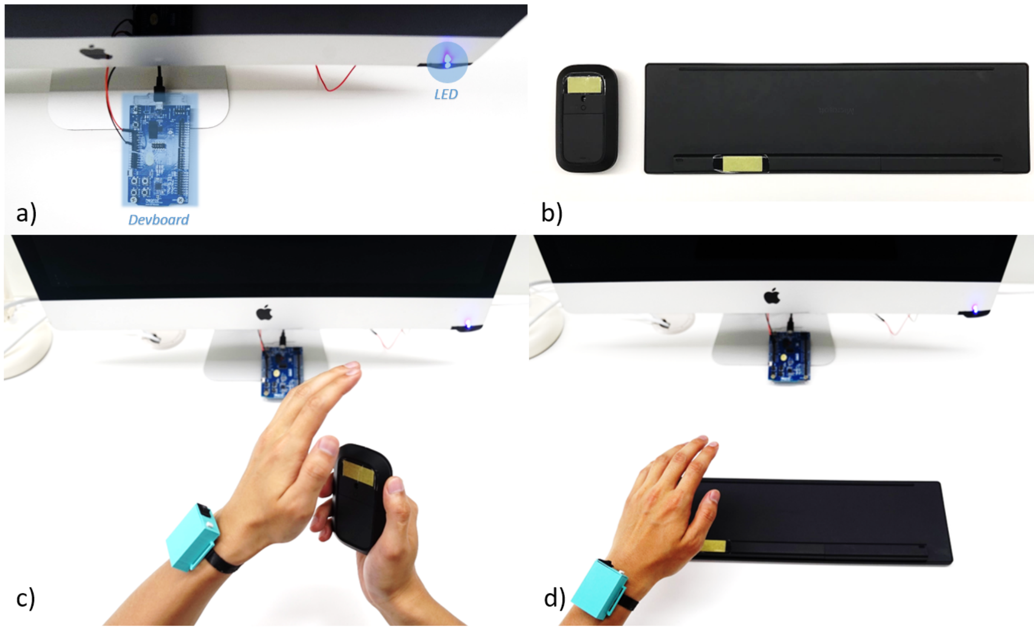

Fig. 3. a) Placement of development board and LED light. b) Marked tapping area on the mouse and the keyboard (flipped). c) A participant taps on the mouse. d) A participant taps on the keyboard.

truth and was compared with the data recovered from RSSI signal; 2) Provide blinking pattern design guidelines by analyzing participants' synchronization performances for different codes at different digit periods; 3) Provide data for determining the optimal correlation threshold and sliding window size.

4.1 Participants and Apparatus

We recruited 12 right-handed students (9 males) from a local campus as participants, with an average age of 23.4 (SD = 1.9). Each participant was compensated 15 USD for their time.

We used a Bluetooth transceiver with ARM Cortex-M4 core (nRF52832 by Nordic Semiconductor[1]) development board (devboard) as the scanning device, and a 3mm diameter blue LED light driven by the chip's IO port to display blinking patterns. The LED was illuminated for digit '1' and dimmed for digit '0' to better match users' mental modal, i.e., act (tap) when something happens (light illuminated) and relax (release) otherwise. The LED was taped onto the bottom right corner of a monitor, where LED lights are typically located (Figure 3a). We placed the devboard 0.5m away from the participants and connected it to a PC via USB cable to report RSSI values.

We used a set of off-the-shelf Microsoft Designer Bluetooth mouse and keyboard[2] as advertisers. On each device, we marked the surface area near the antenna with yellow tape (Figure 3b). We chose the mouse to study users' tapping behaviors on small and light devices, and the keyboard for larger and heavier devices. We asked participants to tap on both devices using their left hand, with the mouse held in the right hand (Figure 3c) and the keyboard placed on the table (Figure 3d). As right-handed people generally use the mouse with their right hand,

[1]https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52832
[2]https://www.microsoft.com/accessories/en-us/products/keyboards/designer-bluetooth-desktop/7n9-00001

it is more natural for them to pick up and hold the mouse using right hand and tap with the left. The keyboard's marked tapping area was on the left side, which makes it easier to tap using the left hand.

An IMU module, a Bluetooth module, and a Li-polymer battery were placed inside a small 3D-printed box and strapped on the participant's left wrist. We used Bluetooth for data communication to provide a tapping experience without constrains from cables.

4.2 Experiment Design and Procedure

The code-set used in the study facilitated the analysis of codes with different lengths, duty cycles, and number of taps. To reduce the amount of time required for each participant, we used the following code-set:

$$\{01, 001, 011, 0001, 0011, 0111, 00001, 00011, 00111, 01111, 01001, 01011\}$$

The set contains codes with four different lengths including ten single-tap codes (e.g. 01) and two double-tap codes (e.g. 01001). There are also four codes with the same length but different duty cycles (00001, 00011, 00111, 01111).

We used a within-subjects experiment design with three factors: *device* (mouse vs. keyboard), *code* (as described above), and *digit period* (200ms, 400ms, and 600ms). 200 ms is near the human visual synchronization threshold [29], while patterns with digit period > 600 ms are too long and difficult to learn.

During the study, the participant first completed a demographic questionnaire. We then explained the task and demonstrated how to tap on each device. The participant was asked to cover the whole marked area during tapping to ensure sufficiently large RSSI changes. The participant then warmed up for one minute using randomly generated codes, and then completed 3 sessions of tasks, each corresponding to a digit period. Each session contained 2 blocks, each corresponding to an advertising device. In each block, the participant performed synchronized tapping with each code in the code-set for 20 seconds, with 15 second rest between adjacent codes. The order of different sessions and blocks were counter-balanced using Latin squares, while the code appeared in random order. No feedback on tapping performance was provided in this study, instead, the participant was required to notify the experimenter when he/she felt the tapping was in synchronization with the blinking pattern.

A two-minute break was enforced between each session, during which the participant rated the mental and physical demands on a 7-point Likert scale questionnaire (a higher score indicates more effort). After completing all three sessions, we interviewed the participant on his/her preferences among the three sessions and different types of codes. The experiment took about one hour.

