Dwell+: Multi-Level Mode Selection Using Vibrotactile Cues

Yi-Chi Liao* **Yen-Chiu Chen*** **Liwei Chan†** **Bing-Yu Chen‡**
*‡National Taiwan University †National Chiao Tung University
*{yichi,ballchen}@cmlab.csie.ntu.edu.tw †liweichan@cs.nctu.edu.tw ‡robin@ntu.edu.tw

ABSTRACT
We present *Dwell+*, a method that boosts the effectiveness of typical dwell selection by augmenting the passive dwell duration with active haptic ticks which promptly drives rapid switches of modes forward through the user's skin sensations. In this way, *Dwell+* enables multi-level dwell selection using rapid haptic ticks. To select a mode from a button, users dwell-touch the button until the mode of selection is haptically prompted.

Our haptic stimulation design consists of a short 10ms vibrotacile feedback that indicates a mode arriving and a break that separates consecutive modes. We first tested the effectiveness of 170ms, 150ms, 130ms, and 110ms intervals between modes for a 10-level selection. The results reveal that 3-beats-per-chunk rhythm design, *e.g.*, displaying longer 25ms vibrations initially for all three modes, could potentially achieve higher accuracy. The second study reveals significant improvement wherein a 94.5% accuracy was achieved for a 10-level Dwell+ selection using the 170ms interval with 3-beats-per-chunk design, and a 93.82% rate of accuracy using the more frequent 150ms interval with similar chunks for 5-level selection. The performance of conducting touch and receiving vibration from disparate hands was investigated for our final study to provide a wider range of usage. Our applications demonstrated implementing Dwell+ across interfaces, such as text input on a smartwatch, enhancing touch space for HMDs, boosting modalities of stylus-based tool selection, and extending the input vocabulary of physical interfaces.

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User Interfaces – Haptic I/O, Input devices and strategies.

Author Keywords
Dwell; Touch; Vibrotactile Feedback; Numerosity Perception; Haptically-augmented Input; Touchscreen; Input Modality; Finger; Smartwatch; Stylus; Experiment.

INTRODUCTION
Dwell, also known as *touch-and-hold* and *press-and-hold*, is a well-known input method for mode switching, commonly found in early mouse-clicking [29] and pen-based interaction [31], and more recently also in touchscreens *e.g.*, mobile phones and smartwatches. Holding a touch on screen for an extended period of, for example, 1 second, activates the mode switching, *e.g.*, invoke a marking menu. Dwell, however, is generally considered not effective due to the inevitable long dwell period in order to avoid unintended input and the limitation of only a single mode.

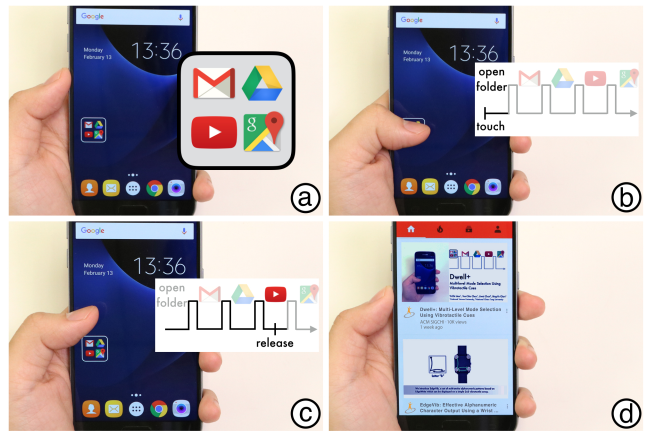

Figure 1. The interaction process of *Dwell+*. (a) A folder contains several Google apps is placed in the background. (b) The user touch-and-presses the folder icon to activate Dwell+ selection. (b) The user releases the icon after receiving the second vibration which corresponds to the Youtube app. (d) Youtube opens after the selection.

In past research, modes have been increased via various modalites of a touch. *Two-step mode switching* determines a mode based upon subsequent finger motions, such as rolling [30], shear direction [11], and in-air motion [8]. *Single-tap mode switching* has been further improved in regards to time- and space-efficiency by exploiting immediate modalities derived from tapping pressure [13], contact points in the touch finger [12, 17], and fingerprints [33].

While the aforementioned approaches are effective in creating modes, they suffer from (1) requiring users to learn and memorize *extra motions other than simple tapping*, *e.g.*, tapping with certain pressures, postures and motion gestures, as well as from (2) requiring *extra sensors* for a touch interface to be enabled.

Dwell+
This paper introduces *Dwell+*, a multi-level dwell selection method enabled by augmenting dwelling with rapid haptic ticks. These ticks utilized through the user's skin sensations indicate the switches of modes, enabling users to directly access a mode using a single touch with varied dwelling durations. Compared to previous works that enhance input modal-

UIST 2017, October 22–25, 2017, Quebec City, QC, Canada

ACM ISBN 978-1-4503-4981-9/17/10$15.00.
https://doi.org/10.1145/3126594.3126627

ity by performing different postures or motions, Dwell+ requires minimal motor space, *i.e.*, a single touch. Thus, Dwell+ can even be performed on devices with single-touch point sensing, allowing it to be applied to existing button-input devices.

Figure 1 demonstrates the activation of an in-folder app using Dwell+. Here, after users folding touch on the folder button, a first vibration comes in at 150ms. Releasing the button before the first vibration happens activates a normal touch, i.e., unfolding the folder, and releasing it after the nth vibration arrives activates the nth app in the folder. For instance, the user activates the gmail app at a second slot by releasing the touch after the first haptic tick has been perceived.

Unlike typical dwell selections which passively await touch to activate a mode switch, Dwell+ actively drives fast switches of modes with rapid haptic ticks. Our implementation adopted vibrotactile feedback to deliver the haptic ticks. Owing to the acute sensitivity of the human finger to vibrations, a 10 millisecond vibrotactile feedback is sufficient for users to pick up isolated haptic ticks. Consecutive ticks are separated by breaks whose lengths determine the efficiency of Dwell+ interactions. Shorter breaks benefit faster switches but may erroneously activate the next mode when users release their touch too late, and may require a great cognitive load.

To explore the design space of Dwell+, the following factors were determined through a series of user studies: (1) the effective lengths of breaks to separate consecutive haptic ticks, (2) the number of effective modes that can be enabled with monotonous haptic ticks, (3) whether a rhythmical design helps boost the number of effective modes. Our user studies revealed that 170ms-, 150ms-, 130ms-, and 110ms-long breaks are generally effective for 4-level mode selection, and an improved design using chunks further increased the performance up to 10 levels for 170ms- and 5 levels for 150ms-long breaks.

The contributions of this work are three-fold: (1) the development of Dwell+, a novel approach that enables multi-level dwell selection by augmenting dwelling with haptic ticks; (2) user studies to identify effective designs for Dwell+ with mono- and chunking-based ticks; (3) a set of applications to demonstrate the applicability of Dwell+ in contexts from vibration-enabled touchscreen devices to non-vibrating touch interfaces.

RELATED WORK

In this section, we review previous research on dwell selection, enhancing input modality of touch interaction, and numerosity perception for switch access scanning as well as using vibrotactile feedback as notification.

Dwell Selection

Dwell, also known as holding, long-press, press-and-hold and touch-and-hold, is a mode-switching method based on time. In early time, Kurtenbach *et al.* [22] first proposed to invoke a marking menus by holding a stylus on screen for 1/3 second. Pook *et al.* [29] introduced the same function with long-press using mouse's right button. As such, dwell is commonly used to provide 2-mode operations in stylus input [25, 31], *e.g.*, "draw and command" and "inking and gesturing".

Dwell select is also found in touchscreens. For instance, dwelling touch allows a user to activate dragging actions or menu invocations via touch interface. Also, researchers have introduced two-fingered performance of hold-and-move gestures to reduce dwelling times before dragging objects [20], and undoing dwell-based direct manipulation, such as moving and resizing a window, via a spring design [1]. In natural interaction such as eye-gazing [18, 19, 35] and body motion input (*e.g.*, kinect), dwell was commonly adopted as commitment gesture.

