PinPad: Touchpad Interaction with Fast and High-Resolution Tactile Output

Jingun Jung
HCI Lab, KAIST
Daejeon, South Korea
jung.jingun@kaist.ac.kr

Eunhye Youn
HCI Lab, KAIST
Daejeon, South Korea
yeh1989@gmail.com

Geehyuk Lee
HCI Lab, KAIST
Daejeon, South Korea
geehyuk@gmail.com

ABSTRACT

We explored new interaction scenarios that can be realized when a touchpad outputs fast and high-resolution spatio-temporal tactile patterns to the touch-sensitive skin on the fingertips of a user. We first constructed a special tactile multi-touch touchpad called PinPad, which was capable of outputting fast and high-resolution tactile patterns using a 40 × 25 array of actuated pins. We then developed various interaction scenarios that could be realized using the prototype: 1) Tactile Target, 2) Guide and Constraint, 3) Multi-finger Output, and 4) Dynamic Partition. To evaluate the PinPad scenarios, we implemented demo applications, and conducted interviews with users to collect feedback about their experiences with PinPad and the PinPad scenarios. The participants confirmed the effectiveness of spatio-temporal outputs of PinPad in the scenarios. In particular, they provided diverse feedback regarding the unique tactile experiences of the fast and high-resolution outputs of PinPad.

ACM Classification Keywords

H.5.2. User Interfaces: Input devices and strategies; H.5.2. User Interfaces: Haptic I/O

Author Keywords

PinPad; tactile touchpad interaction; fast and high-resolution tactile output

INTRODUCTION

A touchpad is now the de facto standard pointing device for laptop computers. Recent touchpads are, in fact, more than just pointing devices because they now have large multi-touch sensing surfaces that enable users to employ various multi-touch gestures, such as two-finger scrolling and pinching. Further, the application of touchpads is expanding to other domains beyond computers, such as car infotainment environments [20] and smart TV environments [3].

Because of their important role, continued research efforts have been made to improve their capabilities, for instance, in enriching input vocabulary using force sensing [26], enabling more direct manipulation using hand-silhouette feedback [19], and enabling a large palm-rejecting touchpad using an optical technique [8]. However, the studies are mostly focused on the input aspect of a touchpad. A touchpad can deliver tactile information to the fingers, as well as sense their movements, but research efforts on the output aspect of a touchpad have been relatively less focused on. Many studies on providing tactile click feedback and controlling surface friction exist in touch-interface literature, but, to the best of our knowledge, no study has explored the output channel through the high-resolution touch-sensitive skin of the fingertips.

Therefore, in this study, we performed a horizontal, breadth-first investigation to enumerate new interaction scenarios that can be realized when a touchpad outputs fast and high-resolution spatio-temporal tactile patterns. For this purpose, we constructed a special tactile multi-touch touchpad, called PinPad that was capable of outputting fast and high-resolution tactile patterns using a 40 × 25 array of actuated pins. We then developed various interaction scenarios that could be realized using the prototype. We classified them into the four categories: 1) Tactile Target, 2) Guide and Constraint, 3) Multi-finger Output, and 4) Dynamic Partition. We then implemented demo applications to experience some of the key PinPad scenarios, and conducted interviews with users to collect feedback on their first experiences with the PinPad prototype and PinPad scenarios.

In the remainder of the paper, we review related work, describe the PinPad prototype, present the PinPad interaction scenarios, describe the demo applications, and summarize user feedback from the interviews. Finally, we discuss limitations and future work in the conclusion.

RELATED WORK

Various types of tactile output technologies, which may be used to tactually augment touchpads, have been researched. They can be characterized by their spatial and temporal resolutions and may be positioned in the design space of tactile display surfaces, as shown in Figure 1. For example, tactile click feedback methods are in the (low, high) corner, texture modulation methods, such as TeslaTouch [1], belong to the (low, medium) region, and some of the shape-changing displays belong to the (medium, medium) region. Pin-array displays, such as HyperBraille [16], which were developed for blind people, belong to the (medium, high) region because they are mostly used for displaying high-resolution but slowly

CHI 2017, May 6-11, 2017, Denver, CO, USA.

http://dx.doi.org/10.1145/3025453.3025971

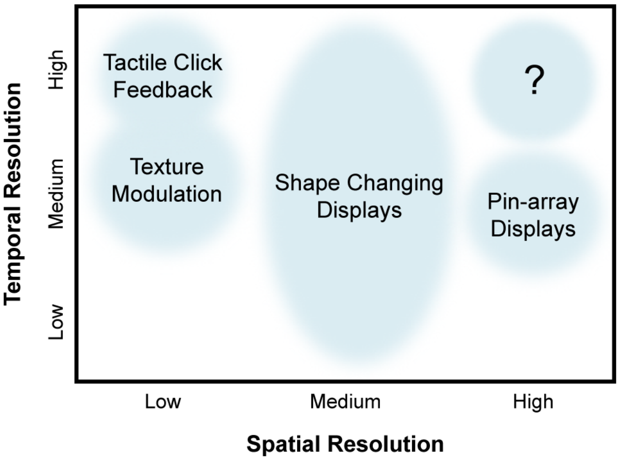

Figure 1. The design space of tactile display surfaces.

varying patterns for blind users to perceive. As the picture shows, the (high, high) region is yet to be explored. We are interested in new interaction techniques that can be realized in the (high, high) region, and our goal in the present study, in particular, is to explore new interaction possibilities when a touchpad is augmented with the (high, high) tactile output. To the best of our knowledge, this is an unexplored territory. In this section, we review research examples in each of the major regions in Figure 1 focusing on the possible interaction techniques.

Tactile Click Feedback

Tactile click feedback is a common type of tactile feedback for touch devices. For example, the Force Touch trackpad from Apple provides tactile click feedback using an electromagnetic actuator when users press the surface. In fact, tactile click feedback has a long history. Fukumoto and Sugimura [6] attached an actuator to a PDA and drove it with a short pulse when a user tapped on the screen. Poupyrev et al. [23] developed their own actuator, which was adequate for simulating the click feeling of a physical button for a mobile device. Rekimoto and Schwesig [26] utilized a tactile 'click' to distinguish the touch and the press states for their pressure-sensing device. Kim and Lee [14] presented a haptic feedback method not only for the 'click' sensation but also for press and release sensations similar to the experience of using a physical button. The main purpose of tactile click feedback is to provide users sureness in carrying out selections through touch operations.