4.3 Results Analysis and Discussion

We collected the acceleration and RSSI data of $12\, participants \times 3\, digit\, periods \times 2\, advertisers \times 12\, codes = 864$ tap trials with a total time duration of 17,280s (4.8 hour). We compared the results of taps sensed by acceleration and RSSI signals to see if Tap-to-Pair could reliably detect taps, then analyzed both quantitative and qualitative results to provide suggestions for the choice of digit period and code-set. We used RM-ANOVA for tests on parametric data, and Friedman test for subjective ratings. We used a Pearson correlation coefficient over the entire duration of a tapping block to indicate the overall synchronization performance for the block (ρ_{block}). We analyzed the results with merged data from both devices since no significant difference of correlation (ρ_{block_ACC}) was found ($p = .09$, calculated using the accelerometer data).

4.3.1 RSSI vs Accelerometer. The accelerometer data was processed to yield a binary sequence representing the tapping behaviors as described in Section 3.2 (example data shown in Figure 4a, processing details in APPENDIX A). No significant difference was found between ρ_{block_RSSI} and ρ_{block_ACC}, indicating similar performance of the accelerometer and RSSI when using correlations to quantify the degree of tap synchronization ($p = .49$). To

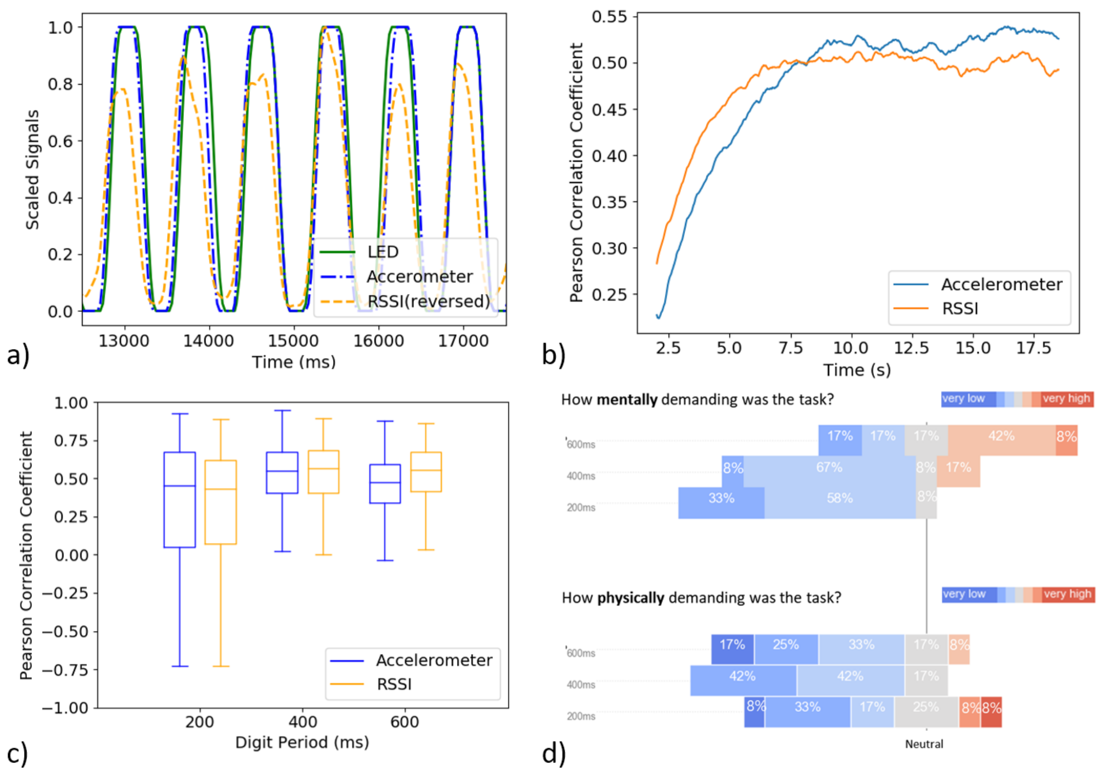

Fig. 4. a) LED status (green), scaled RSSI (orange), and tap signals recovered from accelerometer data (blue) for a tapping sequence. All signals are 10 point smoothed. b) Averaged Pearson correlation coefficient between the blinking pattern and accelerometer (blue), RSSI (orange) using a 2s sliding window. c) Boxplot of overall correlations of accelerometer (blue) and RSSI(orange) for different digit periods; d) Stack plots of mental and physical demands rating for different digit periods (7 means very high demand, 1 means very low demand).

observe the trend of the synchronization performance over time, $\rho_{sliding\_RSSI}$ and $\rho_{sliding\_ACC}$ were calculated using a two-second sliding window for each tapping block and were averaged over all blocks (Figure 4b). Both correlation curves first increase for about 7.5s then plateaued. The greater level of noise in RSSI signal led to a higher $\rho_{sliding\_RSSI}$ before saturation and a lower $\rho_{sliding\_RSSI}$ when it had plateaued. The averaged self-reported successful synchronization time was 7.7s, which shows the participants had a relatively accurate feeling of the synchronization performance.

4.3.2 Performances of Different Digit Periods. We only used RSSI data for analysis henceforth. Participants' synchronization performance for the 200ms digit period is significantly worse than that of longer digit periods ($F_{1,11} = 14.6, p < .01$). The negative correlations at 200ms (Figure 4c) show that the participant had trouble staying in phase with the blinking pattern (i.e. tap when LED is off and release when LED is on). Several participants commented that they were more likely to tap out of phase during the 200ms session since the blinks were too fast.