Though dwell selection is easy-to-use and highly adaptable to most touch-based interactions, as aforementioned, it is generally considered ineffective due to its limitations. Thus, Dwell+ which is based upon dwell, enhances effectiveness both in terms of input space and time with haptic ticks that proactively fast drive multi-level dwell selections.

Exploring Finger-touch Modalities

Many previous works have attempted to increase input modality by allowing for discernment of different touch features. *Single-tap mode switching*, for example, utilizes different parts of the finger [12, 17], different levels of pressure [13, 36], the area of the finger contact [5] or different fingers [15, 33] to activate a number of modes at the instance of finger contact. On the other hand, *two-step mode switching* encodes modes with motions that come before or after the landing touch. For example, rolling [4, 30], swiping [7] or shearing direction [11], motion in air [8], and motion of the device [14] have been explored to expand input space on both smartphones and smartwatches.

The aforementioned techniques all take into account some touch features other than the fact of touch itself. These operations require users to learn various gestures and underlying mapping. In comparison, Dwell+ is based on simply touch and it actively prompts haptic-switches of modes to users. In addition, Dwell+ requires lesser effort to perform than said techniques.

Numerosity Perception for Switch Access Scanning

Previous research has investigated the limitation of numerosity perception. Lechelt *et al.* [23] tested numerosity perception and compared subjects performance among visual, auditory and haptic channels. Ternes *et al.* [34] and Pasquero *et al.* [28] experimented with human perception of temporal tactile messages. Regarding applications, Bianchi *et al.* [3] adapt numerosity sense for secret digit-input and Kuribara *et al.* [21] used tactile perception for users to secretly enter a pin-code through counting vibrations.

Based on previous works, Dwell+ explores a combination of tactile numerosity perception and hand motion. Furthermore, it uses numerosity perception, *i.e.*, counting vibrations, to enable an indirect, invisible and efficient *Switch-Access-Scanning*[1] [2, 32] mode selection.

[1] `https://en.wikipedia.org/wiki/Switch_access_scanning`

Vibrotactile Feedback as Notification

Finally, vibrotactile feedback has been extensively explored as an intimate and private communication channel to display notifications through temporal patterns [6] and spatial patterns such as direction [24] and alphanumeric letters [26]. Like these works, Dwell+ also utilizes vibrotactile feedback as notification, but uses it to improve users' awareness of dwelling time, resulting in a performance boost in multi-level dwell selections with single touch.

STUDY OVERVIEW

User Study 1 discussed herein measures users' ability in coordination of skin perceptions and motor controls to perform Dwell+ operations associated with vibration intervals of various lengths. The results of the study indicate effective designs for Dwell+ to use *monotonous vibrations*.

Single haptic tick involves a 10ms vibration, which is so short that it is nearly impossible for visual or audio observation. Four candidate intervals (170, 150, 130, 110ms), and two common gestures (*thumb-input* for single-handed usage and *index-finger-input* for two-handed situations) were included. Among all candidate time-intervals, the 170ms interval resulted in the highest accuracies across 10-level selection, being 88.44% accurate (s=11.94%) on average for thumb input, and 90.75% accurate (s=13.79%) for index finger input, respectively. In addition, for all the intervals, locating targets within 3 vibrations showed significantly higher accuracy than those in higher levels, indicating that the effective levels of monotonous vibrations are within three vibrations.

To improve user performance, we propose designing Dwell+ vibrations with chunking. Here, chunking [27] refers to a mechanism that facilitates human recognition by binding individual pieces of information together. To embody chunking in the vibrations, we applied longer vibrations at the first of every three vibrations to generate a 3-beats-per-chunk rhythm. *User Study 2* evaluates the chunking design showing a significant improvement of around 94.5% accuracy for 10-level selection with the 170ms interval and 93.82% accuracy for 5-level selection with the 150ms interval.

Finally, *User Study 3* further investigated using Dwell+ under conditions where the occurrence of touches and vibrations are disparate. The study instructed users to perform touch with one hand, while perceiving vibrations through a separate smartphone in the other hand. The results indicate comparable performance to User Study 2, which can shed light on wider usage scenarios, such as augmenting physical buttons with Dwell+ through a vibrating smartwatch on the wrist or a smartphone in the user's pocket.

USER STUDY 1: BASELINE PERFORMANCE AND EFFECTIVE LEVEL OF MONOTONOUS-DWELL+

The goal of this study was to measure the effectiveness of monotonous Dwell+ configured with four time-intervals (170ms, 150ms, 130ms, and 110ms) and 10ms-vibration for 10-level selection (Figure 2). The four candidate intervals were determined from a 6-participant pilot test which shows that less-than-110ms intervals are too fast to enable selection at any target vibration. It also shows that more-than-170ms intervals leads to a sense of inefficiency for most users.

Moreover, since human beings have a limited cognitive load, the difficulties of counting vibrations increases as the amount of vibrations accumulate. Based on the performance of mono-vibration Dwell+, this study is also meant to derive an effective level with highest accuracies as to the design guidelines for chunking vibrations that potentially achieves better accuracy for Dwell+ selections.

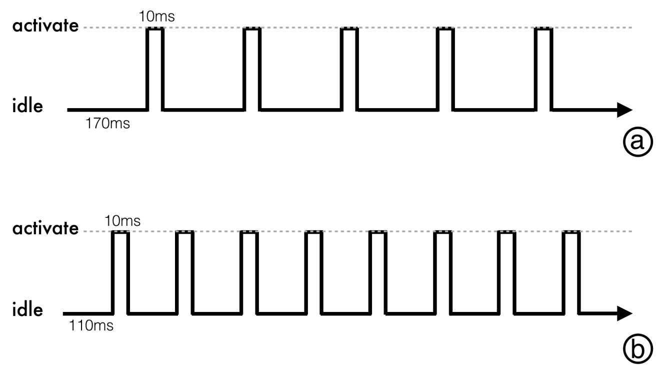

Figure 2. Two selected vibration patterns of intervals within one second: (a) 170ms interval between each 10ms-vibration which generates 6 modes within a second, and (b) 110ms interval between each vibration that delivers 9 modes within the a second.

Study Design

To prevent the transfer effects between intervals inherent in within-subjects design, we employed a between-subjects 2×10 factorial design. The independent variables were INTERVAL (110ms, 130ms, 150ms, 170ms) and TARGET VIBRATION (selecting a target of vibration 0 to 9). 32 participants were randomly assigned to one of the intervals, so that there were 8 participants per interval, and the experiment for each participant contained 4 blocks of trials.

In addition, two common gestures (Figure 3), *i.e.*, tapping with thumb and tapping with index finger, were both tested within each block and the results are discussed separately. There were 50 random dwell-selection trials (5 rounds of 0 to 9 vibrations) for each gesture in a block. For counterbalancing, in every block of every interval, 4 participants were asked to perform thumb-input first and then perform the index-finger-input, while the other 4 participants were asked to perform index-finger-input first then thumb-input.

In summary, the experimental design is: *4 intervals × 8 participants per interval × 4 blocks per participant × 2 gestures per block × 50 trials per gesture = 12,800 data points.*

Apparatus

Figure 4 illustrates the study interface. There is only one button on the screen. A digit between 0 and 9 is displayed with the text "Target" indicates the TARGET VIBRATION of the current trial. In every trial, once the participant had pressed the button, a series of 10ms vibrations divided by the interval of the condition were generated until the button was released. Then, the actual count of the vibrations generated during the time pressed was displayed with the text "Answer"

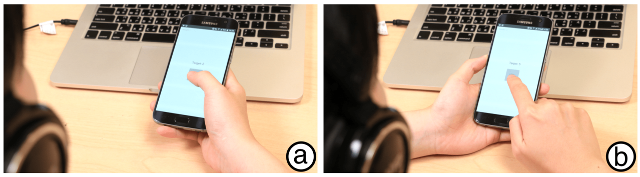

Figure 3. Two common gestures for input on touchscreen: (a) thumb-input in one-handed interaction, and (b) index-finger-input in two-handed interaction, were tested in User Study 1.