Texture Modulation

Several methods have been proposed to control the texture of a touch surface. They can be used to augment GUI objects and enhance the performance of GUI manipulations. Bau et al. [1] proposed a texture modulation method using a phenomenon called 'electrovibration', which is created by the electrostatic force between a finger and a touch surface. They also presented application scenarios including simulation of a surface with variable friction thereby tactually augmenting the touch interface. Kim et al. [15] used the same method to render tactile 3D features on a touchscreen. Another texture modulation method used 'squeeze film effect' [30]. Levesque et al. [18] used the method to increase the friction on the area of a target and showed that the pointing performance was enhanced. Jansen et al. [12] proposed a method to change the texture of a surface using a magnetorheological fluid. It could also simulate haptic impressions like pressing a mechanical button and inducing a vibration to represent the active area. However, these methods provided texture modulation that was spatially global and could not produce high-resolution spatial patterns.

Shape Changing Displays

As the name suggests, the goal of shape-changing displays is to output a static or dynamic shape that users can trace and feel. Pioneering examples of this are Feelex [11] and Lumen [22]. Examples that are more recent include TRANSFORM [10] and InForm [5]. They consist of an array of actuators and focus on representing and manipulating large-scale 3D shapes. They demonstrated new possibilities using dynamic 3D shapes to provide affordance and constraints for interaction, but were distantly related to tactile feedback for touchpads.

The studies that actively change the shape of a touch surface in order to provide tactile cues for target objects are more closely related. Harrison and Hudson [9] presented a changeable physical button using pneumatic actuation. Tactus technologies [4] created a system that inflates predefined button areas on demand using microfluidics. Miruchna et al. [21] used a thermo-responsive hydrogel layer to provide protruding cues on a touchscreen and augmented GUI elements such as buttons and sliders. Tsimeris et al. [29] realized localized haptic feedback for an on-screen keyboard using an array of electromagnets and permanent magnets. Sahoo et al. [27] presented a deformable screen made of elastic fabric and indium-tin-oxide electrodes. It could provide visual and tactile feedback for touch inputs using the local movement of a surface. Shape-changing displays have not usually employed high-resolution tactile output, but their temporal spectrum is wide, ranging from slow pneumatic actuation [9] to fast electromagnetic actuation [27], represented by a vertical long region in Figure 1.

Pin-array Display

Pin-array displays have long been used to enable blind people to perceive letters and graphical shapes. Early pioneering examples include Optacon [7] and Pantobraille [25]. Optacon used an optical scanning module and a pin-array module to convert an optical pattern into a tactile pattern. Pantobraille combined a force feedback device and a Braille display to enable users to perceive large textures and forms. The most recent examples of advanced pin-array displays are from the HyperBraille project—two-dimensional arrays of Braille modules combined with embedded touch sensors. Many studies utilizing HyperBraille have been conducted. For example, Spindeler et al. [28] enabled blind people to perceive and manipulate GUI widgets. Prescher et al. [24] presented a planar Braille Window System supporting various views of a graphical window system and interactions with multi-touch gestures.

In terms of the spatial resolution of the tactile output, these studies are similar to our current study. However, in terms of the temporal resolution of the tactile output, these studies focused on the medium region, as shown in Figure 1, because tactile output in these studies was mainly used for delivering shape information and was required to be static most of the time for blind people to perceive. In addition, their research problems were different from ours because tactile output in their cases was the main interaction modality; our research question is—how the tactile output of a touchpad can supplement GUI interaction where vision is the main interaction modality.

PINPAD PROTOTYPE

In order to build a touchpad that can output fast and high-resolution tactile patterns, we surveyed commercially available Braille display modules and found a vertical-type Braille display module from metec AG to be a promising option. We examined it to see if it satisfied our requirements. The first was spatial resolution. It was designed for Braille displays, and therefore, the pin spacing was 2.5 mm. As it approaches the tactile spatial threshold of the fingertip, we thought that it is good enough for our purpose.

The second question that we examined was whether it would be fast enough for building a tactile display with a high refresh-rate. Data transmission through an array of modules was fast enough, taking less than 10 ms for an array of 100 modules. A possible bottleneck was the mechanical speed of the pins. The rising time of the pin was ~100 ms as per the datasheet. We were concerned about this duration because it would mean that the maximum refresh rate of the tactile display was 10 Hz. As it turned out later, this was not a critical problem because the data transmission rate determined the smoothness of the pin pattern translations, which was most important in the PinPad interaction scenarios that we explored in this study.

The third question was whether it would be possible to add a multi-touch sensor layer on the modules. In fact, the Braille module had built-in touch sensors, but the sensors were not suitable for our purpose because they worked only when the Braille pins were down. In addition, their outputs were binary, meaning that we could not use interpolation to implement a high-resolution touchpad sensor. Therefore, we had to consider adding a custom-made touchpad sensor layer on the Braille modules. We were not sure whether a touchpad sensor would work when it had many holes for the Braille pins to pass through. In addition, we were not sure whether a touchpad sensor would work with an air gap created due to the raised pins. Fortunately, the touchpad sensor circuit that we chose was good enough to overcome these issues.

Finally, we needed to determine the size of the touchpad. It had to be large enough to allow multi-touch operations, and therefore the lower bound that we intended was 60 mm × 100 mm, about the size of a touchpad in modern laptops. The upper bound was determined by the budget available for the prototype. With each Braille module costing about €55, we decided to use 20 × 5 array modules, which enabled us to make a 62.5 mm × 100 mm touchpad.

