Towards Simulating Push Button Behavior on Touch Surfaces

Figure 1: Implementation of our approach to recreate the input and output characteristics of mechanical push buttons on touch surfaces using a remote tactile interface.

Hendrik Richter
University of Munich LMU
Amalienstrasse 17
80333 Munich, Germany
hendrik.richter@ifi.lmu.de

Matthias Schmidmaier
University of Munich LMU
Amalienstrasse 17
80333 Munich, Germany
schmidmaier@cip.ifi.lmu.de

Abstract

Mechanical push buttons provide various multimodal stimuli before, during and after manual activation. This makes them fast, discriminable and easy to use with virtually no visual or cognitive attention. The transfer of these rich multimodal characteristics to touch based digital interfaces entails complex and costly technology for force sensing. In the paper, we describe a theoretical model, which makes up for the lack of force data by substituting applied input pressure with input time. Subsequently, we demonstrate a working implementation of our model using a remote tactile interface. Finally, we briefly discuss the results gained from 24 user interviews on the feasibility of our approach.

Keywords

touch input, push buttons, remote tactile feedback

ACM Classification Keywords

H5.2: User Interfaces. - Haptic I/O.

General Terms

Experimentation, Human Factors

TEI 2012, Feb 19-22, 2012, Kingston, Ontario, Canada.
ACM 978-1-4503-0541-9/11/08-09.

Introduction

When pressing a mechanical push button, rich tactile cues such as varying forces and changing displacements convey the button's form, current state and function. Mechanical buttons demand virtually no visual or cognitive attention, their feedback allows for typing on a computer keyboard or turning the in-car ventilation knob fast and precisely. Generally, this kind of sensory feedback is referred to as proactive and reactive feedback, i.e. feedback before and after the moment of activation. Non-visual feedback occurring before or after activation can help to significantly reduce the errors made and significantly increase performance [14]. However, these phases of typically tactile feedback before and after input, which we know from interactions with physical objects, do not exist on touch-based surfaces. On common touch screens, touching is equivalent to activation [2]. The eyes-free manual exploration of interactive elements on touch screens is not possible without unintended input. Force sensing as an additional input signal allows for this state of touching without activation. However, the common approaches to sense the force that the touching finger applies to the screen involve the integration of hardware into the interface, which hampers scalability and impedes multi touch. Likewise, researchers are looking into ways to transfer the beneficial effects of haptic feedback characteristics from push buttons to touch interfaces [5]. Various evaluations in the fields of mobile interaction or tabletop research demonstrate the positive effects of additional tactile feedback such as the increase of interaction speed or user satisfaction and the minimization of information overload or errors made [12]. However, on commercially available devices such as mobile phones or tablets, the common tactile stimulus still is a short 'buzz' during virtual button activation. The need for costly and complex hardware for sensing the force of input during the brief moment of touch contact might be the reason for this reduction of tactile richness. The absence of common standards on the design of haptic stimuli is stated as another reason why true tactile feedback for touch interfaces hasn't hit the market yet [15]. In this paper, we propose methods to simplify touch input and multimodal output on interactive surfaces. For touch input, we propose a model that helps to implement push button behavior on touch interfaces without force sensing hardware. We abstracted from measured characteristics of physical buttons. Subsequently, our model describes the substitution of mechanical forces during button activation with dynamic dwell times during touch input. We use this substitution of input conditions for the dynamic rendering of output; i.e. visual, auditory or tactile feedback. For output based on our model, we chose to exemplary implement tactile feedback. We utilize tactile feedback for the augmentation of interaction phases directly before, during and after an interaction. We explored the capabilities of our approach in a usage scenario involving 24 participants and a remote tactile interface (see figure 1).

