Exploring Continuous Pressure Input for Mobile Phones

Edward C. Clarkson, Shwetak N. Patel, Jeffrey S. Pierce, and Gregory D. Abowd
College of Computing and GVU Center
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332-0280
{edcclark, shwetak, jpierce, abowd}@cc.gatech.edu

ABSTRACT

The input capabilities of mobile phones are limited by their physical form factor. Approaches to augmenting those capabilities that expand the input space without negatively impacting size or weight are particularly desirable. We propose adding simple pressure sensors under the keypad buttons to provide multiple channels of continuous pressure input. Pressure input supports a larger and more interesting interaction space without some of the unusual or unwanted qualities of some other approaches. We describe an implementation of our pressure-augmented system and show a number of interaction techniques, some old and some new, facilitated by continual pressure. We contrast these techniques with previous sensor-augmentation devices and highlight notable differences and advantages.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces – Interaction styles, Input devices and strategies.

Keywords: Mobile computing, mobile input, pressure sensing, force sensing.

INTRODUCTION

Mobile computing devices are inherently limited in their capabilities relative to their larger, stationary cousins. Because of this, HCI designers and researchers have explored a wide variety of techniques for improving the mobile user experience. Size constraints and form factors (e.g., the standard mobile phone keypad layout) place awkward demands on the implementations of input schemes on these platforms. Handsets are also quite limited in their sensory capabilities—generally they are restricted to fewer than 40 binary state (bi-state) buttons. As a result, input (either arbitrary or text) on these devices is problematic and significantly slower than on their larger counterparts.

As such, this has been a rich area for exploration in research, leading to a variety of approaches for mobile input. One has been to augment handsets with additional simple sensors. Various uses of simple touch sensing on mobile devices has been explored in a number of works, often in combination with other augmented sensors [2,3,4]. Rekimoto's PreSense system uses a touch-sensitive keypad, transforming buttons from bi-state to tri-state sensors [7]. This enables what he calls "previewable user interfaces." Other work has used modified hardware to implement tri-state input mechanisms, which has the additional benefit of haptic feedback [11].

Figure 1 – Our pressure-augmented mobile handset (left). Pressure is sensed via a capacitive sensor under each button.

Taking the bi- to tri-state progression to its extreme, there are infinite-state sensors that allow the specification of continuous values. Continuous sensing can also reduce to bi- or tri-state input as well, simply by partitioning the continuum into the desired number of regions. Tilt sensing is one such continuous sensing method that has been widely explored on mobile devices. Some frequent uses have been for scrolling [2,3,8], list selection [2], and text entry [5,10]. But tilting has its limitations. It requires gross manipulation of the input device, which can interfere with observation of the output and can be influenced by external events (e.g., motion during use). Such manipulations can also be socially undesirable, since users may not want to attract attention to themselves while they use their device.

Pressure is another promising candidate for an input modality that can yield continuous input data. Blaskó and Feiner have experimented with pressure-sensitive strips, but their work was concerned with menu interactions [1]. Ramos *et al.* [9] examined some basic properties of pressure input using a Wacom pen tablet. Their central findings indicate users can reliably select no more than six discrete ranges of pressure and perform selections best by dwelling or quick-releasing the input stylus. They also define and illustrate a number of interaction techniques based on pressure input.

We believe pressure input has a number of appealing properties that make it suitable for use with mobile phones and mobile phone keypads. The hardware requirements for pressure sensing are inexpensive and require no changes to key layouts or form factors, leveraging users' familiarity with existing devices. In addition, pressure sensors can be made sensitive enough to detect the touch of a physical button before it is depressed enough to click. This allows previewable user interface interactions with the additional ability for the user to specify (via pressure) a scale or magnitude for either a previewed action or the action.

Pressure, like tilt sensing, provides continual-state input that affords smooth interaction techniques such as scrolling. However, unlike tilt sensing, it does not necessitate additional gross physical movements that might inhibit observation of display output, attract undesirable attention, or be strongly affected by outside movements.

Thus, continuous pressure sensing provides exploration of classes of interaction techniques not possible or difficult with other (e.g., tilt or motion) sensors. This work proposes and describes several classes of interaction techniques that are possible using pressure-sensitive mobile handsets. We illustrate those classes by describing a number of interaction techniques in each class and describe our implementations of some of those techniques on our augmented prototype handset.

PROTOTYPE PLATFORM HARDWARE DESIGN

Our prototype design is based on a standard Motorola i730 handset (see Figure 1). We chose a thin-film piezoresistive force sensor (Tekscan FlexiForce) so we could insert the sensors underneath the buttons without altering the handset substantially. The FlexiForce sensors are .208 mm thick with a 9.53 mm diameter circular sensing area. Their response time is rated at < 5 μs with a maximum load of 110 N (25 lbs). The sensor measures force by the capacitor discharge time. Since the resistance of the capacitive sensor decreases with increasing load, larger forces discharge the capacitor faster (and the value of the sensor output). Consequently, the smaller discharge times from larger forces entail a proportional relationship between load and sampling frequency.