(Figure 4(b,c)) to complete the trial. If the actual vibration met the targeted vibration, the trial was recorded as a success; otherwise, it was recorded as a fail. The study interface was implemented on the Samsung Galaxy S7 smartphone model.

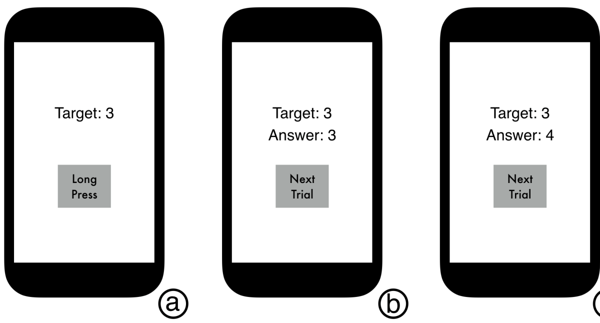

Figure 4. The illustration of our study interface. (a) The interface before user pressing the button. (b) A successful trial in which the actual vibration count meets the targeted vibration. (c) A failing trial in which the user has pressed the button longer than target.

Participants

We recruited 32 participants (15 female, age from 22 to 25) from National Taiwan University.

Tasks and Procedures

There were four testing blocks; each block contained two gestures, *thumb-input* and *index-finger-input*. To familiarize participants with the testing procedure, a training block was provided. Each block took about 4 minutes, and there were 3-minute breaks between blocks. In total, each participant took less than 35 minutes to complete the study. To prevent participants from receiving vibrations by counting the vibration sounds, they were asked to wear a headset emitting pink noise.

Results

The overall accuracies of *thumb-input* gesture are 88.44% (s=13.8%), 75.05% (s=20.9%), 72.5% (s=19.7%), and 68.4% (s=19.4%) for the 170ms, 150ms, 130ms, and 110ms intervals, respectively (Figure 5), and the accuracies of *index-finger-input* are 90.75% (s=12%), 78.5% (s=19.1%), 75.5% (s=19.2%), and 70.25% (s=19.5%) for the 170ms, 150ms, 130ms, and 110ms intervals, respectively (Figure 6).

To derive an effective level, within which accuracy is retained, under all conditions, we categorized the intervals into groups of similar accuracy, and investigated the target vibrations with the highest accuracy for each group. All the data for analysis was tested for normality and homogeneity of variance.

Categorizing Intervals Based on Accuracy

Both usages of the two gestures showed *no learning effects* during the process by two-way (block×target vibration) ANOVAs, which showed no interaction between factors, and no differences between the four blocks (p>0.05). We then aggregated the accuracies across blocks and examined two gestures separately by running two 4×10 (intervals×targeted vibrations) two-way ANOVAs.

The results show no interaction between INTERVAL and TARGET VIBRATION for the usage of the two gestures (thumb: $F_{27,280}$=0.79, p>0.05; index-finger: $F_{27,280}$=0.94, p>0.05). Significant differences between INTERVALS were also found in main effects analysis for both gestures (thumb: $F_{3,280}$=28.85, p<0.001; index-finger: $F_{3,280}$=31.41, p<0.001), and post hoc Tukey HSD suggests that 170ms interval has significantly higher accuracy than all the other intervals, *i.e.*, 150ms, 130ms and 110ms (p<0.01).

Based on the differences between intervals, the data was arranged into four groups: *a) 170ms thumb, b) 170ms index finger, c) 150ms, 130ms, 110ms thumb, and d) 150ms, 130ms, 110ms index finger*, and each group was then discussed separately.

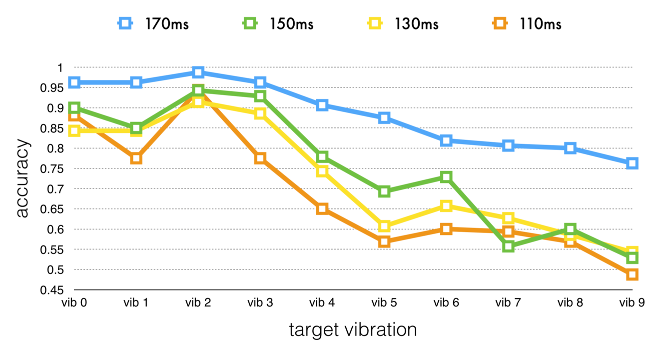

Figure 5. The performance of Dwell+ of User Study 1 for the thumb-input gesture.

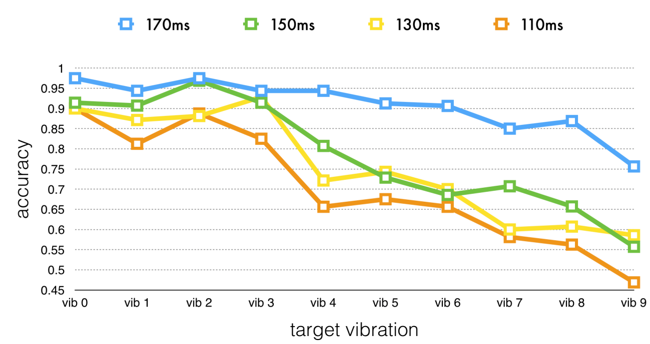

Figure 6. The performance of Dwell+ of User Study 1 for the index-finger-input gesture.

Deriving Effective Level of Each Group

Here, we attempt to determine the effective level of the four different groups by analysis of the effects of targeted vibrations on accuracy.

a) 170ms thumb: We ran one-way ANOVA to examine the effects of TARGET VIBRATION on accuracy. There is a significant difference between different vibrations in thumb-input gesture ($F_{9,70}$=3.821, p<0.01), and the Tukey HSD post hoc test shows that vibrations 0 to 3 have the highest accuracies (p<0.05).

b) 170ms index-finger: One-way ANOVA shows a significant difference between vibrations ($F_{9,70}$=3.173, p<0.01). The Tukey HSD post hoc test shows that vibrations 0 to 4 have the highest accuracies among all (p<0.05).

c) 150ms, 130ms and 110ms thumb: 3×10 (interval×target vibration) two-way ANOVA shows no interaction between INTERVAL and TARGET VIBRATION ($F_{18,210}$=0.462, p>0.05). Main effects analysis showed differences between vibrations ($F_{9,210}$=20, p<0.001), and vibration 0 to 3 have higher accuracies.

d) 150ms, 130ms and 110ms index-finger: 3×10 (interval×target vibration) two-way ANOVA shows no interaction between factors on accuracy ($F_{18,210}$=0.305, p>0.05). Significant differences between the TARGETED VIBRATIONS are found ($F_{9,210}$=18.9, p<0.001), and target vibration 0 to 3 have significantly higher accuracies (p<0.05).

Summarizing the aforementioned results, accuracies of most intervals retain their highest between target vibrations 0 to 3, and drop significantly after this level. The general effective level of mono-Dwell+ is determined to be within three vibrations.

Comparing Accuracies between Two Gestures

Finally, we ran 2×10 (gesture×target vibration) two-way repeated measures ANOVAs on each interval to test if there were differences between the usages of gestures on accuracy. Results show no interaction between two factors, and also no differences between the gestures for all intervals (p>0.05).

Discussion

Given these results, Dwell+ allows selection with higher accuracies under an effective level, *i.e.*, vibrations 0 to 3. Exceeding this level, user accuracy to select a certain level drops as the target vibrations grow higher (*e.g.*, with higher target level), which constrains the usage of Dwell+ in scenarios requiring more modes. Intuitively, a solution to this limitation is to extend the interval between vibrations; as suggested, a longer interval achieves higher accuracy. Unfortunately, even the accuracies of 170ms interval drops to less that 90% when the target vibration comes up to 5. To support even more levels, the interval must be much extended which results in another severe problem, *low time-efficiency*. Particularly, a higher target vibration suffers from a longer dwell period. This leads to an important question in the development of Dwell+: *To what extent can we provide higher-level selection without suffering low time-efficiency?*

Our solution is *organizing a series of vibrations into several three-beat chunks*. Since accuracy is retained when counting less than three vibrations, the chunking pattern allows the participants to recognize every vibration as a combination of chunks and beats. The implementation of chunking design is explained just hereafter.