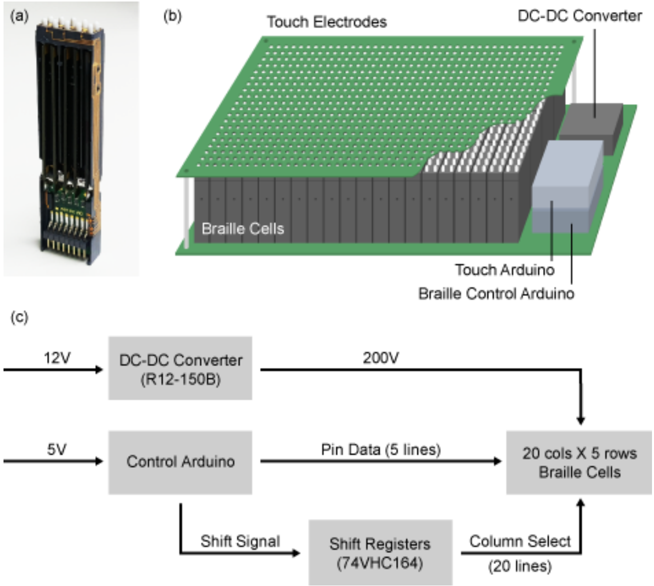

Figure 2. The pin-array display: (a) the Braille module, (b) the overall structure, and (c) the system block diagram.

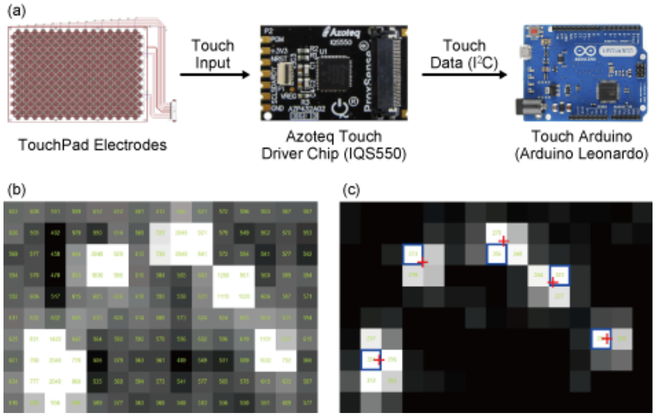

Figure 3. The touchpad sensor: (a) the system block diagram, (b) raw sensor data, and (c) sensor data after thresholding and estimated touch positions.

Implementation

Figure 2a shows the Braille module (Modul D2, metec). It has 2 (x) × 5 (y) pins with a pin spacing of 2.5 mm and a face area of 5 mm (x) × 12.5 mm (y). The pin stroke is approximately 0.7 mm, the minimum pin force is 0.3 N, and the pin rise-time is 100 ms.

Figure 2b shows the overall structure of the prototype. A 20 (x) × 5 (y) array of the Braille modules is on the backplane board, thereby a 40 (x) × 25 (y) array of pins on the prototype. The backplane board provides two power supplies to the modules—5 V for the logic circuit and 200 V for the piezoelectric driver. As shown in Figure 2c, the backplane has a data bus circuit from the Arduino board to the modules. The bus clock is 1 MHz, the data size for each module is 16 bits, and therefore, data transfer to all modules can be carried out within 10 ms.

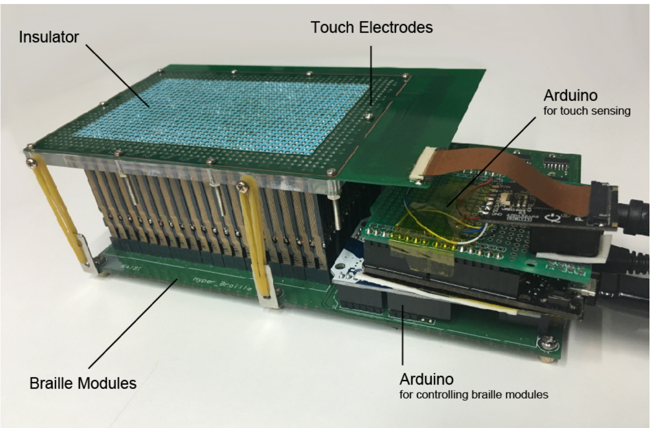

Figure 4. The PinPad prototype.

Figure 3a shows the block diagram of the touch sensor including the sensor electrode pattern with 15 (x) × 10 (y) electrodes. The electrode position and spacing were appropriately selected to minimize the effect of the pinholes. The electrodes were driven and sensed by a touchscreen driver (ProxSense IQS550-TS43, Azoteq). We chose this driver because all the information to customize the driver for our special purposes, such as touch detection over the raised pins, is publicly available[1]. Figure 3b shows raw sensor data from the touchpad driver when four fingers are on the prototype. As the sensitivities of the electrodes were not uniform, a non-uniform thresholding was needed to detect touches. Figure 3c shows sensor data after thresholding. Red crosses in the picture represent estimated touch positions, and they are determined based on the peak cell in a blob and its four neighbors. For precision, the x and y coordinates of a touch position are determined by the center of mass of the three horizontal cells and three vertical cells including the peak cell, respectively. The time taken by the touchpad driver to scan all electrodes mainly depends on an integration time parameter. Large integration time increases the sensor's sensitivity but also increases scan time. We opted for sensitivity to cope with the air gap caused by the raised pins, and the resulting scan rate was about 11 Hz.

Figure 4 shows a picture of the final PinPad prototype[2]. A laser-cut paper layer placed on the touchpad acts as an insulator between the electrodes and the fingers. A PC application, controlling the two Arduino boards, paced the cycle of reading the touchpad data and controlling the Braille pins. The speed of the cycle was mainly limited by the scan time of the touchpad driver.

PINPAD INTERACTION

We had a series of brainstorming sessions to develop new interaction scenarios that we would be able to demonstrate using the PinPad prototype. We present them in this section. They can be categorized as: 1) Tactile Target, 2) Guide and Constraint, 3) Multi-finger Output, and 4) Dynamic Partition.

Tactile Target

Physical controls provide tactile cues that enable users to locate and manipulate them without visual feedback. For instance, one can locate the power button on the side of a smartphone without looking once it is in the hand. In contrast, GUI targets do not have tactile cues and cannot be located by feeling. It may be possible to turn silent targets into tactile targets by providing a tactile representation on the touchpad. Figure 5a shows an example. When a finger is moving toward a checkbox, a tactile pattern may be given in the predicted position on the touchpad so that the finger can perceive the checkbox by tactile sensation.