The FlexiForce sensors fit between the keys and their contact points, allowing the phone to function normally. However, the sensors are controlled from a Basic Stamp 2p (BS2p) microcontroller, which processes the output from the sensors and relays the data to a host computer (which can also be relayed to the phone itself) via a standard 9-pin serial cable. The BS2p is a single-threaded chip; we have implemented a poll-and-focus algorithm to maximize data sampling resolution while minimizing latency. The BS2p polls each sensor at approximately 3 Hz while all sensors are unloaded. As soon as a sensor detects a nonzero force, the chip polls only that sensor until the sensor is completely unloaded. This approach allows the relatively modest BS2p to process the sensor data much faster when loaded, but with the tradeoff that our prototype cannot process simultaneous sensor presses. Future iterations of the prototype can be improved by using a more advanced analog-to-digital converter, which would allow both concurrent and high-sampling rate processing.

INTERACTION TECHNIQUE CLASSES FOR PRESSURE-AUGMENTED MOBILE COMPUTING

We believe that pressure as an input modality is widely applicable to interaction techniques on mobile devices, especially those that use the standard phone keypad. We have developed a number of interaction techniques that we have classified into three broad classes. These classes are not intended to be exhaustive (or necessarily exclusive); but the variety does suggest promise for this added modality to a mobile phone. Our techniques also follow relevant guidelines suggested by Ramos, *et al.*: they provide real-time, continuous visual feedback and use no more than 6 divisions of the pressure range for discrete selection tasks.

Direct Specification

This class refers to pressure interactions which directly indicate some parameter of the user's action. Interaction techniques of this type are ones which gain little advantage from or cannot be previewed. Examples which are trivial to implement include setting device preferences such as ringer volume and screen brightness/contrast.

We have created an input technique called *PressureTap*, which allows for text input based on varying levels of pressure. Partitioning the pressure distribution allows typing to occur normally with only abnormally heavy presses producing the alternate letters, such as the numerical value of that key. This interaction technique is reminiscent of a software version of the 'pop-through' mouse [11]. Similarly, heavy force during typing produces upper- instead of lowercase letters. In addition, a user can completely bypass MultiTap and use just PressureTap for text input (see Figure 2). We have conducted some initial user studies on using PressureTap as a complete replacement for MultiTap. Those studies indicate that PressureTap on our current hardware is not as fast as MultiTap; however, we expect that enhancements to our hardware may yield better results.

Scrolling is a class of operations which is of particular importance on mobile devices because of their limited screen

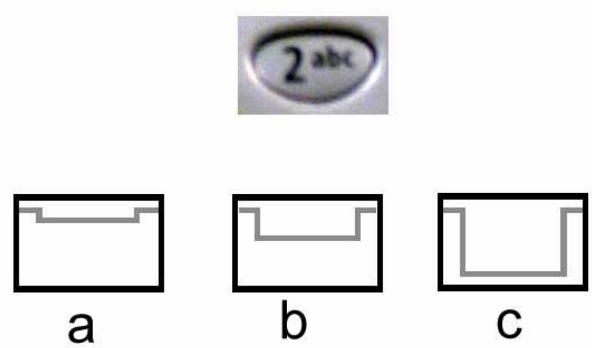

Figure 2 – The PressureTap interaction technique. The key is depressed slightly (light force) for selection 'a', more for 'b' and most for 'c'.

space. We can use pressure to effect scrolling such that the user has control over the speed of the interaction. Our implementation of this technique, called *PressureScroll,* uses map scrolling as an example domain, although other areas such as web browsing are readily available. We use the 1-9 buttons on the keypad like a directional pad: 1 moves the view area up and to the left; 2 moves the view up, and so on. The scroll speed is proportional to the force detected on the button. The view from the start of the scroll operation is outlined in red to provide context.

We have also implemented an interaction technique for three-dimensional (3D) navigation called *PressureNav*. We can manipulate 3D object models in free-space using the keypad buttons, mapping pressure to the speed of the object manipulations. Similarly, we use this technique to navigate a 3D environment; pressure is used to control the velocity of the movement throughout the virtual space.

Although not traditionally used on mobile devices, techniques for 3D navigation are widely used in gaming applications, which is a fast-growing area in the mobile computing domain. For example, the implementations described above could be useful in a wide variety of interactive games such as those in first-person shooter, racing, or martial arts combat genres. Although tilting could be used instead, the necessity of moving a mobile device's display to register tilt input is particularly cumbersome for gaming applications, and the number of tilt axes is practically limited to two. Pressure affords a more stable display and provides as many continuous input axes are there are pressure-sensitive buttons.