USING 3-BEAT-PER-CHUNK DESIGN FOR ENHANCING THE EFFECTIVE LEVEL OF DWELL+

To extend the effective level, we propose presenting a sequence of vibrations with a *3-beats-per-chunk rhythm.*

Figure 7 illustrates the difference between a monotonous design and a 3-beats-per-chunk design. In the 3-beats-per-chunk design, the first beat of every three vibrations carries a longer vibrating duration, *i.e.*, 25ms, to present a more discernible haptic cue. As such, every vibration can be recognized as a combination of chunks and beats. For instance, a fifth vibration can be recognized as the second vibration in the second chunk, as opposed to the fifth beat. The length of long vibration utilized, 25ms, was determined through a 6-participant pilot test which shows that participants can clearly differentiate between 25ms and 10ms vibrations.

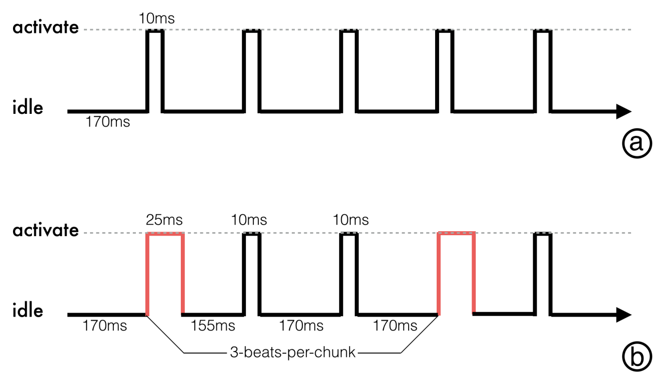

Figure 7. Illustration of 3-beats-per-chunk design applied to the original 170ms interval. Different from the original mono-vibration (a), the modified version (b) generates a long vibrating duration to the first of every three vibrations.

In addition, to keep an identical length across vibration cycles, the gap following the 25ms long vibration is set at 15ms (*i.e.*, 25ms minus 10ms) shorter than the original intervals. Hence, the duration of shorter gaps per the original 170ms, 150ms, 130ms and 110ms intervals are 155ms, 135ms, 115ms and 95ms, respectively.

USER STUDY 2: PERFORMANCE OF CHUNKING-DWELL+ AND THE OPTIMAL INTERVALS

This study was to evaluate the performance of *3-beats-per-chunk design* with the same time-interval sets. Furthermore, it is intended to assist in determining an optimal interval based on the results.

Study Design and Apparatus

The study design and apparatus were same as User Study 1 except the 3-beats-per-chunk design was applied. The experimental design was: *4 intervals × 8 participants per interval × 4 blocks per participant × 2 gestures per block × 50 trials per gesture = 12,800 data points.*

Participants

We recruited 32 participants (14 female, age from 21 to 25) for this study from our university. Participants were assigned evenly and randomly into four groups of different intervals.

Tasks and Procedures

The tasks and procedures are same as User Study 1 except for the introduction of our 3-beats-per-chunk design as discussed herein prior.

Results

The accuracies of the usage of the *thumb-input gesture* are 94.47% (s=5.6%), 88.44% (s=11.4%), 79.62% (s=14.3%), and 72.72% (s=20.9%) for the 170ms, 150ms, 130ms, and 110ms intervals, respectively (Figure 8), and the accuracies for the *index-finger input* gesture are 94.53% (s=6.6%), 89.25% (s=12.1%), 83.06% (s=13.6%), and 73.0% (s=22%) for the 170ms, 150ms, 130ms, and 110ms intervals, respectively (Figure 9). The results are analyzed in two parts: a) deriving the optimal interval for Dwell+ using chunking vibration, and b) comparing the accuracy between the use of mono-Dwell+ and chunking-Dwell+. All the data given hereafter has been tested for normality and homogeneity of variance.

Deriving the Optimal Interval by Accuracy

Owing to no differences between the four blocks ($p>0.05$), we aggregated the accuracies across blocks and examined two gestures separately by running two 4×10 (intervals×targeted vibrations) two-way ANOVAs.

Performances of the two gestures again reveal similar results. Significant effects between INTERVALS are found for both thumb ($F_{3,280}$=49, $p<0.001$) and index-finger ($F_{3,280}$=43.58, $p<0.001$). For both gestures, the Tukey HSD post hoc test shows that greater intervals have higher accuracy than the shorter ones ($p<0.05$), *i.e.*, accuracy of 170ms interval >150ms >130ms >110ms.

Breaking the results down further and looking at the target vibrations, we see the differences between intervals are generally greater in higher-level selections, *i.e.*, vibration 4 to 9, than those in lower-level selections, *i.e.*, vibrations 0 to 3. Take the 170ms and 110ms intervals with thumb-input gesture as an example. The differences in accuracy are greater at target vibration 9 (91.2% vs 58.1%) than that at target vibration 2 (98.1% vs 93.1%). The results indicate that greater intervals benefit more from chunking design for the higher levels. This finding also explains the significant interaction between INTERVAL and TARGET VIBRATION on accuracy for both thumb ($F_{27,280}$=7.492, $p<0.05$) and index-finger gestures ($F_{27,280}$=7.236, $p<0.05$).

The 170ms and 150ms intervals offer significantly higher and more reliable accuracies; thus, we decided upon these two intervals as the optimal intervals for the *chunking design*. The following analysis discusses the improvement from mono-Dwell+ to chunking-Dwell+ regarding the two intervals only.

Comparing the Performance between Mono-Dwell+ and Chunking-Dwell+

First, the data on the usage of the two gestures of the two optimal intervals were aggregated running two 2×10 (gestures×target vibration) two-way repeated measure ANOVAs on the 170ms and 150ms intervals, respectively. Since there was no interaction between the two factors for both intervals (170ms: $F_{9,63}$=0.53, $p>0.05$; 150ms: $F_{9,63}$=2.974, $p>0.05$), and also no differences between the usage of the gestures (170ms: $F_{1,63}$=3.08, $p>0.05$; 150ms: $F_{9,63}$=0.209, $p>0.05$), we can aggregate the data of two gestures for both intervals for further analysis.

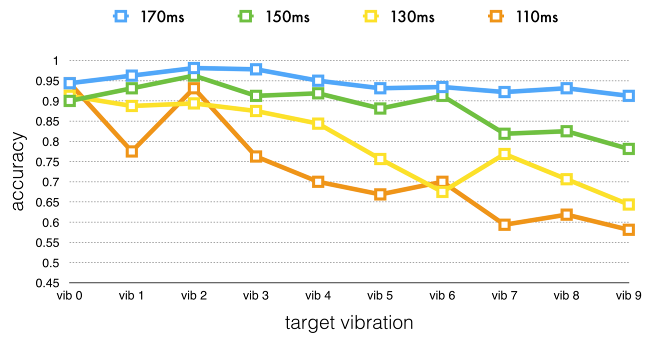

Figure 8. The performance of Dwell+ with chunk design in User Study 2 for usage of the thumb-input gesture.

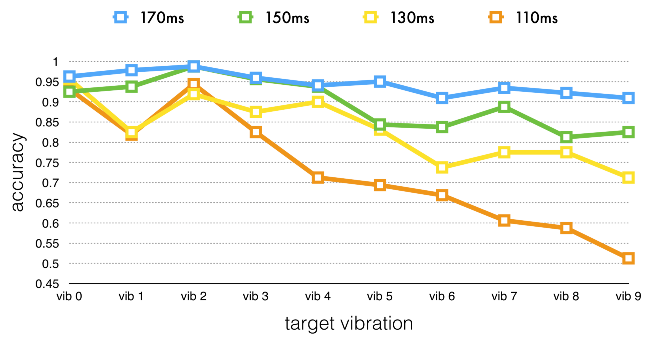

Figure 9. The performance of Dwell+ with chunk design of User Study 2 for usage of the index-finger-input gesture.