Related Work

The force of pressure on the screen can be sensed directly or indirectly: For example, sensors under the touch surface directly measure the total amount of pressure on the display [7]. Unfortunately, this only enables single-touch input and does not work well on large-scale surfaces such as tabletops or interactive walls. Some researchers have segmented the display into individually movable elements such as pins, pads and even pneumatic air chambers [4]. Every single ‘tactile pixel’ also senses the force applied to it. As a consequence, the flexibility and visual and tactile resolution of this type of tactile display suffers from severe mechanical constraints. Another way to directly estimate the amount of pressure onto a screen is to analyze the size of the contacting fingertip using capacitive or optical sensing [1]. However, these values differ substantially for every single finger of the hand and for multiple users. For the indirect approximation of input force, Cao et al. [3] propose a virtual force metaphor, in which different amounts of contact area apply to different amounts of input force. Likewise, the indirect use of tangibles as a means of input and output prevents the direct manipulation of the depicted virtual elements [8]. Basically all aforementioned approaches require the integration of hardware such as sensors, electro-magnetic actuators, optical-tracking equipment or even pneumatic devices into the touch display. This makes common pressure sensing on direct touch surfaces hardly scalable, mechanically complex and costly. In 2003, Nashel et al. [11] suggested the use of linger or dwell time during touch input for the estimation of pressure on touch devices without formalizing or generalizing this notion. Additionally, this approach of constant dwell time did not take into account the varying characteristics of mechanical elements such as push buttons.

Recreating Push Button Behaviour

Activating a physical push button is a multimodal experience that can be segmented in five phases: (1) Taking aim and reaching towards the button. (2) By touching the element, we are informed again about the button’s function and current state. (3) When the button is pressed down, varying forces and degrees of displacement provide feedback during the reversible act of pressing. (4) Subsequently, the confirming ‘snap’ can be heard and felt. The force of the button moving back to the starting position acknowledges the action.
(5) Finally, our fingers leave the buttons surface. Depending on mechanical characteristics of the push button, this sequence can be performed in a very short period of time. The movement speed of the button during these phases depends on the force that is applied. The more force is needed, the slower the movement of the button. This essentially means: The harder the button, the slower the action. This in turns means that we may substitute the force of input with the speed of input (which also affects the speed of output/feedback). Our descriptive model is based on this substitution. With this work, we transferred aforementioned phases 2 (manual exploration), 3 (movement) and 4 (confirmation) of mechanical push buttons to flat, interactive surfaces. To get there, we first measured the correlation between the forces applied by the finger to the button’s surface and the button’s resulting displacement. Second, we segmented the resulting force profiles into discrete sections with fixed ratios of force and displacement. Finally, we substituted the required input force in these sections with input dwell time. These dynamic speed variances can be used to design visual, tactile or auditory feedback for virtual push buttons on interactive surfaces. We demonstrate a working implementation and preliminary evaluation of our model using a remote tactile interface.

Physical Button Analysis

In an initial analysis, we measured the relationship between compression force and button displacement for numerous real-world buttons, which differed in

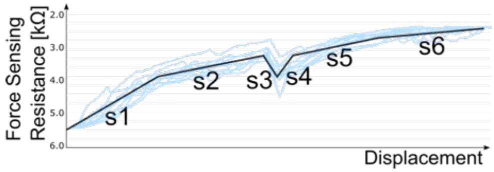

Figure 2: Overlay of measured force diagrams of a basic push button and separation into six linear sections.

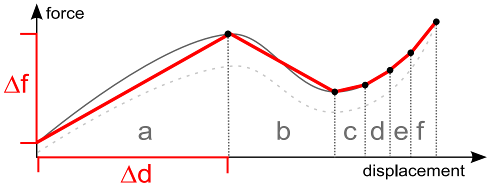

Figure 3: Schematic example for segmentation of force profiles.

characteristic features, such as the required activation force or path length. Following the principle of Nagurka's measurement system [10], we built a device consisting of a stepper motor and a Force Sensing Resistor[1] (FSR sensor) that enabled us to push down a button stepwise and capture the applied force. As the resulting force-displacement curves lacked resolution in both displacement and force values, we additionally measured button presses with an FSR sensor mounted to the fingertip for each button. We translated the FSR values into force in Newton using the sensor's resistance-force curve and averaged the resulting relations for each button.

Deducing the Descriptive Model

We then analyzed the graphs of all measured buttons and identified several key path components or sections, defined by change of slope, which exist in each measured force-path model (see figure 2). By averaging the FSR values for the start and end of each section we determined fixed ratios between FSR value and displacement for each section. Figure 3 describes these findings more formally. By looking at abrupt changes in slope, we can segment the force profile into single sections (here: six sections a to f), each with fixed ratio of force and displacement. For section a, the amount of input force Δf that is needed to achieve displacement Δd is rather high. According to our model, we would substitute this input force with a high amount of input dwell time, resulting in slow feedback for that section of a virtual button. For section f, preceding the button's hard stop, the opposite is true: here, we need a low amount of force to pass through the section, resulting in short dwell times and fast feedback.