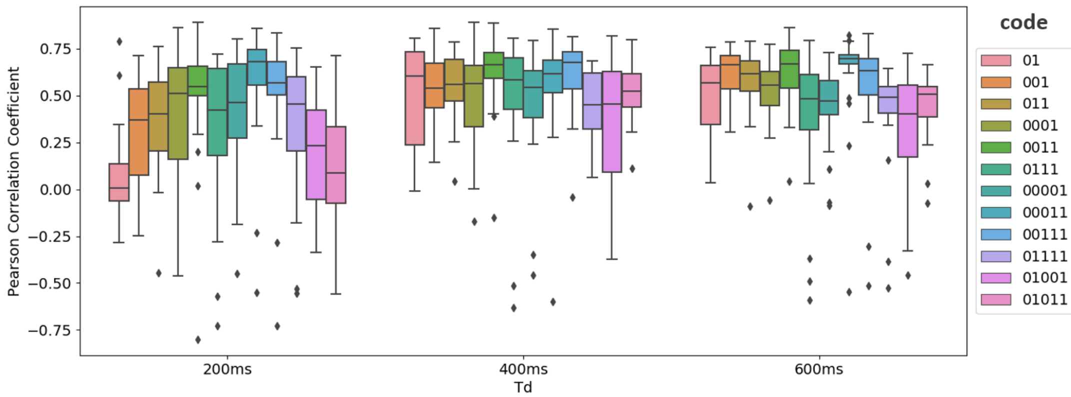

Fig. 5. Boxplot of the RSSI based correlation of different codes with the three digit periods.

We ran binomial tests and found that a majority of ratings for physical ($p < .0001$) and mental ($p < .01$) demands fall on the low side (Figure 4d). According to the data, digit periods had a significant effect on participants' mental effort ratings ($\chi^2(2) = 8.05$, $p < .05$), with post-hoc results showing longer digit periods are more mentally demanding (Figure 4d). We suspect that this is because the participants have to focus for a longer time to learn patterns with long periods. There was no significant difference for physical effort ratings between the three digit periods though ($p = .57$). Several participants did complain that the 200ms session was tiring. For example, P7 said "I had to pay more attention when the blinking is slow. The fastest blinking session made my hands tired."

In the interview portion of the study, among all 12 participants, 6 participants chose 200ms as the most preferred digit period, 4 chose 400ms, and 2 gave a tie between 200ms and 400ms. Interestingly, the 600ms period was unanimously rejected. We found that participants tend to observe the pattern first instead of reacting to it immediately. By doing so, they can move in anticipation for better synchronization performance. Moreover, after they have memorized the pattern, they do not need to pay as much attention to the blinking LED, which alleviates the associated mental burden. 600ms was rejected because all participants agreed that patterns were too slow and took too long to observe and learn.

4.3.3 Performances of Different Codes. We categorized the codes into groups with different lengths (2,3,4,5), duty cycles (>0.5, =0.5, <0.5), and number of taps (single, double). Code length yielded a significant effect on participants' synchronization performance ($F_{3,33} = 11.6$, $p < .0001$), with post-hoc results showing codes of middle lengths (3 and 4) have significantly better performances than that of others (2 and 5) ($F_{2,22} = 11.5$, $p < .001$, Figure 5). The high tapping frequency of code '01' made the participants more likely to tap out of phase. In the interview, 7 participants preferred double taps for easier learning experience. P12 mentioned that "The different tap duration can help me learn so I can pay less attention". Even though they are easier to learn, the double-tap codes led to significantly lower performance than that of single-tap codes ($F_{1,11} = 50.7$, $p < .0001$). We suspect the participants focused too much on the duration differences instead of the absolute timing, which was used to calculate the correlation coefficients. We did not find the duty cycle have significant effect on correlations ($p = .34$). The participants did have a preference for lower duty cycle though. Eight participants mentioned that they did not like codes with high duty cycle. P3 said "I feel like I can tap more precisely than release".

5 PROTOTYPE IMPLEMENTATION

In this section, we introduce the implementation of Tap-to-Pair prototype on the scanning device in four parts: 1) Determining the digit period used for the blinking pattern; 2) Determining the sliding window and the threshold used in the algorithm; 3) Providing code-set design suggestions; 4) Implementing the correlation algorithm and blinking display on the scanner.

5.1 Digit Period

In Study 1, we found that longer digit periods were more mentally demanding, however participants also reported the 200ms session created higher levels of fatigue. The synchronization performance of the 200ms session was also worse than that of longer periods (Figure 4c and Figure 5), as users tend to tap out of phase. So even though more participants prefer 200ms, we used a digit period of 400ms in our prototype scanning device, which better balances synchronization performance with the users' preferences.

5.2 Sliding Window and Threshold

The sliding window and threshold are set to strike a balance between the pairing time and false pairing rate (chances of connection with non-target scanners). We define *first pairing time*, which is the first time when the calculated correlation is higher than the threshold. Using the RSSI data collected in the previous study, we ran simulations with different window sizes and thresholds to calculate the first pairing time and worst case false pairing rate for each combination. These two parameters can be adjusted according to different scenarios. For example, a scenario with multiple Tap-to-Pair compatible scanning devices will require a longer sliding window and threshold to avoid false positives.

Since most wireless devices need users to start advertising intentionally (e.g. by pressing a button), there are usually no false positives before the advertising starts. Moreover, when the device is paired, it will stop advertising. So false pairings can only happen during tapping, when the correlation coefficient of non-target scanners are also higher than the threshold. To evaluate potential false positives for each code, the patterns generated by the other 11 codes in the code-set are shifted circularly within the sliding window and correlated with the collected RSSI data. We then calculate the first pairing time with the target code, and the shortest pairing time with the non-target codes.

The simulation was run for all codes in the code-set, and their false first pairing times were then averaged. The optimization goal was to maximize the ratio of averaged false first pairing times to true first pairing times, so that the advertiser pairs with the target scanner before other scanners. The simulation results (Figure 6a) show that a sliding window of 3s and a threshold of 0.75 yield the largest ratio.

5.3 Code-set Design Implications

We simulated the first pairing time matrix using a 3s window size and a 0.75 threshold (Figure 6b), and found codes with the same length have a higher chance for false pairings. This indicates that people are less sensitive to duty cycles. However, we would like to point out that this simulation reflects the worst-case scenario, and fewer false pairings would be expected in a real setting. Based on the study and simulation results, we generated the following code-set design suggestions,

- Use single-tap codes since they yield better synchronization performance.
- Use codes with low duty cycles as they are preferred by the participants.
- Use codes with medium lengths since they yield better performance than short and long ones.
- Use codes with different lengths on different scanners to lower the false pairing rate.