We compared the aggregated accuracy of the two intervals to the mono-Dwell+ by 2×10 (types of vibration×target vibration) two-way ANOVAs. Results show chunking-Dwell+ has significantly higher accuracy than the mono-Dwell+ for both the 170ms interval ($F_{1,300}$=24, $p<0.001$) (Figure 10) and 150ms interval ($F_{9,300}$=65.1, $p<0.01$) (Figure 11).

The differences between chunking-Dwell+ and mono-Dwell+ are mainly found in the higher-level selection, *i.e.*, target vibration 5 to 9 for 170ms, and vibration 4 to 9 for 150ms intervals, which explains the interaction between VIBRATION TYPE and TARGET VIBRATION (170ms: $F_{9,300}$=2.69, $p<0.01$; 150ms: $F_{9,300}$=3.75, $p<0.01$)

Discussion

This 3-beats-per-chunk design has proved to successfully *extend the effective level of Dwell+*, where the differences between mono- and chunking-vibration are mainly in the higher-level selection. Also, we found that the greater intervals have higher accuracies due to the chunking design.

The two greater intervals in our study, *i.e.*, 170ms (94.5%) and 150ms (88.8%), offer more reliable performance; thus, we decided to use these intervals with chunks as the optimal design. In general, the 170ms-chunking version provides higher accuracy and more effective modes, so it is suitable

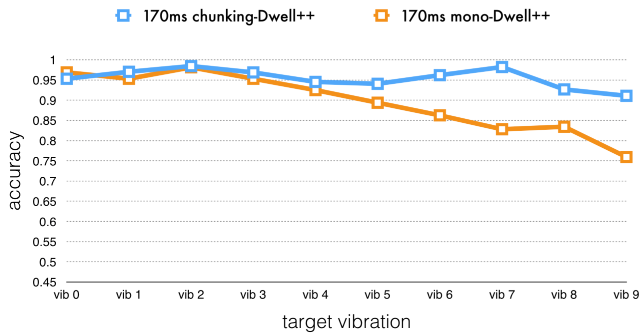

Figure 10. Overall comparison of the chunking and original (mono-) version of Dwell+ using the 170ms time interval.

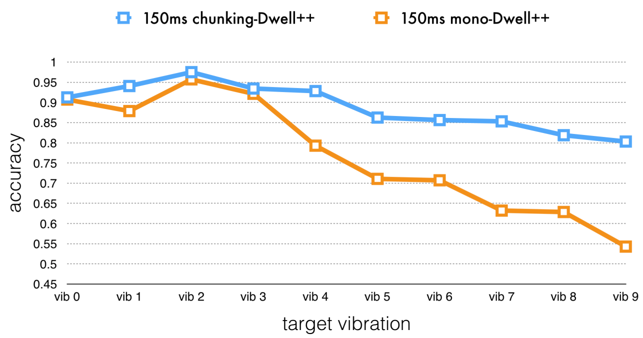

Figure 11. Overall comparison of the chunking and original (mono-) version of Dwell+ using the 150ms time interval.

for interfaces requiring greater input space such as performing digit input by a single tapping. Though 150ms-chunking has worse performance at higher levels, the accuracy of 5-level selection is 93.82% and it benefits from its low time-efficiency; thus, it might be desired for fast-selection such as for direct-launch of apps inside a folder.

USER STUDY 3: CHUNKING-DWELL+ WITH DISPARATE TOUCH AND VIBRATION FUNCTIONS

After deriving the optimal design for vibration, this user study investigated an alternative version of Dwell+, which separates touch interaction from where the vibration is perceived, *i.e.*, the vibration is emitted from a different device that the finger touches. Such variation would enable many practical usages of Dwell+ as demonstrated in the applications, a user could sense the vibration and perform Dwell+ touch using a stylus with the opposite hand on a drawing app (Figure 17(b)). Therefore, this study attempts to answer *whether user accuracy remains under such a condition where touch and vibration are not co-located.*

Study Design and Apparatus

The participants were asked to hold the smartphone displaying the derived Dwell+ pattern, *e.g.*, 150ms- or 170ms- interval 3-beat per chunk of vibrations in their left hands. Unlike previous studies, the participants do not touch the button on the screen, but a button set on the desk enabled with a touch-sensitive copper foil (Figure 12). When triggered, the button activates the smartphone in the left hands of the participants to emit vibrations.

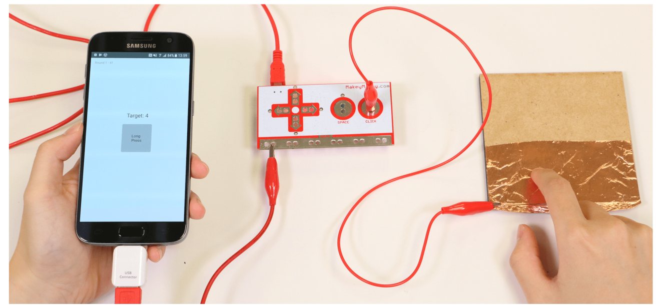

Figure 12. The apparatus used for User Study 3 with the participant touching a copper foil, which is connected to the smartphone through a Makey Makey board.

This apparatus is similar to that used in User Study 2. The only difference is that the handset is connected to touch-sensitive copper foil through a Makey Makey board[2].

Since only the 170ms and 150ms intervals are used for the index-finger-input gesture in this study, the study design is: *2 intervals* $\times$ *8 participants per interval* $\times$ *4 blocks per participant* $\times$ *1 gestures per block* $\times$ *50 trials per gesture = 3,200 data points.*

Participants

16 participants were recruited (10 female, age from 22 to 24) from our university.

Tasks and Procedures

The tasks and procedures are the same as those used in User Study 2 except that participants were asked to touch the desk button instead of the screen button.

Results

We aggregated the accuracies across blocks because there were no differences between them. The overall accuracies are 94.1% (s=6.1%) and 85.7% (s=11.8%) for the 170ms and 150ms intervals, respectively. After aggregating the data from the two gestures from User Study 2, and all the data was tested for normality and homogeneity of variance. We compared the accuracy of *two-handed disparate touch and vibration condition* and *touch-on-screen condition* by 2$\times$10 (conditions$\times$target vibrations) two-way ANOVAs.

Figure 13 and Figure 14 show the performance of 170ms and 150ms intervals of two conditions, respectively. Results show no interaction between CONDITION and TARGET VIBRATION for two both the 170ms ($F_{9,220}$=0.7, p>0.05) and the 150ms ($F_{9,220}$=0.54, p>0.05) intervals, and also no significant differences between two CONDITIONS for both the 170ms ($F_{1,220}$=0.204, p>0.05) and the 150ms intervals ($F_{9,220}$=0.221, p>0.05).

Discussion

This study shows that *two-handed disparate touch and vibration condition* works as touching on the screen which opens the possibilities of performing Dwell+ on non-vibrating interfaces, *e.g.*, keyboards, touchpads, physical buttons, etc., by

[2] `http://www.makeymakey.com`

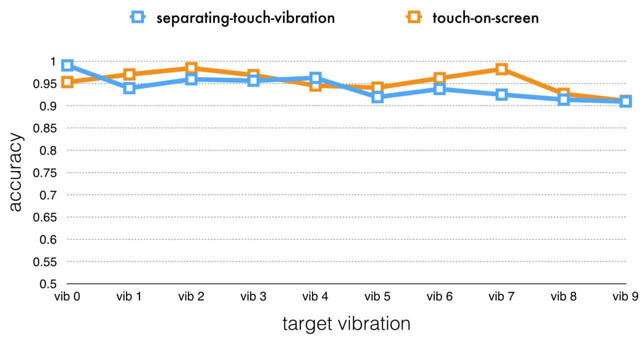

Figure 13. The performance of Dwell+ using 170ms interval with chunks in two conditions.