Using tactile representations to create tactile targets may be more useful in cases where visual feedback may not be effective. Figure 5b shows such an example. After a search command, a word processor highlights all of the targets found in the text area, and shows the positions of the targets on the scrollbar (red lines). In this case, users often split their visual attention between the text area and the scrollbar. After users perceive an overview of the targets visualized in the scrollbar, they may be able to scroll the document, focusing on the text area, if the touchpad provides the tactile representations of the targets.

Guide and Constraint

Edge scrolling is a common feature in modern touchpads. It is an effective technique because a touchpad edge provides tactile cues to guide and constrain finger movements. PinPad can provide such tactile cues dynamically, following fingers, to guide and constrain their movements. At the same time, the tactile cues may play the role of a 'feedforward' provider of possible actions.

1D Control Guide

Many GUI controls, such as sliders and scrollbars, have one degree of freedom. When a finger is on a slider, for example, the touchpad may provide a linear tactile cue to guide the finger's movement as shown in Figure 5c. In addition, the touchpad may provide a 'bumpy' tactile cue to constrain the finger's movement at the boundaries of the slider. The linear tactile cue in this example also provides feedforward about a possible action on the slider (horizontal movement). 2D object handles in a CAD program are another example of 1D control when their movements are constrained, for instance, by a fixed aspect ratio. In this example, the role of tactile cues as a feedforward provider may be more important because object handles can provide limited visual cues compared with sliders and scrollbars. A multi-level menu is an example of a dynamic 1D control depending on its context. When it is open, the touchpad may provide a vertical tactile cue to help the finger move within the menu. When the finger reaches an item with a sub-menu, the touchpad may provide a horizontal tactile cue to help the finger open the sub-menu.

Content Guide

The content in a window may have a preferred cursor movement direction. For instance, the text area in a window may prefer horizontal cursor movements when it is in text selection mode. As shown in Figure 5d, tactile lines on the touchpad may encourage the finger movement in the direction of the text

---

[1] http://www.azoteq.com/

[2] Detail information to reproduce the prototype, including the circuit schematics, PCB designs, Arduino codes, and Windows interface program, is available from http://hcil.kaist.ac.kr.

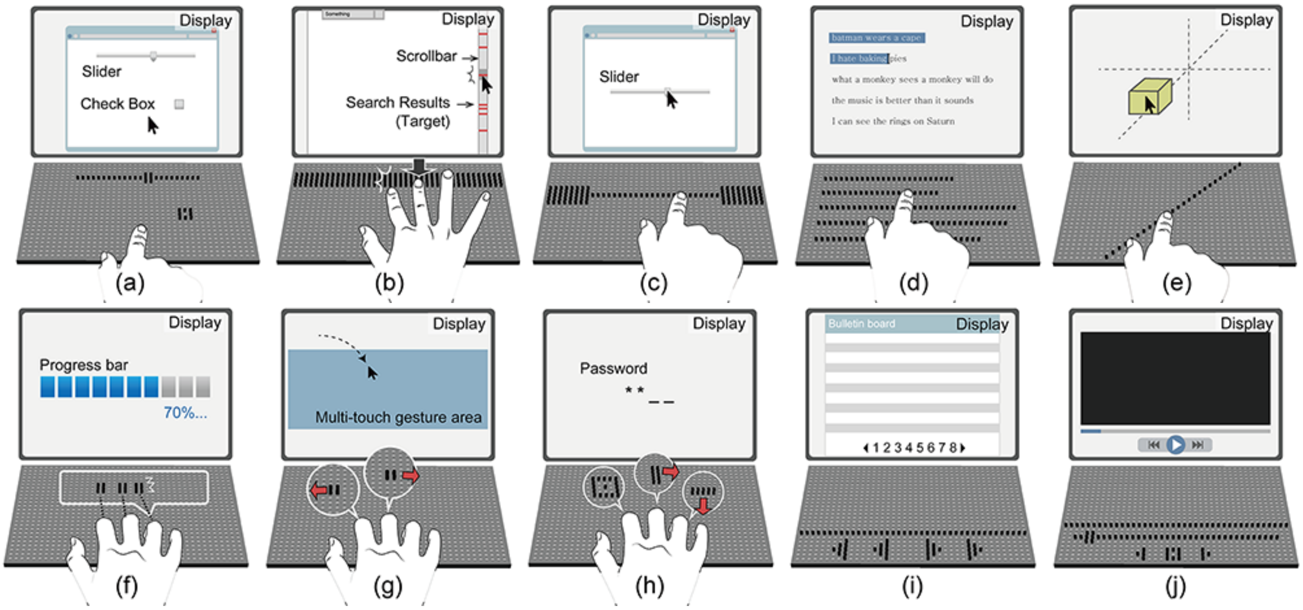

Figure 5. Tactile Target scenarios: (a) Tactile Controls and (b) Tactile Scrollbar. Guide and Constraint scenarios: (c) 1D Control Guide, (d) Content Guide, and (e) 3D Drawing Guide. Multi-finger Output scenarios: (f) Tactile Display, (g) Multi-finger Feedforward, and (h) Secure Message. Dynamic Partition scenarios: (i) Dynamic Page Control and (j) Dynamic Video Control.

to facilitate safe text selection. Selecting text encompassing multiple lines will still be possible, but will require an effort to move against the texture of the tactile lines. Another example of content with a preferred direction may be a long document in scrolling mode. When two fingers are on the touchpad, it may provide vertical tactile lines to indicate and encourage vertical movements. Yet another example is that of 3D drawings as shown in Figure 5e. An object in a 3D drawing may need to be moved along the coordinate axes or along another object. Depending on the current move mode, the touchpad may provide a suitable tactile cue in the preferred directions.