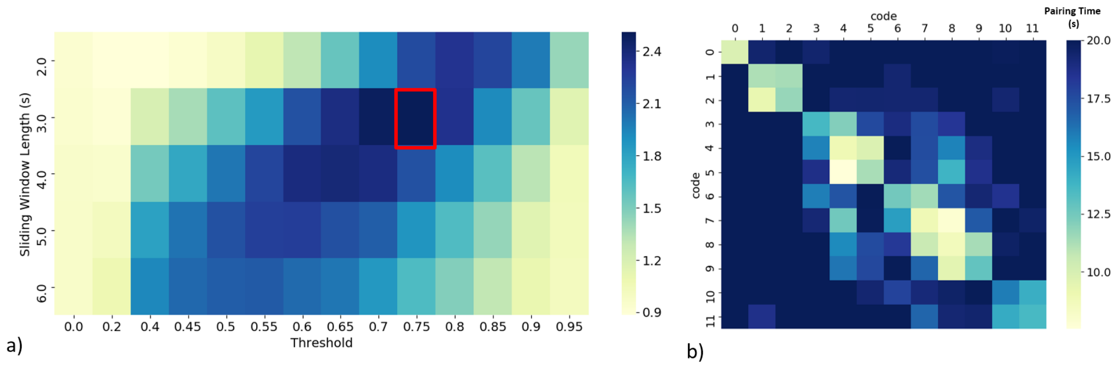

Fig. 6. a) Ratio of true first pairing and averaged false first pairing times for different sliding windows and thresholds (the larger the better). The combination that yields the largest ratio is marked with a red box. b) First pairing time matrix with 3s sliding window and 0.75 threshold. Codes 0-11 correspond to the codes used in the previous study with increasing order.

5.4 On-Chip Implementation

In our prototype system, we implemented the algorithm on-chip using the same Bluetooth devboard as in the previous study. The on-chip implementation shows that Tap-to-Pair's recognition algorithm can run on a commercial, low-cost Bluetooth IC. Note that no modifications were made to the Bluetooth stack, which means that Tap-to-Pair works with the current Bluetooth protocol.

The algorithm collects RSSI values for each advertising device using a 30ms timer, which is fast enough to capture RSSI signal changes caused by hand taps. For each device, we maintain a circular First-In-First-Out (FIFO) RSSI queue with a length of 100 ($0.03 \times 100 = 3s$ sliding window). A LED status queue with the same length is also maintained. The Pearson correlation coefficient is then calculated in real-time between each RSSI queue and the LED status queue. If the coefficient is larger than 0.75, the scanner sends a connection request to the corresponding advertiser. The device's RSSI queue is cleared after connecting.

6 USER STUDY 2: EVALUATION

We conducted a second user study to evaluate the performance of Tap-to-Pair with multiple scanners. The goals of the study were 1) To valid the feasibility and usability of Tap-to-Pair; 2) To evaluate Tap-to-Pair's performance at different distances between advertisers and scanners; 3) To measure the false and failed pairing rates between different codes.

6.1 Participants and Apparatus

We recruited another 12 participants (10 males) who did not take part in Study 1. All participants were staff or students from the local institution with ages ranging from 18 to 34 (Mean = 26.8, SD = 10.2). Each participant was compensated 10 USD for their time.

The same mouse and keyboard used in the previous study were used to evaluate Tap-to-Pair's performance for both small and large advertisers. The advertiser's Bluetooth signal can typically reach scanners 5-7m away in indoor environments. Considering daily use cases and sizes of regular rooms, we placed three devboards at 1m (code 01), 2m (code 0001, to the left of the user), and 5m (code 001) away from the participant (Figure 7). We selected codes $\{01, 001, 0001\}$ following the code-set design guidelines: single-tap codes with low duty cycles and

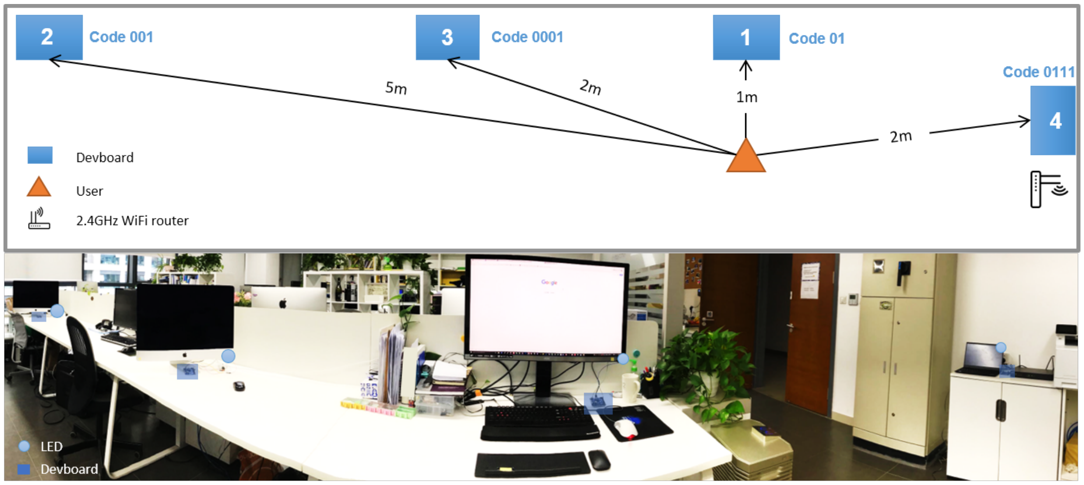

Fig. 7. Placement of the four development boards and their corresponding codes. One was placed in front of the participant, two were placed to the left, and one was placed to the right.

different lengths. We added one scanner at 2m away on the right side of the user using code 0111 to evaluate the false pairing rate between codes with the same length (0001 vs 0111). All devboards were programmed with their display pattern and our recognition algorithm and then connected to the same PC via USB cables to report the received RSSIs. We taped a 3mm diameter blue LED to the bottom left of the monitors and the top left of the laptop, both of which were driven by one IO port of the devboard to display blinking patterns. Fast blinks of the LEDs indicate a connection event when the correlation with any advertiser passes the threshold. A working 2.4GHz WiFi router was placed near the participant to mimic a typical electromagnetic environment in the office.