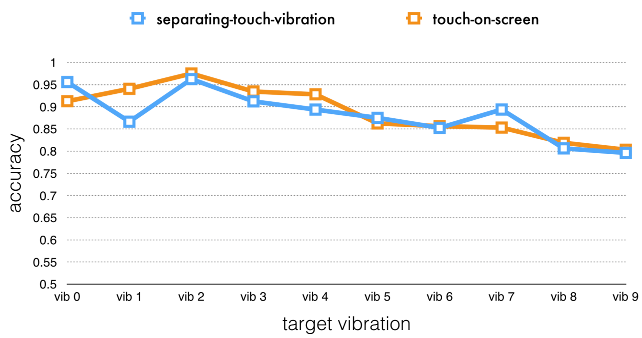

Figure 14. The performance of Dwell+ using 150ms interval with chunks in two conditions.

using wearable devices, *e.g.*, rings, watches, to deliver vibrations.

Due to the fact that different body regions have different levels of tactile sensitivity [9, 10, 16], different vibration parameters (*e.g.*, duration, intensity, amplitude) are required to clearly deliver the tactile cues to the various body regions. Hence, this study was an initial investigation taking this factor into account, and the performance and derived interval cannot be directly applied to other areas of the skin.

EXAMPLE APPLICATIONS

Owing to its simple extension to dwell select, Dwell+ can be easily integrated with any touch interfaces. To demonstrate the applicability of Dwell+, we implemented two categories of examples, adding Dwell+ to many previously explored touch interfaces with and without vibration capability.

Dwell+ on Vibration-Enabled Touchscreens

Touchscreens can be found in everyday interaction now. Most touchscreen devices have built-in vibrotactile actuators, thus they are ready to be utilized by Dwell+.

Direct-Launching In-Folder Apps

Dwell+ allows users to directly launch an app encapsulated in a folder, as shown in Figure 15(a). In this example, a folder button shows a preview of 2×2 apps in the folder. With a simple tapping, *i.e.*, no dwelling, the user can open the folder to see all apps as per the function of the current regular interfaces (Figure 15(c)). To activate a certain app on the 2×2 preview with Dwell+, the user can touch-and-dwell the button until the corresponding vibration, as indicated in Figure 15(b), is emitted and perceived. For example, if a user wants to open the second app, he can contact the icon and wait for the second vibration and release. In this application, since it requires only 5 modes, *i.e.*, vibrations 0 to 4, we implemented the vibrations using the 150ms interval with chunks for higher efficiency on a Samsung Galaxy S7.

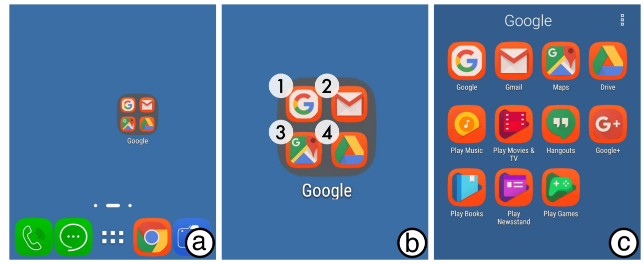

Figure 15. Using Dwell+ to fast-select apps inside a folder: (a) A regular folder icon with a preview of 2×2 apps in the folder. (b) The corresponding target vibration for each app. (c) Simple tapping would open the folder to see all apps as per regular use.

Supporting Single-Handed Typing

Typical qwerty keyboard on a smartphone is not easy for single-handed typing especially when its screen size becomes larger. Using Dwell+, we can rearrange the keys into a reachable range for the thumb by providing a simple group keypad based on the T9 keyboard layout as shown in Figure 16(a). Since each button is comprised of only 3 to 4 characters, this application uses the 150ms interval for quick typing and is also implemented on a Samsung Galaxy S7 with a 5.1 touchscreen.

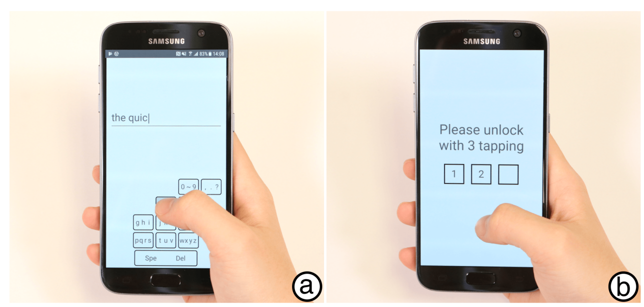

Figure 16. (a) A Dwell+ keyboard for easy single-handed typing. (b) Unlocking the device with three Dwell+ digit input.

Unlocking by Subtle Tapping

Typing passwords or drawing patterns are common methods to unlock personal devices; however, these methods require detectable gestures leading to lower privacy and safety. We implemented a subtle unlock method on a Samsung Galaxy S7 with Dwell+ as shown in Figure 16(b). With this method, every touch activates the Dwell+ selection as a private digit input. For example, a simple tap types digit 0, and a touch-and-dwell up to 3 vibrations inputs digit 3. Such an unlock method takes only subtle taps making them difficult to detect.

Entering Texts on Small Screens

Dwell+ allows users to perform text entry on small screens, like smartwatches, by providing a simple group keypad as shown in Figure 17(a). Each group contains at most four alphabetical letters (or comma, period, space, and delete) allowing for direct accessibility via a single dwell touch. We implemented the Dwell+-keyboard on an Asus ZenWatch 2 with a 1.63 touchscreen. The bigger group button also helps to reduce touch-landing errors while performing text-entry on the move. The vibrations of this application and the following watch-vibrated applications were mainly displayed on the wrist, which has weaker tactile sensitivity than that on the palm. Hence, we set a longer vibrating duration, 30ms, and a longer interval, 180ms, to ensure clear recognition.

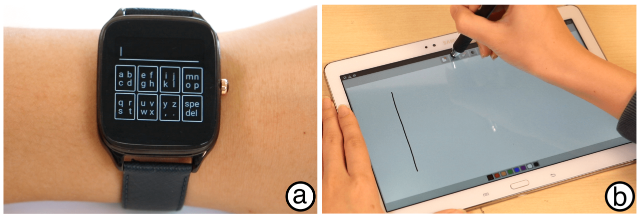

Figure 17. (a) Small-screen text input enabled by Dwell+; each button contains a set of characters which can be selected by a touch. (b) Dwell+ enables fast tool-changing with a stylus on a tablet.

Boosting the Stylus-Based Interaction

Some tablets or smartphones are supporting stylus input nowadays. Though most of such devices are vibration-enabled, as we mentioned prior in the motivation of User Study 3, users cannot receive the vibration well through the stylus. Taking a drawing application as an example as shown in Figure 17(b), it usually requires two steps for selecting tools with different levels. Dwell+ can boost efficiency by transferring the second-step selection into Dwell+ modes. This application was built on a Samsung Galaxy Note 10.1 tablet, and the user perceives the haptic ticks through his left hand holding the tablet.

Dwell+ on Non-Vibrating Touch Interfaces

Unlike touchscreen devices, many touchable objects almost have no built-in haptic output. To enable Dwell+ on those objects in our living environment, we can either 1) embed haptic actuators (*e.g.*, vibrotactile motors) into them, or 2) let users wear or hold haptic actuators, such as wearing a smartwatch or simply holding a smartphone in hand.

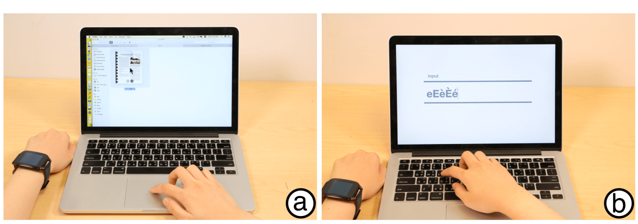

Figure 18. A user can use Dwell+ for fast mode-switching while using a laptop by either inputting through the (a) trackpad or (b) keyboard. The vibration is generated from a smartwatch.