Summing up, the substitution for a single section can be described using the following formula:

$\text{dwellTime}_{section}$ = (Δforce + forceStart) * delayFactor	
$\text{dwellTime}_{section}$	duration of section [msec]
Δforce	amount of force for this section [N]
forceStart	force needed to start the button's movement [N]
delayFactor	describes the relation between force and dwellTime

Table 1: Substitution model.

Generally, our approach allows for a user-defined accuracy of approximation: The more individual segments that are defined or known from measurements, the more accurate the force profile can be recreated.

Implementation

In general, tactile feedback during the manual exploration on touch displays can be communicated directly [6] or indirectly using remote tactile interfaces, which move the tactile stimulus from the interacting fingertip to other parts of the body [13]. Remote tactile interfaces avoid the complex integration of potentially numerous tactile actuators into the touch device. Our remote tactile interface prototype consists of a high torque servomotor[2] and a linkage system (see figure 4 and figure 5). The user is resting the non-dominant hand on the device while touching the interactive surface with the dominant hand. Other interfaces with remote tactile feedback provide the stimuli to the wrist or forearm, thus permitting multi touch input [9]. We chose the hand as area of application for evaluational

---

[1] FSR type IEEFSR-150NS

[2] Modelcraft MC-630 MG

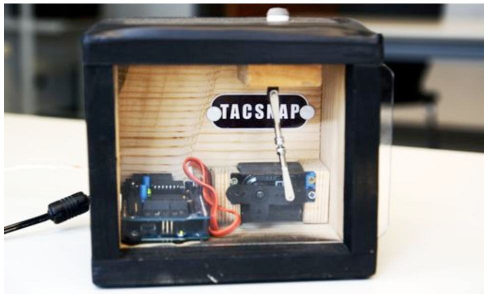

Figure 4: Remote Tactile Interface used in the evaluation.

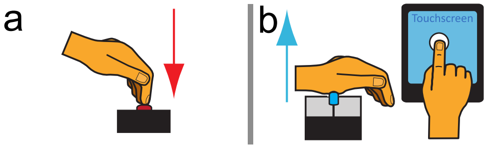

Figure 5: Functional principle of replicating a physical push-button (a) using our remote tactile interface (b).

purposes only, also due to the well-known characteristics of tactile mechanoreceptors in the skin of the human palm. When the dominant fingertip touches a virtual push button, the contact pin of the device pushes against the ball of the user's thumb. Hence, the virtual button's tactile characteristics are transferred to the non-dominant hand in inverted vertical direction to simulate the resistance of a physical button. This vertical implementation allows for the simulation of the skin's increasing deformation resembling the deformation happening during the interaction with a physical button. After activation, the contact pin retreats. The pin's displacement matches the physical button's displacement we measured beforehand. The servomotor moves the prototype's contact pin in small discrete steps of 0.25 mm. By varying the delay between these single steps for each section, we defined the section's total duration or playback speed. The button characteristics depicted in figure 2 are simulated as follows: The user touches the virtual push button on the touch screen, resting the other hand on the feedback device. For section s1, needing a high force, the tactile feedback device moves upwards slowly and steadily increases pressure on the user's hand. The interface's displacement matches the replicated button's displacement. For section s6, which needs less force in the real world, the device moves much faster. In a preliminary usage scenario, we asked 24 computer literate participants (8 female) to test and compare 3 physical push buttons (see figure 6) to their virtual representations using the remote tactile interface for touch surfaces. The virtual push buttons on the touch screen resembled the real buttons in size and color. The visual representation of a moving mechanical button was designed using the model; the button area darkens with variable velocities while being pushed down. Participants were free to try and compare physical and virtual buttons. They also were asked to adjust the delayFactor (which describes the relation between substituted input force and resulting dwellTimes for the visual and tactile feedback) to their needs using a widget on the touch screen.

Figure 6: Three typical physical push buttons.