6.2 Experiment Design and Procedure

We used a within-subjects experiment design with two factors: *advertiser* (mouse vs keyboard), *scanner* (Scanner 1, 2, 3, 4). Upon arrival, the participant was asked to sit in front of a desk and complete a demographic questionnaire. The mechanism of Tap-to-Pair was explained, then the participant warmed up for two minutes. We did not instruct users on which hand to use anymore. The experiment contained three sessions. Each session contained two blocks counter-balanced using a Latin square, each corresponding to an advertiser. In each block, the participant attempted to pair with the scanners in a random order. The pairing was timed and any attempt lasting longer than 15s was counted as a failure. After each block, we asked the participant if the pairing time felt fast, medium, or slow. A total of three sessions were conducted. Participants were allowed to rest between sessions.

After all sessions were completed, we asked the participants to use a 5-point Likert scale to rate Tap-to-Pair on effort, enjoyment, as well as provide an overall rating for the association technique (the higher the better). On average, it took less than 30 minutes for a participant to complete the whole experiment.

6.3 Results Analysis and Discussion

We collected $4\ scanners \times 2\ advertisers \times 3\ sessions \times 12\ participants = 288$ pairing attempts in total. The pairing failed for 4.8% (14/288) of all attempts. Among all 14 failures, 11 occurred when using the keyboard. For successfully paired attempts, the average connection time was 5.7s (SD = 2.5s). The participants felt the pairing

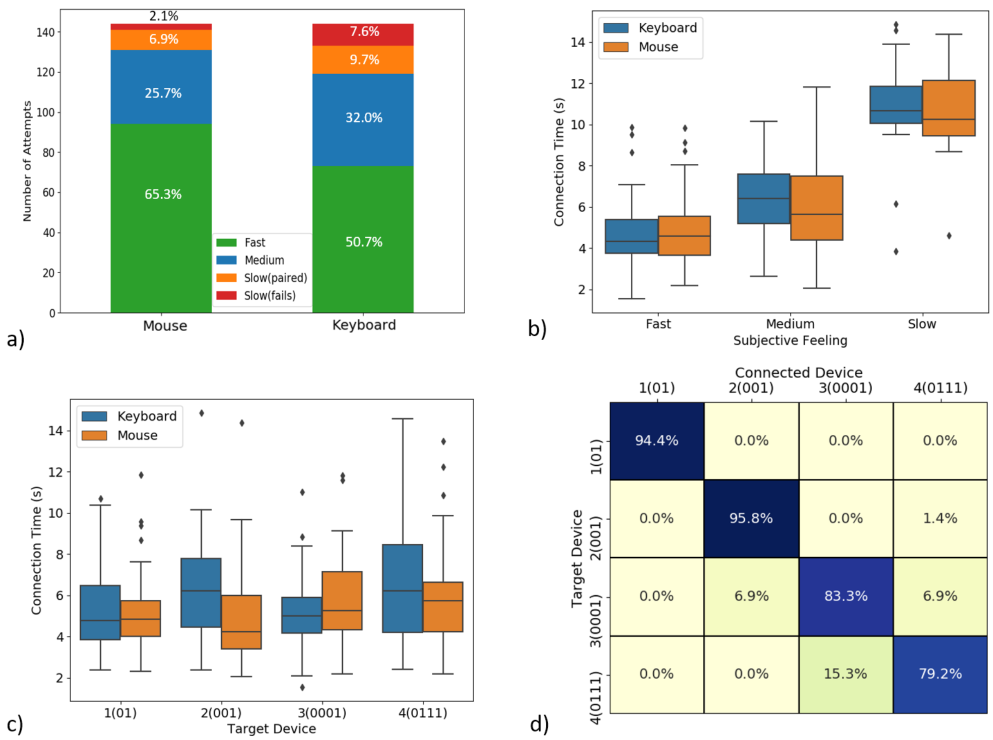

Fig. 8. a) Counts of participants' perceived feelings of pairing times; b) Boxplot of the perceived feelings of successful connection times using the mouse (orange) and the keyboard (blue); c) Boxplot of successful connection times for four target devices using the mouse (orange) and the keyboard (blue); d) Pairing confusion matrix. The diagonals are percentages of successful pairing attempts (paired within 15s), while the others are false pairings. The sums of each row are less than 100% because there were failed attempts.

was fast or medium for 88% attempts (92% for mouse and 83% for keyboard, Figure 8a). Several participants commented that Tap-to-Pair paired faster than they expected. In general, participants considered connection times longer than 9s to be slow (Figure 8b).

RM-ANOVA found that the connection time when using the keyboard was significantly longer than when using the mouse ($F_{1,11} = 8.22$, $p < .05$). We observed that participants' hands were likely to drift away from the marked area on the keyboard, which led to smaller RSSI changes during tapping and thus longer pairing times (Figure 8c). This observation was echoed in comments from P6 "I always tap at the wrong place on the keyboard". There was no significant difference among the connection times between the four scanners ($p = .12$), which demonstrates that Tap-to-Pair has similar performance for scanners at different distances. Note that the advertiser was always in advertising mode in the study, and scanner recorded the RSSI before the participants start tapping. Sometimes these RSSI signals happened to be highly correlated with the target pattern. This explains why some connection times were shorter than the sliding window.

We assigned scanners 1, 2, and 3 with codes of different lengths following our code-set design guidelines. The overall pairing success rate was 93.7%, assuming their false paired attempts with Scanner 4 had the same

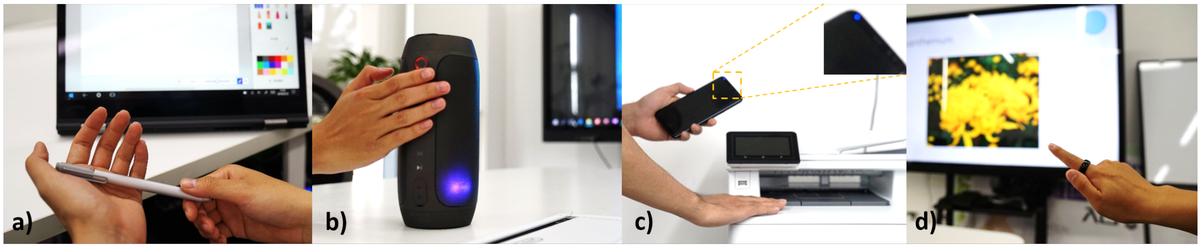

Fig. 9. a) Pairing a stylus with a tablet; b) Pairing a Bluetooth Speaker with a TV; c) Pairing a printer with a smartphone (breathing light zoomed in on the top right corner); d) Pairing a smart ring with a remote display