Providing Mode-Switching for Laptops

Mode-switching is essential when we manipulate a file with a trackpad of a laptop by performing complicated hand gestures. For example, using an Apple MacBook, a single click selects a file, double clicks open the file, two-finger click activates a hidden menu for more functions, etc. In addition to the trackpad, when using the keyboard for typing, there are also several modes to switch through via function keys. For example, pressing caps lock for typing capital letters or lowercase ones. Using Dwell+, users can switch the modes by simply touching the trackpad or typing the keyboard as shown in Figure 18. Because the laptop itself has no built-in haptic output, the vibration is delivered through a smartwatch.

Enhancing Touch Space for HMDs

Some HMDs (Head-Mounted Displays) like Google Glass use a simple touch bar as the main input device and some simple solutions like Google Cardboard even have no input mechanism. For enhancing the input space of such devices, we implemented a prototype by attaching a vibration motor[3] on a Google Glass to envision Dwell+ to support it as shown in Figure 19. Since the ERM motor requires slightly longer rising time, the vibration is set to be 30ms and the interval between is 180ms with 3-beats-per-chunks. This implementation can even be upgraded by combining a physical button with the motor to generally support simple devices.

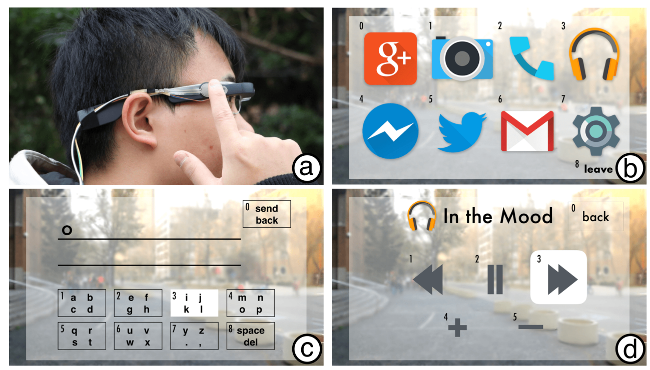

Figure 19. Using Dwell+ to augment the touch space of Google Glass. (a) A user touches the touch bar while receiving the Dwell+ vibration to (b) select apps, (c) input texts, or (d) control the music player.

Controlling IoT Devices

Similar to the simple HMDs, many IoT (Internet of Things) devices have many functions but users tend to control them as simply as possible, *e.g.*, by one button. Hence, Dwell+ can be used to augment the button to provide more input space for the devices. Figure 20 shows an example of Philips Hue, which is a lamp with different colors. By providing the Dwell+ vibrations through a smartwatch, the lamp can be controlled by just one button.

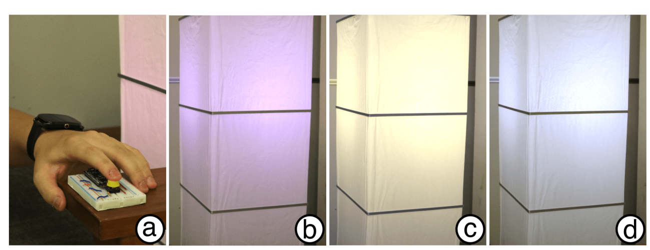

Figure 20. Using Dwell+ to augment a physical button to control an IoT lamp. (a) A user is pressing a physical button and the Dwell+ vibration is delivered via a smartwatch to indicate the colors of the lamp (b,c,d).

[3] Precision Microdrive 310-113 ERM motor

DISCUSSION

Adjustable Intervals and Expert Mode

Our studies find that participants can effectively manage nearly 10 levels of Dwell+ selection with less frequent 170ms intervals, but these effective levels degrade to 5 levels with more frequent 150ms intervals. This tradeoff between efficiency and accuracy could inspire several possible designs. Depending on the required efficiency in the applications, different lengths of interval can be applied. For instance, users can quickly invoke one of the four visible applications on the group button without unfolding the folder (Figure 15). When a greater number of levels are required such as picking a number for the setup of date, less frequent intervals may enable users to be less dependent on visual manipulation while on the move. This design then requires a system level agreement such that users would expect to perceive a faster Dwell+ operation before dwelling the finger on an interface containing a few levels of controls (*e.g.*, music interface in Figure 19).

It is also expected that users will be capable of faster Dwell+ once they receive enough practice over time. It will be therefore interesting to see the interaction between more effective levels and shorter intervals in providing an expert model of Dwell+ operations. Furthermore, the system could gradually increase or decrease the interval duration according to user performance in constant use of the Dwell+ interaction.

Working with Audio or Visual Feedback

While Dwell+ has been designed based on haptic feedback specifically using vibration in this paper, the same concept can be implemented using audio or visual feedback. In comparison, haptic feedback benefits interactions that prefer eyes-free or ears-free use. Audio- or visual-enabled Dwell+ can be implemented in lack-of-haptic-output devices and has benefits when the users skin is blocked from receiving sensations *e.g.*, by gloves. Moreover, Dwell+ can be enhanced by combining multiple feedback channels, but careful consideration should be given if the rich design may affect other concurrent applications or may annoy the users.

Working with other Input Methods

Dwell+ can potentially be integrated with any touch-based interaction to further expand its input space. For instance, these haptic ticks can be added on top of angular touch [17], posture touch [12], or on different fingers [33].

LIMITATIONS

Evaluation of Applications

To show the high applicability and feasibility of our work, a wide range of possible applications were proposed. However, these applications were not evaluated. Further research should investigate the performance and mental workload of the proposed applications.

Exploring Effective Dwell+ Design across Body

User Study 3 reveals that displacing the vibration from the users touch finger to the other hand will not affect the accuracy of selection, which sheds light on the possibilities of perceiving vibration across the body; such as a phone-holding hand, wrist-worn watch, smart ring, smart belt, and even from a phone in a pocket. However, this work did not exhaustively explore the optimal vibration design for other body regions, and the results of User Study 3 should not be applied to other areas of the skin directly. In particular, some body parts require stronger vibrotactile cues and longer intervals to clearly perceive those haptic beats. We encourage further studies to investigate how divergent haptic sources affect Dwell+ performance, and also how elongated breaks between ticks may help retain performance.

Real-world Scenario and Multi-tasking

Our studies were conducted in a well-controlled laboratory environment where users were presented with only the single task of selecting modes. In real world contexts, user performance to perceive haptic ticks from Dwell+ can be hindered when users are at the same time perceiving varied visual, audio, or haptic rhythms from the environment. This can happen frequently when *e.g.*, , using Dwell+ while walking or listening to music when haptic or audio rhythms are formed. This issue can be alleviated by extending the period of intervals and vibrations. We encourage future research to investigate the real-world performance and other improvements.

CONCLUSION

This paper presents Dwell+, a haptically-augmented solution for multi-level mode selection. Three user studies were conducted to explore for effective designs. The results show that a 94.5% rate of accuracy is achieved for 10-level selection using the 170ms interval with 3-beats-per-chunk vibration patterns, and a 93.82% rate of accuracy when using the 150ms interval with the same vibration design.

Since results were shown that the accuracies are almost the same while conducting touch and receiving vibrations from disparate body regions, Dwell+ can be extensively used for a wide range of interfaces by wearing or holding haptic actuators. Applications of Dwell+ include smartphones, smartwatches, stylus, HMDs, laptops and IoT devices.

For future work these researchers consider: a) exploring other effective Dwell+ designs for different usages, b) working with multiple output channels and other input methods, and c) investigating real-world and expert performance.

ACKNOWLEDGEMENTS

This research was supported in part by the Ministry of Science and Technology of Taiwan (MOST 106-2633-E-002-001), National Taiwan University (NTU-106R104045) and Intel Corporation.