Multi-finger Output

In the previous scenarios, the touchpad provides a tactile cue to aid an input operation. The tactile channel of the touchpad may itself be used as an active output channel. The original use of the Braille cells, which PinPad is made of, is for outputting symbolic messages. Combined with a multi-touch sensing capability, PinPad can deliver distinct spatio-temporal patterns to multiple fingers on it simultaneously. Scenarios in this category require the fingers to rest on the touchpad. This requirement is not in conflict with standard touchpad operations because resting all fingers on the touchpad does not usually trigger any function. In order to use the multi-finger output scenarios more actively, we may need to disable the functions assigned to four-finger gestures.

Tactile Display

A tactile display transfers information using multiple tactors on the skin, such as on the back [13] and on the wrist [17]. Similarly, a touchpad may transfer information by providing distinct tactile patterns on different fingertips. As illustrated in Figure 5f, the touchpad may indicate the current progress of a job, such as copying a file and compiling a program, on the four fingertips on it. The tactile display function may also be used for notifying the user of various system and application events, such as mail arrivals and calendar events.

Multi-finger Feedforward

In the Guide and Constraint subsection, we already presented examples where the touchpad may use a dynamic tactile cue to provide feedforward about possible actions. The same concept may be applied to multi-finger cases. When the cursor enters a region that supports pinch gestures, the touchpad may provide wave patterns moving in opposite directions to the index and middle fingers as illustrated in Figure 5g. In other words, the touchpad may provide feedforward about pinch gestures using the tactile display function.

Secure Message

The tactile display transfers information using tactile patterns (tactons), and therefore is intrinsically secure. The secure nature of the tactile display may be used to implement a secure password interface. An example is given in Bianchi et al. [2]. They proposed a tactile password interface using three tactile buttons. The system could output individual tactons on the buttons simultaneously and the user was able to select the correct tacton. By repeating this process, the user could select a series of tactons, which formed a password. The exact scenario can be realized using PinPad and its tactile display function (Figure 5h). The tactile patterns on PinPad may be visible if they are closely observed. In order to avoid this problem, we may need to use small spatio-temporal patterns that will be completely covered by the fingers.

Dynamic Partition

A touchpad may be partitioned into multiple regions supporting different functions. For instance, a touchpad may use the right edge area for vertical scrolling and the remaining area for pointing. However, touchpad partitioning may have visibility problems; the user may not be aware of its existence. In addition, using partitioned sections may be difficult because their boundaries are not visible. Touchpad partitioning may become more useful with PinPad because it can provide tactile boundaries and tactile patterns in the partitioned sections. Figure 5i shows an example of this—Dynamic Page Control. When a Web page with page navigation control is active, PinPad may create a section on the touchpad corresponding to the navigation control. The user may use the section to access the navigation controls directly. Figure 5j shows another example—Dynamic Video Control. When a movie player is active, PinPad may create a partitioned section corresponding to the video play control. The section may stay active even after the play control disappears from the screen, enabling the user to control the video with tactile feedback only. In these scenarios, visual feedback by the tactile patterns on PinPad, though limited, plays an important role—enabling the user to perceive the existence and function of a partitioned section before touching the surface. Now that the partitioned section is visible, it may also be made dynamic; it may only be active in a relevant context and does not claim a static region.

PINPAD APPLICATIONS

In order to experience and evaluate the PinPad scenarios, we implemented demo applications for seven of them (Figure 6). We describe each of them below.

Tactile Controls (Tactile Target): As shown in Figure 6a, the screen shows a few controls, such as push buttons and textboxes. The cursor is initially away from the controls. When the user puts a finger on the touchpad, the tactile representations of the controls appear on the touchpad. As the user slides the finger on the touchpad, they can feel the controls. When a control is selected, its tactile representation changes, and the user may tap to activate it.

Tactile Scrollbar (Tactile Target): As shown in Figure 6b, the screen initially shows a window with a scrollbar. It is assumed that a text search was done, and therefore, the text area shows the found targets highlighted in yellow, and the scrollbar shows the corresponding positions in red. In this application, all dragging operations are mapped to the movements of the scrollbar handle. Dragging on the touchpad, the user can move the scrollbar handle and feel the targets on the scrollbar. There is a bump feedback when the user tries to scroll up toward the top of the document or down toward the bottom of the document. The user then perceives that more scrolling is impossible. The virtual document height is about 4 times that of the window, and the user can traverse the whole scrollbar in about six full strokes on the touchpad.

Menu Guide (1D Control Guide): As shown in Figure 6c, the screen initially shows the top-level menu of a multi-level menu. The user intends to select a command in a sub-menu. When the user arrives at a menu item with a sub-menu, the application presents a horizontal tactile guide on the touchpad. The user opens the sub-menu by following the tactile guide. The tactile guide disappears when the cursor enters the sub-menu, and the user continues to move to select a target item.

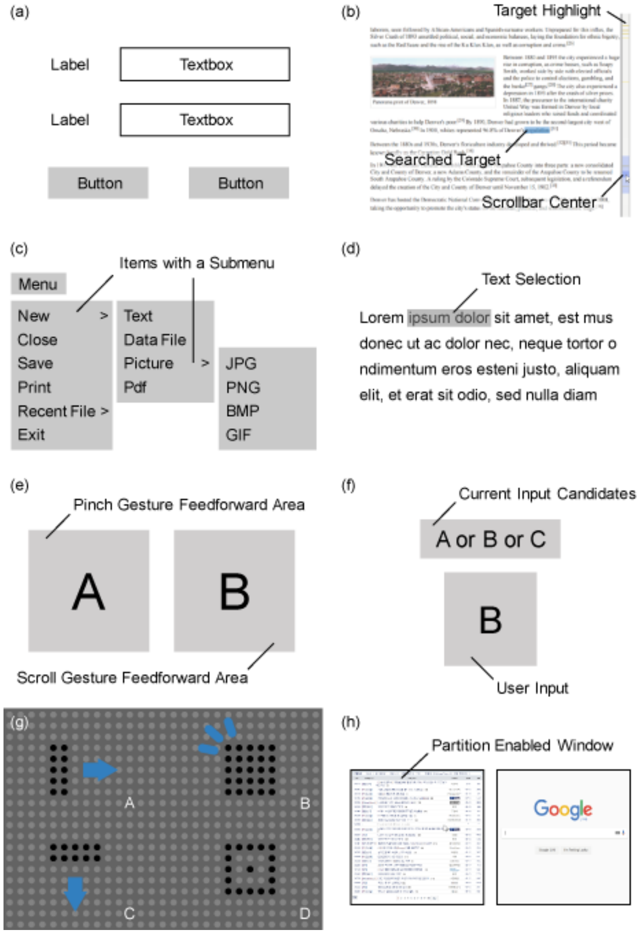

Figure 6. Demo applications implementing some of the PinPad scenarios: (a) Tactile Controls, (b) Tactile Scrollbar, (c) Menu Guide, (d) Text Guide, (e) Two-finger Feedforward, (f) Tactile Password, (g) Tactons used in a tactile password, and (h) Dynamic Page Control.