failed and false pairing rates (Figure 8d). For example, Scanner 3 had a overall failed pairing rate of 2.9% and a false pairing rate of 6.9% with Scanner 2, then 90.2% of the 6.9% false pairs with Scanner 4 were assumed to be successful pairs. The overall pairing success rate was 88.3% for all four devices, with a false pairing rate of 7.6% and pairing fail rate of 4.1%. The lower success rate was mainly due to the relatively high confusion between Scanner 3 and 4 (5.5%). This matched our simulation results, which indicated that codes with the same length are more easily confused with one another. We observed that participants tend to release their hands early even when the LED light is still ON, which could explain the higher chances for 0111 to be recognized as 0001 (15.3%) than the other way around (6.9%). Participants' hands were likely to change positions when switching sides (left to the right) to target Scanner 4, which could also lead to a lower pairing success rate for Scanner 4.

Most participants (11/12) rated the effort required in using Tap-to-Pair as low (effort >= 3, Median = 3.5). All participants (12/12) reported that they enjoyed the pairing experience (enjoyment >= 3, Median = 4), and liked to use Tap-to-Pair (overall rating >= 3, Median = 4). Participants described Tap-to-Pair as "really fun to use" (P3 and P11) and "very robust" (P4).

7 APPLICATIONS

In this section, we explore how Tap-to-Pair can create easy and spontaneous device association experiences with four potential use cases.

Stylus Increasingly, one stylus may be shared between multiple computing devices, especially during collaboration. Currently, users have to disconnect the stylus from one device and connect it to another, which changes the displayed content on screen and can potentially disrupt users' workflow. Also, the resources for stylus are extremely constrained due to their small size and special shape. The ability of Tap-to-Pair to work with resource-constrained advertising devices might afford a smoother pairing experience by initiating association from the stylus itself (Figure 9a).

Wireless Speaker One of the most widely available advertising devices is the wireless speaker. These speakers are usually placed further away from large displays like TVs. To pair with the wireless speaker, users usually have to operate an interface on the remote display. With Tap-to-Pair however, users can pair the speaker with or switch the speaker between different large displays from the speaker itself (Figure 9b), which is usually closer to the users.

Wireless Printer Many wireless printers support peer-to-peer connection with smartphones for mobile printing. Currently, this feature is usually enabled using NFC technology[3], which requires the smart phone

[3]An example can be found at https://support.hp.com/id-en/document/c04174486

to support NFC. Tap-to-Pair however, can work with both Bluetooth and WiFi-Direct technologies, which already exist in most smartphones (Figure 9c).

Wireless Ring Wireless rings can be convenient controllers due to the dexterity of our fingers. However, the small size of these rings severely restricts their hardware resources. Tap-to-Pair however, can initiate association from a wireless ring with taps of a thumb (Figure 9d), which would enable an easy and consistent input experience on distal displays.

8 DISCUSSION

8.1 Synchronous Tapping Behaviors

In the first study, the short single-tap codes and double-tap codes had worse performance, even though they were considered easier to learn and preferred by many participants. After the participants learned the code and "found the rhythm", they were more likely to lose focus and tap according to their own rhythm. The faster the participants learned the code, the earlier and easier they payed less attention to the blinking pattern and were more likely to tap out of sync. This overconfidence could be one of the major reasons for the inferior synchronization performance of such codes. To improve Tap-to-Pair's performance for such codes, visual indications of the correlation coefficient could be added to help users stay focused. For example, RGB LED could be used to emit green light when the correlation is low and red light when correlation is high. Visual feedback could also help users to better match the pattern's duty cycle, reducing the confusion rate between codes with the same length. An extra LED could blink quickly during digit 1s (contact) to remind the user to hold the tap.

Since the participants have to focus on the blinking displays during the association process, their hands are likely to move away from the marked tapping area while tapping. We observed in the study that sometimes they had to stop and adjust tapping positions. Such interruptions slow down the device association process and complicate the pairing experience. A curved surface around the area could provide haptic feedback to prevent users' hands from drifting away. Special structures or materials that produce different sounds when tapped on could also help.

8.2 Scalability

To use Tap-to-Pair on off-the-shelf advertisers, there have to be areas on the device where tapping can cause signal changes larger than the electromagnetic noise. In this paper, the antennas of the mouse and the keyboard are placed near the surface of the casing. Tapping near the antennas can induce adequately large signal changes. This practice is common for many devices, as placing the antenna near the exterior reduces signal blockage. A one-time search routine could be carried out to find the tapping area for unmodified advertisers. Users can record the transmitted signal changes when tapping different parts of the device and mark the area with the largest RSSI drop. Manufacturers could also mark such areas with almost no extra cost. The antenna can also be directly etched on the device's surface to lower the physical demand of Tap-to-Pair, so that finger taps on the antenna can induce large signal drops.

Even though our prototype system uses Bluetooth technology, Tap-to-Pair can be scaled to other wireless protocols as long as the protocol provides a signal strength indicator (e.g. ZigBee and WiFi Direct). It also works for protocols that have a star topology, such as WiFi and RFID. A central router could record wireless signal changes of all connected devices, and correlate the signal changes of each device with blinking patterns of other devices.

Tap-to-Pair could also be implemented on current scanners with software changes instead of on-chip firmware changes. For example, BlueZ[4] can access Bluetooth modules on devices running Android, iOS, and Linux. Users could still initiate association using GUI on the scanner after the implementation of Tap-to-Pair, since no changes

[4]http://www.bluez.org/about/

were made on the protocol level. We believe such compatibility makes Tap-to-Pair a good complement device association scheme on current scanning devices.

Most scanners do not transmit signals, and advertisers only transmit in the advertising mode, which usually happens right before pairing. So, it is unlikely for Tap-to-Pair to jam the frequency channels even with a large number of advertisers and scanners. Even if certain frequency channels are jammed, different wireless protocols have their own mechanisms to choose a less crowded channel to broadcast (e.g. Clear Channel Assessment specified in IEEE 802.11).