In guided interviews after the test phase, users were asked about their experiences. When asked about the stimuli that were designed using our substitution model, users stated that the "buttons are easily discriminable" based on displacement and speed variances. Users were also able to perceive various "forces and pressure points" during direct touch input. Due to this additional tactile sensation, the virtual buttons felt "similar to the physical buttons". When asked for improvements, they suggested adding audio feedback. Most important, some users stated that "the speed sometimes doesn't fit". This shows the importance of more formal evaluations to analyze general correlational values between force and dwell time.

Conclusion and Future Work

In this paper, we described our work in progress to recreate the input and output behavior of mechanical push buttons on arbitrary interactive surfaces. We propose the use of dynamic input dwell times and varying output speed as a substitute for measuring finger pressure on touch surfaces. We explored our approach using a purpose-built remote tactile interface in a preliminary evaluation. Three evaluations will further substantiate our approach: At the moment, we work analyzing the correlation of input force and variable dwell times. The touch screen in the planned evaluation shows several visually identical buttons. The section speed of output is different for each button, ranging from low to high. Asking the participants to indicate the button that is “the hardest to press” could allow for further insights into the user’s acceptance of our substitution. Second, mechanical buttons with known force path characteristics will be used in another study. By measuring the speed of pushing during a task, we could identify defined correlations between input force and button movement speed. Finally, we could validate the found correlations again by letting participants compare mechanical push buttons and their virtual representations, which are based on our model. Our next step is to assay the model’s representation in the visual and auditory modality and whether individual user customization is necessary. The design of new button-like interfaces with enriched visual, tactile and auditory feedback on touch surfaces is possible by designing new path-force-speed correlations which go beyond the ones found in real world mechanical buttons.

References

[1] Benko, H., Wilson, A. D., and Baudisch, P, Precise Selection Techniques for Multi-Touch Screens, Proc. CHI 2006, 1263-1272.

[2] Buxton, W., A three-state model of graphical input, Human-computer interaction-INTERACT, 1990, 449–456.

[3] Cao, X., Wilson, A.D., Balakrishnan, R., Hinckley, K., Hudson, S.E. ShapeTouch: Leveraging contact shape on interactive surfaces. Proc. TABLETOP 2008, 139-146.

[4] Harrison, C. and Hudson, S. E. Providing Dynamically Changeable Physical Buttons on a Visual Display. Proc. CHI 2009, 299-308.

[5] Hoggan, E., Brewster, S.A., and Johnston, J. Investigating the Effectiveness of Tactile Feedback for Mobile Touchscreens. Proc. CHI 2008, 1573.

[6] Iwata, H., Yano, H., Nakaizumi, F., and Kawamura, R. Project FEELEX: adding haptic surface to graphics. Proc. SIGGRAPH '01, 476.

[7] MacKenzie, I. S., Sellen, A. and Buxton, W., A comparison of input devices in element pointing and dragging tasks. Proc. CHI 1991, 161-166.

[8] Marquardt, N., Nacenta, M., Young, J., Carpendale, S., Greenberg, S., and Sharlin, E. The Haptic Tabletop Puck: Tactile Feedback for Interactive Tabletops. Proc. ITS 2009, 93-100.

[9] McAdam, C., and Brewster, S. Distal tactile feedback for text entry on tabletop computers. Proc. of BCS-HCI '09, 504–511.

[10] Nagurka, M.L. and Marklin, R.M., Measurement of Stiffness and Damping Characteristics of Computer Keyboard Keys, ASME Journal of Dynamic Systems, Measurement and Control, Vol. 127, 2005, 283-288.

[11] Nashel, A. and Razzaque, S. Tactile virtual buttons for mobile devices. In CHI EA 2003, 854-855.

[12] Poupyrev, I., Maruyama, S., and Rekimoto, J. Ambient touch: designing tactile interfaces for handheld devices. Proc. UIST 2002, 60.

[13] Richter, H., Blaha, B., Wiethoff, A., Baur, D. and Butz, A. Tactile feedback without a big fuss: simple actuators for high-resolution phantom sensations. Proc. UbiComp 2011, 85-88.

[14] Sellen, A., Kurtenbach, G. and Buxton, W., The role of visual and kinesthetic feedback in the prevention of mode errors. Proc. of INTERACT 1990, 667-673

[15] Wright, A. The touchy subject of haptics. *Commun. ACM* 54, 1 (January 2011), 20-22.