Text Guide (Content Guide): As shown in Figure 6d, the screen initially shows a window with text. The user can enter the text selection mode by double-tapping in the window. During the selection mode, tactile lines appear on the touchpad. The user can expand the text selection by dragging the finger aided by the tactile lines. The tactile line-spacing corresponds to the text line-spacing and therefore, it enables the user to count lines in the case of selection of multiple lines.

Two-finger Feedforward (Multi-finger Feedforward): As shown in Figure 6e, the screen initially shows two windows—A and B. The user rests their fingers on the touchpad. When the cursor is outside these windows, no stimuli are presented to the fingers. When the cursor is inside window A, vertical wavy tactons moving in opposite directions are presented to the first two fingers. The user then perceives that pinch gestures can be used, and uses a pinch gesture to enlarge the content of window A. If the cursor is in window B, horizontal lines occur as wavy tactons in the same direction as the first two fingers. The user perceives that scroll gestures can be used in window B.

Tactile Password (Secure Message): As shown in Figure 6f, the screen initially shows a password dialog box. In response to an instruction, the user puts fingers on the touchpad. The application then presents three distinct tactons on the first three fingers on the touchpad. The user then responds by tapping on the proper tacton. After this interaction, the application indicates which tacton is tapped to check whether it is right or not. The application uses four distinct tactons, and therefore the length of the password with n combinations is $2n$ bits ($n \times \log_2 4$). Figure 6g shows the four tactons used. The first tacton is a vertical line moving to the right. The second tacton is a blinking solid square. The third tacton is a horizontal line moving downward. The last tacton is a static empty square with a single dot.

Dynamic Page Control (Dynamic Partition): As shown in Figure 6h, the screen initially shows two windows, with one of them showing a Web page with a page-navigation control consisting of next, previous, jump-forward, and jump-backward buttons. When the window with the navigation controls becomes active, a partitioned section corresponding to the page-navigation control appears on the touchpad. The user can feel buttons in the section and tap on them to navigate between pages.

USER INTERVIEWS

We conducted interviews with users to collect their experiences with PinPad and PinPad scenarios. Our focus was on whether they could perceive various tactile expressions of PinPad, and whether they felt that the PinPad scenarios were useful and helpful. We also hoped to find unexpected problems in the hardware and the scenarios that would guide our future prototype and interaction design.

We recruited eight participants (3 females) from our university. Their ages ranged from 18 to 30 (average age = 22.0). They had varying prior experiences with haptic touchscreens and touchpads, ranging from none to routine use. They were all right-handed.

We interviewed two participants at a time. We explained the goal of the interview and the functions and limitations of our prototype. We asked them to focus more on the interaction concept than the quality of the prototype. They were allowed to use PinPad for about 3 minutes before the experiment.

We explained an application with its demo. Two participants then took turns to use the application and were allowed to discuss their experiences freely. We asked them to 'think-aloud' and recorded what they said. After repeating this for the seven demo applications, we asked them to answer a questionnaire, which consisted of 7-point Likert scale questions and additional open-ended questions. The interview took about an hour for each pair of participants. The whole interview was video-recorded in each case.

Results

Table 1 summarizes the answers to the 7-point Likert scale questions. Most of the average scores are above 5. We first summarize major participant comments for each application, and then report common feedback and observations at the end of the section.

Tactile Controls: All participants except one could perceive the tactile textboxes and buttons. All of them agreed that the scenario would be useful in real-world applications. One of the participants commented: "I felt it was rough and sandy because of the pins."

Tactile Scrollbar: All participants agreed they could perceive both targets and the end of a document. The answers about usefulness were divided. Two of them commented that they usually used a keyboard to jump to the next target, so they did not think the scenario was useful. Others liked it because they could quickly find targets. Regarding the tactile feedback of targets, they commented that they felt "lines passing by" and "convex shapes." In relation to the tactile feedback for the end of a document, five commented: "I felt like I was being blocked." One mentioned, "I can more quickly sense the end of the document with the tactile feedback than with a visual feedback."

Menu Guide: Half of the participants answered that they experienced an overshoot when they tried to open a submenu in a cascaded menu. In this scenario, they agreed that the tactile feedback was helpful in recognizing an item within a submenu and in opening a submenu. One of the participants commented: "It would be better if there is a tactile stimulus when a cursor leaves a menu."

Text Guide: All participants, except one, could perceive the tactile line representing text lines and spaces between them. They mentioned it was easier to select a text block with the help of tactile lines. Some mentioned, "To represent words, sentences, and paragraphs with tactile feedback will be more helpful because they are the units of text selection." In addition, most participants liked the tactile feedback because it enabled them to distinguish between a tracking mode and a selection mode clearly.

Two-finger Feedforward: All participants could perceive the two feedforward patterns for the pinch and scrolling gestures. They agreed that such feedforward would be effective in real applications. However, one of them questioned whether it would be possible to design an intuitive feedforward pattern for each gesture when there are many gestures to use.

Tactile Password: All participants agreed that the tactile password has an advantage in terms of security. While most of them could finally distinguish the four tactons, all participants said that distinguishing patterns under the three fingers at the same time was difficult. In particular, they tended to be confused between two tactons of opposite movement directions. One of them mentioned, "I have to concentrate on the sensation of one finger after another in order to distinguish tactons."