Three participants complained that the blinking LEDs were distracting during the evaluation. This is especially true when there are many scanners. To alleviate this issue, a special pattern could be used to trigger the LEDS to begin blinking. Once the special pattern is detected, all scanners start to blink for a set amount of time, then stop automatically. In this way the visual distractions only exist during the association process.

8.3 Disconnection, Authentication, and Pattern Generation

We did not implement a disconnection mechanism in this paper, since there are many ways to disconnect from a scanner, such as to power cycle the advertiser or to put it into advertising mode. In most cases, switching between different devices is why users disconnect the advertiser. When connected, users can synchronize tapping with the pattern of another scanner to switch, since RSSI can still be recorded during connection. Users could also tap a special sequence as short cut to switch to the previous scanner.

Our prototype Tap-to-Pair scanner relies on the wireless protocols' own authentication process for secure pairing, which usually demand more resources on the scanner (e.g. a keyboard to input pin code). An authentication mechanism similar to TapSongs [39] can be implemented. Users can store a short piece of rhythm on scanners. After the advertiser is associated with the scanner, users can tap the rhythm to authenticate. In this way users can securely pair with multiple scanners by remembering only one piece of rhythm.

The blinking pattern for each scanner is programmed on-chip in the evaluation. The pattern for each scanner can also be generated in an ad-hoc manner. For example, all Tap-to-Pair compatible scanners can broadcast their pattern information, and a new scanner can first scan for the existing pattern information, then pick a pattern following the code-set design suggestions.

9 LIMITATIONS AND FUTURE WORK

Our evaluation of Tap-to-Pair was done in a controlled lab environment. In a noisy electromagnetic environment, Tap-to-Pair still works as long as the RSSI received by scanners are above the noise floor, since only relative changes of RSSI matter for Tap-to-Pair. However, in a dynamic environment people may cause RSSI drops by accidentally blocking the signal transmission path, which can lead to longer connection time and increase the false pairing rate. A field study is required to further evaluate Tap-to-Pair's performance in the wild.

Tap-to-Pair only works when the taps can cause signal changes as fast as the blinking pattern. In this paper, we used a Bluetooth advertiser with advertising intervals around 30ms and showed that it is fast enough to detect hand taps. Power constrained advertisers can use short intervals to advertise for the first 30s, then reduce their advertisement frequency over time. This technique is already employed by many Bluetooth devices, including the mouse and the keyboard used in this paper.

Tap-to-Pair requires the scanning device to scan continuously, which consumes more power. Many scanners like PCs or TVs are often connected to outlets. For power constrained scanners, periodic scans can be enabled by setting scan intervals to save power. For example, we can monitor the standard deviation of RSSI, and only enter continuous scan mode when it is higher than a threshold.

Current correlation algorithm only relies on absolute timing to calculate the correlation coefficient. We plan to develop algorithms that consider the relative time difference between adjacent taps, which tolerate less accurate

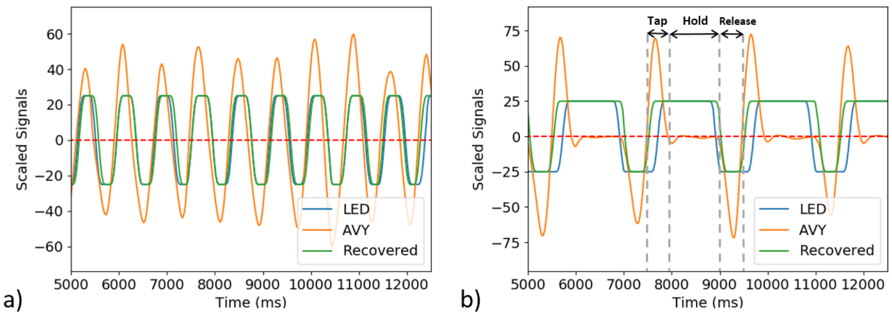

Fig. 10. Sample accelerometer data with 400ms digit period, code 01 (a) and 00111 (b). LED signal and recovered tap signal are scaled, while AVY is not.

ALGORITHM 1: Tap and release detection

Input: Smoothed AVY data array *avy* for one tapping block.
Output: Detected events array *events*.
$events[] = 0$;
$index_{ppeaks}, index_{npeaks}$ = peakdet(avy);
$index_{pcrossings}, index_{ncrossings}$ = crossingdet($index_{ppeaks}$,$index_{npeaks}$, avy);
for *each pindex in* $index_{pcrossings}$ **do**
    $nindex$ = findnearest($pindex$, $index_{ncrossings}$);
    $events[pindex : nindex] = 1$;
end

taps. By combing synchronization with code recognition, users can pay less attention to the blinking pattern after they have learned the pattern, which reduces the overall mental demands.

10 CONCLUSION

Tap-to-Pair is a new device association mechanism that can initiate pairing from off-the-shelf advertising devices without hardware or firmware modifications. The two devices are associated when the Pearson correlation coefficient of the wireless signal strength and the blinking pattern are higher than a threshold. We conducted a user study to show that RSSI signals can detect taps with performance similar to that of an IMU, and provide guidelines for scanners' pattern designs. The evaluation showed that users can initiate pairing using Tap-to-Pair with three scanners with a pairing success rate of 93.7%. We believe that by initiating pairing from advertising devices, Tap-to-Pair can improve the device association experience for current wireless devices, and unlock more possibilities for impromptu interactions in smart spaces.

A DETECT TAP, HOLD, AND RELEASE EVENTS USING ACCELEROMETER

We used an accelerometer because vision based methods like Optical Track may limit participants' postures due to occlusion. Adding sensors to the devices would change their form factors, which may greatly impact the tapping behavior. We tied accelerometers on participants' left wrist to keep the impact on tapping behavior to a minimum. The sample rate was 33Hz.

We find that the angle velocity on Y axis (referred to as AVY henceforth) has large and clean changes during tap and release events under current placement, which we used for later detection. A positive pulse of AVY means a tap event (hand starts moving toward the device), while a negative pulse means a release event (hand starts moving away from the device). We designate '1' to time periods from a tap start to a release start, which includes both tap and hold periods (Figure 10b). Time periods from a release start to a tap start is designated '0'. We can see that the LED and recovered signals are strictly out of phase during mismatches, which explains why for the 600ms digital period, the RSSI signal's tap sensing performance is better than that of the accelerometer.