REFERENCES

1. Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. 2012. Dwell-and-spring: Undo for Direct Manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1957–1966. DOI:http://dx.doi.org/10.1145/2207676.2208339
2. D. R. Beukelman and Mirenda P/. 2005. *Augmentative and Alternative Communication*. Brooks Publishing.

3. Andrea Bianchi, Ian Oakley, and Dong Soo Kwon. 2012. Counting Clicks and Beeps: Exploring Numerosity Based Haptic and Audio PIN Entry. *Interact. Comput.* 24, 5 (Sept. 2012), 409–422. DOI: **http://dx.doi.org/10.1016/j.intcom.2012.06.005**

4. David Bonnet, Caroline Appert, and Michel Beaudouin-Lafon. 2013. Extending the Vocabulary of Touch Events with ThumbRock. In *Proceedings of Graphics Interface 2013 (GI '13)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 221–228. **http://dl.acm.org/citation.cfm?id=2532129.2532166**

5. Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The Fat Thumb: Using the Thumb's Contact Size for Single-handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 39–48. DOI: **http://dx.doi.org/10.1145/2371574.2371582**

6. Stephen Brewster and Lorna M. Brown. 2004. Tactons: Structured Tactile Messages for Non-visual Information Display. In *Proceedings of the Fifth Conference on Australasian User Interface - Volume 28 (AUIC '04)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 15–23. **http://dl.acm.org/citation.cfm?id=976310.976313**

7. Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014a. Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 615–620. DOI: **http://dx.doi.org/10.1145/2642918.2647354**

8. Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. 2014b. Air+Touch: Interweaving Touch & In-air Gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 519–525. DOI: **http://dx.doi.org/10.1145/2642918.2647392**

9. Frederick W.J. Cody, Rebecca A.D. Garside, Donna Lloyd, and Ellen Poliakoff. 2008. Tactile spatial acuity varies with site and axis in the human upper limb. *Neuroscience Letters* 433, 2 (2008), 103 – 108. DOI: **http://dx.doi.org/10.1016/j.neulet.2007.12.054**

10. James C. Craig and Keith B. Lyle. 2001. A comparison of tactile spatial sensitivity on the palm and fingerpad. *Perception & Psychophysics* 63, 2 (01 Feb 2001), 337–347. DOI: **http://dx.doi.org/10.3758/BF03194474**

11. Chris Harrison and Scott Hudson. 2012. Using Shear As a Supplemental Two-dimensional Input Channel for Rich Touchscreen Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 3149–3152. DOI: **http://dx.doi.org/10.1145/2207676.2208730**

12. Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 627–636. DOI: **http://dx.doi.org/10.1145/2047196.2047279**

13. Seongkook Heo and Geehyuk Lee. 2011. Forcetap: Extending the Input Vocabulary of Mobile Touch Screens by Adding Tap Gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 113–122. DOI: **http://dx.doi.org/10.1145/2037373.2037393**

14. Ken Hinckley and Hyunyoung Song. 2011. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 801–810. DOI: **http://dx.doi.org/10.1145/1978942.1979059**

15. Christian Holz and Patrick Baudisch. 2013. Fiberio: A Touchscreen That Senses Fingerprints. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 41–50. DOI: **http://dx.doi.org/10.1145/2501988.2502021**

16. Minoru Hoshiyama, Ryusuke Kakigi, and Yohei Tamura. 2004. Temporal discrimination threshold on various parts of the body. *Muscle & nerve* 29, 2 (2004), 243–247.

17. Da-Yuan Huang, Ming-Chang Tsai, Ying-Chao Tung, Min-Lun Tsai, Yen-Ting Yeh, Liwei Chan, Yi-Ping Hung, and Mike Y. Chen. 2014. TouchSense: Expanding Touchscreen Input Vocabulary Using Different Areas of Users' Finger Pads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 189–192. DOI: **http://dx.doi.org/10.1145/2556288.2557258**

18. RJ Jacob and Keith S Karn. 2003. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind* 2, 3 (2003), 4.

19. Robert J. K. Jacob. 1990. What You Look at is What You Get: Eye Movement-based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. ACM, New York, NY, USA, 11–18. DOI: **http://dx.doi.org/10.1145/97243.97246**

20. Alexander Kulik, Jan Dittrich, and Bernd Froehlich. 2012. The Hold-and-move Gesture for Multi-touch Interfaces. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 49–58. DOI: **http://dx.doi.org/10.1145/2371574.2371583**

21. Takuro Kuribara, Buntarou Shizuki, and Jiro Tanaka. 2014. Vibrainput: Two-step PIN Entry System Based on Vibration and Visual Information. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 2473–2478. DOI: http://dx.doi.org/10.1145/2559206.2581187

22. Gordon Kurtenbach and William Buxton. 1994. User Learning and Performance with Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 258–264. DOI: http://dx.doi.org/10.1145/191666.191759

23. EUGENE C. Lechelt. 1975. TEMPORAL NUMEROSITY DISCRIMINATION: INTERMODAL COMPARISONS REVISITED. *British Journal of Psychology* 66, 1 (1975), 101–108. DOI: http://dx.doi.org/10.1111/j.2044-8295.1975.tb01444.x

24. Seungyon "Claire" Lee and Thad Starner. 2010. BuzzWear: Alert Perception in Wearable Tactile Displays on the Wrist. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 433–442. DOI: http://dx.doi.org/10.1145/1753326.1753392

25. Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 461–470. DOI: http://dx.doi.org/10.1145/1054972.1055036

26. Yi-Chi Liao, Yi-Ling Chen, Jo-Yu Lo, Rong-Hao Liang, Liwei Chan, and Bing-Yu Chen. 2016. EdgeVib: Effective Alphanumeric Character Output Using a Wrist-Worn Tactile Display. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 595–601. DOI: http://dx.doi.org/10.1145/2984511.2984522

27. G. A. Miller. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review* 63, 2 (1956).

28. Jerome Pasquero, Scott J. Stobbe, and Noel Stonehouse. 2011. A Haptic Wristwatch for Eyes-free Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 3257–3266. DOI: http://dx.doi.org/10.1145/1978942.1979425

29. Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. 2000. Control Menus: Excecution and Control in a Single Interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems (CHI EA '00)*. ACM, New York, NY, USA, 263–264. DOI: http://dx.doi.org/10.1145/633292.633446

30. Anne Roudaut, Eric Lecolinet, and Yves Guiard. 2009. MicroRolls: Expanding Touch-screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 927–936. DOI: http://dx.doi.org/10.1145/1518701.1518843

31. Eric Saund and Edward Lank. 2003. Stylus Input and Editing Without Prior Selection of Mode. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 213–216. DOI: http://dx.doi.org/10.1145/964696.964720

32. R. C. Simpson and H. H. Koester. 1999. Adaptive one-switch row-column scanning. *IEEE Transactions on Rehabilitation Engineering* 7, 4 (Dec 1999), 464–473. DOI: http://dx.doi.org/10.1109/86.808950

33. Atsushi Sugiura and Yoshiyuki Koseki. 1998. A User Interface Using Fingerprint Recognition: Holding Commands and Data Objects on Fingers. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST '98)*. ACM, New York, NY, USA, 71–79. DOI: http://dx.doi.org/10.1145/288392.288575

34. David Ternes and Karon E. MacLean. 2008. *Designing Large Sets of Haptic Icons with Rhythm*. Springer Berlin Heidelberg, Berlin, Heidelberg, 199–208. DOI: http://dx.doi.org/10.1007/978-3-540-69057-3_24

35. Colin Ware and Harutune H. Mikaelian. 1987. An Evaluation of an Eye Tracker As a Device for Computer Input2. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)*. ACM, New York, NY, USA, 183–188. DOI: http://dx.doi.org/10.1145/29933.275627

36. Graham Wilson, Craig Stewart, and Stephen A. Brewster. 2010. Pressure-based Menu Selection for Mobile Devices. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. ACM, New York, NY, USA, 181–190. DOI: http://dx.doi.org/10.1145/1851600.1851631