Dynamic Page Control: All participants agreed that the tactile partition line on the touchpad was helpful in distinguishing between two areas and in stopping the finger from moving into the page control area while moving the cursor. On the other hand, some of them had difficulty in distinguishing between

Table 1. Answers (mean scores) to the 7-point Likert scale questions in the interviews. The standard deviations (SD) are given in parentheses.

Applications	Questions	Mean scores (SD)
Tactile Controls	I can perceive the textbox tactually.	5.86 (0.90)
	I can perceive the button tactually.	5.86 (0.90)
	The tactile representations of controls will be useful in the future.	5.71 (0.95)
Tactile Scrollbar	I can perceive search results tactually while scrolling.	6.29 (0.76)
	The tactile stimulus at the end of a document is helpful.	6.57 (0.53)
	Search results expressed tactually on the touchpad will be useful.	6.14 (0.90)
	The end of scrolling expressed tactually on the touchpad will be useful in the future.	6.17 (0.41)
Menu Guide	I have a prior experience of losing control while opening a sub-menu in a cascaded menu.	5.00 (1.41)
	The tactile feedback is useful for navigating a cascaded menu.	5.29 (1.11)
Text Guide	I have a prior experience of having difficulty while selecting text using a touchpad.	5.86 (0.69)
	I can easily perceive the text lines tactually.	5.86 (0.69)
	I can distinguish between different text lines tactually.	5.71 (1.70)
	The tactile guide is helpful in drag-selecting text in a line.	5.86 (1.75)
	The tactile guide is helpful in drag-selecting text spanning multiple lines.	5.29 (1.11)
Two-finger Feedforward	I can easily perceive the pinch-gesture feedforward.	5.86 (0.69)
	I can easily perceive the scroll-gesture feedforward.	6.00 (0.82)
	Tactile gesture feedforward will be useful in the future.	5.86 (0.69)
Tactile Password	I can distinguish between the four tactons.	5.00 (1.63)
	I can perceive the stimuli given to fingers simultaneously.	4.71 (1.80)
	Tactile representation of a password will improve the security level in the future.	6.00 (0.82)
Dynamic Page Control	The tactile borderline prevents me from touching the tactile widget by mistake.	5.71 (0.49)
	I can turn pages using the tactile widget without looking at it.	5.00 (1.15)
	I can stop a touchpad gesture immediately on perceiving the borderline.	5.43 (0.98)
	The tactile widget buttons are distinguishable.	4.71 (1.25)

the shapes of tactile buttons in the page control area. One of them commented that the locations of the tactile buttons were more helpful in distinguishing the buttons than their shapes.

All participants answered that they had never experienced such high-resolution stimuli on the fingers. To the question "What do you think about the combination of touch inputs and the fast and high-resolution spatio-temporal tactile outputs?" everyone answered positively. One of them, however, complained about the mismatch between the tactile feedback and the visual information. Some participants commented: "The incessant tactile stimuli might be disturbing, so they should appear only when they are necessary." When we asked them to point out the differences between the PinPad outputs and the vibrotactile outputs, most of them mentioned the diversity of the tactile feedback. Three participants said that a more delicate control of the touch input was possible with the tactile feedback. Other comments were 'funny', 'intuitive', 'quieter than vibration,' and 'per-finger stimuli'.

Participants' preferences regarding the scenarios were diverse. One participant preferred the Dynamic Page Control scenario, as the most useful one. Other participant said that the Tactile Scrollbar and Two-finger Feedforward would be most useful. Some participants said that Tactile Password was unique and fancy. As the interview was short, we were concerned that the participants might not be able to adapt to the new interface and stimuli, but almost all participants seemed to adapt well to most of the scenarios.

DISCUSSION

Limitations of the Prototype

The PinPad prototype worked well enough to be used in the interviews. However, it certainly has a few glitches that need to be amended in the next stage of this research. First, the tactile pins would be stuck due to the friction with the holes. The pins were pushed up by piezo-benders but were restored back by their own weight. Therefore, a slight mismatch between the pin (or module) spacings and the hole spacings in the touchpad layer would cause the problem. This problem did not affect the tactile sensation much because the stuck pins did not resist the fingers, i.e., they returned to the down position as soon as they were touched. Nevertheless, the stuck pins were sensed by the fingers, especially when the fingers moved fast. This problem has to be fixed.

Second, the touch position data from the touchpad sensor contained noticeable noise. In order to overcome the air gaps due to the raised pins, we had to set the touch threshold as low as possible in addition to setting the integration time of the touchpad driver high, as mentioned in Implementation section. This noise problem was not a serious limitation in the current study, but should be marked for future study, where we will need to measure and compare performance metrics quantitatively in order to show the benefits of PinPad in terms of usability.

Future Work

We used diverse tactile patterns in the scenarios. Participants could distinguish between a horizontal line and a vertical line easily, but they had difficulty in perceiving shapes, such as a triangular button. They could distinguish temporally different patterns even when their shapes were similar. A research question here is—what would be the best tactile representations of GUI objects and operations for PinPad. A related question may be the design of diverse tactons that may be easily distinguished for the PinPad applications.

Another idea put forward during the study was inverting the pin patterns, i.e., using negative (recessed) patterns instead of positive (raised) patterns. In some scenarios, the user moved along the pins that were raised in a row. In these scenarios, there were opinions that it would be more comfortable if the finger could move along a groove (where the pins were not raised) surrounded by raised pins. Using negative patterns may be a better option for some applications.

Tactile patterns, raised patterns in particular, invite pressing actions. In addition, rising pins may invite a reaction. If PinPad can sense physical properties such as localized pressing force, it will be able to react with tactile outputs in more interesting ways. Augmenting PinPad with better finger sensing capability, such as force sensing and hover sensing, may enable even richer interaction scenarios.

A larger touchpad may invite diverse interactions including bi-manual touchpad interaction [8]. We may enlarge PinPad to cover the entire keyboard and touchpad area of a laptop. PinPad then may enable a programmable physical keyboard where, for example, keys have individually unique spatio-temporal tactile labels and programmable compliance behaviors. PinPad may provide various tactile widgets depending on the application context. It is also capable of delivering tactile signals to the palms, which may have a larger bandwidth than the fingertips.