To recover events from the AVY signal, we first found positive and negative peaks[5] of the smoothed AVY signal. Then we found the previous crossing point of a preset threshold for the peaks. After that, we got the recovered events array by assigning '1' for indexes from positive crossings to negative crossings, and '0' for indexes from negative crossings to positive crossings (Algorithm 1).

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their insightful comments. The work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001402, the Natural Science Foundation of China under Grant No.: 61572276, 61672314, and 61521002. The work is also supported by Tsinghua University Research Funding No.20151080408 and Beijing Key Lab of Networked Multimedia.

REFERENCES

[1] 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*. ACM, New York, NY, USA, 583–586. https://doi.org/10.1145/2493432.2493515

[2] Michael Beigl. 1999. Point Click - Interaction in Smart Environments. In *In: HUC'99: First Int. Symposium on Handheld and Ubiquitous Computing, Springer-Verlag*. 311–313.

[3] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*. Vol. 2. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–4. https://doi.org/10.1007/978-3-642-00296-0_5

[4] Matthias Budde, Matthias Berning, Christopher Baumgärtner, Florian Kinn, Timo Kopf, Sven Ochs, Frederik Reiche, Till Riedel, and Michael Beigl. 2013. Point & Control – Interaction in Smart Environments: You Only Click Twice. ACM Press, 303–306. https://doi.org/10.1145/2494091.2494184

[5] 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

[6] 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

[7] Ming Ki Chong and Hans Gellersen. 2011. How Users Associate Wireless Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1909–1918. https://doi.org/10.1145/1978942.1979219

[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

[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

[10] 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*. ACM, New York, NY, USA, 5909–5920. https://doi.org/10.1145/2858036.2858177

[5]Python peak detection algorithm by https://gist.github.com/endolith/250860, which is converted from MATLAB script at http://billauer.co.il/peakdet.html

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

[12] David Fleer and Christian Leichsenring. 2012. MISO: A Context-Sensitive Multimodal Interface for Smart Objects Based on Hand Gestures and Finger Snaps. In *Adjunct Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 93–94. https://doi.org/10.1145/2380296.2380338

[13] 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

[14] Seongmin Ham, Jihyung Lee, and Kyunghan Lee. 2017. QuickTalk: An Association-Free Communication Method for IoT Devices in Proximity. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3 (Sept. 2017), 56:1–56:18. https://doi.org/10.1145/3130921

[15] Philipp Hertel, Alexander W. Hertel, and Sheldon I. Walfish. 2015. Detecting a Communication Tap via Signal Monitoring.

[16] Ken Hinckley. 2003. Synchronous Gestures for Multiple Persons and Computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 149–158. https://doi.org/10.1145/964696.964713

[17] C. H. Li, E. Ofli, N. Chavannes, and N. Kuster. 2009. Effects of Hand Phantom on Mobile Phone Antenna Performance. *IEEE Transactions on Antennas and Propagation* 57, 9 (Sept. 2009), 2763–2770. https://doi.org/10.1109/TAP.2009.2027081

[18] 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

[19] Peter T. Liu. 2015. Low Energy Wireless Proximity Pairing.

[20] Julia Ma, Daniel C. Jenkins, Tim Jackson, Xi Zhu, and Gary W. Poole. 2011. Method for Automatic Pairing to a Wireless Network.

[21] Sébastien Maury, Sylvie Athénes, and Stéphane Chatty. 1999. Rhythmic Menus: Toward Interaction Based on Rhythm. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, 254–255. https://doi.org/10.1145/632716.632873

[22] R. Mayrhofer and H. Gellersen. 2009. Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing* 8, 6 (June 2009), 792–806. https://doi.org/10.1109/TMC.2009.51

[23] Ankit Mohan, Grace Woo, Shinsaku Hiura, Quinn Smithwick, and Ramesh Raskar. 2009. Bokode: Imperceptible Visual Tags for Camera Based Interaction from a Distance. In *ACM SIGGRAPH 2009 Papers*. ACM, New York, NY, USA, 98:1–98:8. https://doi.org/10.1145/1576246.1531404

[24] Ian Oakley, DoYoung Lee, MD. Rasel Islam, and Augusto Esteves. 2015. Beats: Tapping Gestures for Smart Watches. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1237–1246. https://doi.org/10.1145/2702123.2702226

[25] 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*. Springer, Berlin, Heidelberg, 200–207. https://doi.org/10.1007/978-3-540-39653-6_16

[26] Shwetak N. Patel, Jeffrey S. Pierce, and Gregory D. Abowd. 2004. A Gesture-Based Authentication Scheme for Untrusted Public Terminals. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 157–160. https://doi.org/10.1145/1029632.1029658

[27] Chunyi Peng, Guobin Shen, Yongguang Zhang, and Songwu Lu. 2009. Point&Connect: Intention-Based Device Pairing for Mobile Phone Users. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*. ACM, New York, NY, USA, 137–150. https://doi.org/10.1145/1555816.1555831

[28] Jun Rekimoto and Yuji Ayatsuka. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/354666.354667

[29] 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

[30] 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

[31] Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, and Hans Gellersen. 2010. PhoneTouch: A Technique for Direct Phone Interaction on Surfaces. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 13–16. https://doi.org/10.1145/1866029.1866034

[32] 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*. ACM, New York, NY, USA, 263–276. https://doi.org/10.1145/2462456.2464437

[33] Christine Szentgyorgyi and Edward Lank. 2007. Five-Key Text Input Using Rhythmic Mappings. In *Proceedings of the 9th International Conference on Multimodal Interfaces*. ACM, New York, NY, USA, 118–121. https://doi.org/10.1145/1322192.1322214

[34] 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

[35] Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. 2016. AmbiGaze: Direct Control of Ambient Devices by Gaze. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, New York, NY, USA, 812–817. https://doi.org/10.1145/2901790.2901867

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

[37] 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

[38] 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

[39] 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

[40] 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*. ACM, New York, NY, USA, 17–25. https://doi.org/10.1145/2659766.2659773

[41] 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

Received May 2018; revised August 2018; accepted October 2018