CONCLUSION

We constructed a PinPad prototype, developed scenarios for PinPad, implemented demo applications, and collected user feedback about the PinPad and its applications. We believe that this is the first report on the new possibilities of a touchpad with fast and high-resolution tactile feedback. We hope that our experiences shared in this paper will stimulate research on technologies that enable fast and high-resolution tactile feedback in touch-based interfaces. Immediate future work will include improving the current prototype by fixing problems found in this iteration, and demonstrating the benefits of PinPad in terms of objective usability metrics.

ACKNOWLEDGMENTS

This research was supported by the KAIST High Risk High Return Project (HRHRP).

REFERENCES

1. Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. TeslaTouch: electrovibration for touch surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, 283–292.
2. Andrea Bianchi, Ian Oakley, and Dong Soo Kwon. 2010. The secure haptic keypad: a tactile password system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1089–1092.
3. Sangwon Choi, Jaehyun Han, Geehyuk Lee, Narae Lee, and Woohun Lee. 2011. RemoteTouch: touch-screen-like interaction in the tv viewing environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 393–402.
4. Craig Michael Ciesla and Micah B Yairi. 2012. User interface system and method. (May 15 2012). US Patent 8,179,375.
5. Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation.. In *UIST*, Vol. 13. 417–426.
6. Masaaki Fukumoto and Toshiaki Sugimura. 2001. Active click: tactile feedback for touch panels. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*. ACM, 121–122.
7. Louis H Goldish and Harry E Taylor. 1974. The Optacon: A Valuable Device for Blind Persons. *New Outlook for the Blind* 68, 2 (1974), 49–56.
8. Jiseong Gu, Seongkook Heo, Jaehyun Han, Sunjun Kim, and Geehyuk Lee. 2013. LongPad: a touchpad using the entire area below the keyboard of a laptop computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1421–1430.
9. Chris Harrison and Scott E Hudson. 2009. Providing dynamically changeable physical buttons on a visual display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 299–308.
10. Hiroshi Ishii, Daniel Leithinger, Sean Follmer, Amit Zoran, Philipp Schoessler, and Jared Counts. 2015. TRANSFORM: Embodiment of Radical Atoms at Milano Design Week. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 687–694.
11. Hiroo Iwata, Hiroaki Yano, Fumitaka Nakaizumi, and Ryo Kawamura. 2001. Project FEELEX: adding haptic surface to graphics. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 469–476.
12. Yvonne Jansen, Thorsten Karrer, and Jan Borchers. 2010. MudPad: tactile feedback and haptic texture overlay for touch surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 11–14.
13. Lynette A Jones and Kathryn Ray. 2008. Localization and pattern recognition with tactile displays. In *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, 33–39.

14. Sunjun Kim and Geehyuk Lee. 2013. Haptic feedback design for a virtual button along force-displacement curves. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 91–96.
15. Seung-Chan Kim, Ali Israr, and Ivan Poupyrev. 2013. Tactile rendering of 3D features on touch surfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 531–538.
16. Michael Kraus, Thorsten Völkel, and Gerhard Weber. 2008. An off-screen model for tactile graphical user interfaces. In *International Conference on Computers for Handicapped Persons*. Springer, 865–872.
17. Jaeyeon Lee, Jaehyun Han, and Geehyuk Lee. 2015. Investigating the Information Transfer Efficiency of a 3x3 Watch-back Tactile Display. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1229–1232.
18. Vincent Levesque, Louise Oram, Karon MacLean, Andy Cockburn, Nicholas D Marchuk, Dan Johnson, J Edward Colgate, and Michael A Peshkin. 2011. Enhancing physicality in touch interaction with programmable friction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2481–2490.
19. Shahzad Malik and Joe Laszlo. 2004. Visual touchpad: a two-handed gestural input device. In *Proceedings of the 6th international conference on Multimodal interfaces*. ACM, 289–296.
20. Mays. 2013. Move Over Laptops; Touchpads Are on the Rise in Cars. `https://www.cars.com/articles/2013/11/move-over-laptops-touchpads-rise-in-cars`. (Nov. 12 2013). Accessed: 2016-09-21.
21. Viktor Miruchna, Robert Walter, David Lindlbauer, Maren Lehmann, Regine Von Klitzing, and Jörg Müller. 2015. GelTouch: Localized Tactile Feedback Through Thin, Programmable Gel. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 3–10.
22. Ivan Poupyrev, Tatsushi Nashida, Shigeaki Maruyama, Jun Rekimoto, and Yasufumi Yamaji. 2004. Lumen: interactive visual and shape display for calm computing. In *ACM SIGGRAPH 2004 Emerging technologies*. ACM, 17.
23. Ivan Poupyrev, Jun Rekimoto, and Shigeaki Maruyama. 2002. TouchEngine: a tactile display for handheld devices. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. ACM, 644–645.
24. Denise Prescher, Gerhard Weber, and Martin Spindler. 2010. A tactile windowing system for blind users. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 91–98.
25. Christophe Ramstein. 1996. Combining haptic and braille technologies: design issues and pilot study. In *Proceedings of the second annual ACM conference on Assistive technologies*. ACM, 37–44.
26. Jun Rekimoto and Carsten Schwesig. 2006. PreSenseII: bi-directional touch and pressure sensing interactions with tactile feedback. In *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 1253–1258.
27. Deepak Ranjan Sahoo, Kasper Hornbæk, and Sriram Subramanian. 2016. TableHop: An Actuated Fabric Display Using Transparent Electrodes. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 3767–3780.
28. Martin Spindler, Michael Kraus, and Gerhard Weber. 2010. A graphical tactile screen-explorer. In *International Conference on Computers for Handicapped Persons*. Springer, 474–481.
29. Jessica Tsimeris, Colin Dedman, Michael Broughton, and Tom Gedeon. 2013. ForceForm: A Dynamically Deformable Interactive Surface.. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. 175–178.
30. Laura Winfield, John Glassmire, J Edward Colgate, and Michael Peshkin. 2007. T-pad: Tactile pattern display through variable friction reduction. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*. IEEE, 421–426.