Previewable Interactions

Extending traditional keypad buttons into a tri-state sensing platform can result in types of interaction techniques that allow the interface to identify the prospective action explicitly, an important attribute when an action cannot be easily undone. In the PreSense system [7], Rekimoto uses capacitive sensing keypad to detect touch. This extends the buttons to a tri-state sensing platform, which enables what he calls 'previewable user interfaces.' These types of interaction techniques allow the interface to identify the prospective action explicitly, an important attribute when an action cannot be easily undone. Our prototype also allows such previewable interaction techniques since the capacitive sensors we use are sensitive enough to detect forces much less than those necessary to activate the button. Thus, in addition to the techniques described for PreSense, we can also make use of the continuous input channel provided by our sensors.

There are a variety of techniques involving zooming, for example. One of the PreSense examples is a map display application; it divides the screen into a grid, each cell of which corresponds to a key. Each key then zooms to a particular region of the screen that is previewed on touch. We have implemented one possible extended version of this technique which we call *PressureZoom* (see Figure 4). Our hardware, in addition to previewing the zoom level, can also vary the magnification level within the grid. The exact zoom level is directly specified by the pressure on the appropriate key. The user can then smoothly zoom in and out of that grid block by modulating the amount of pressure on the button. Another possibility is for exact zoom level to be previewed along with the overall grid block. Instead of initially zooming to the corresponding block, pressure values can modulate the prospective magnification area (i.e., the highlighted red box shown in Figure 4 left) before actually performing the zoom operation.

There are other possible techniques combining both direct specification scrolling and previewable zooming. The 5 key is unused in the scrolling interaction described above. That key can be assigned to a previewable zoom function which operates similar to either of the implementations described above, but over the entire view rather than a section of the screen.

All of these zooming operations represent interaction techniques which would be difficult or impossible to implement without a continuous-state input channel. Moreover, the use of pressure input in particular is valuable because the techniques would be much more difficult and unintuitive with the use of other types of continuous input such as tilt.

Affective Input

Pressure also affords the exploration of less traditional interaction techniques. Affective computing [6] is the study of computing which "relates to, arises from, or deliberately influences emotions." The force of one's touch can often

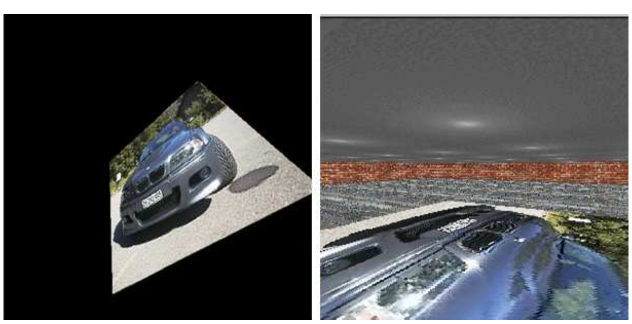

Figure 3 – Two scenes that are manipulable via the PressureNav technique. The object at left is a flat picture being rotated in free space; the scene at right is a navigable 3D environment.

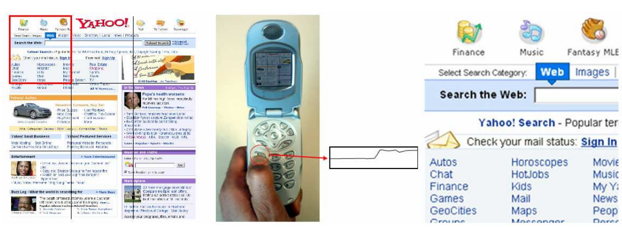

Figure 4 – An illustration of the PressureZoom interaction technique on a web browsing task. The screen is divided into blocks that correspond to keypad buttons. Touching a key highlights the corresponding grid area, while pressure determines the amount of zoom within the grid block.

indicate such emotional states, such as urgency or desire. We have noticed anecdotally that many computer game players instinctually press buttons harder when they want to execute an action quickly or urgently—for example, accelerate their car faster. Some modern game systems leverage this natural behavior and have pressure sensitive buttons (e.g., Sony DualShock 2) that respond to this behavior, actually accelerating according to force levels.

In mobile computing, there are also situations in which the importance of an action might be inferred from user behavior. It is all too common for mobile phones to ring at inopportune times, the result of which generally has their owners frantically punching buttons in an attempt to silence them. But turning off the ringer completely requires users to also remember to turn it back on at a later time. We can use pressure to enable a *Snooze* interaction technique for specifying ringer silence intervals. Button force can be mapped to the length of time for the phone to remain silent, leveraging the fact that more forceful presses are likely to be more important to the user.

Text messaging is also a widely-used application that can be enhanced via affective techniques. Using pressure again as a proxy for urgency or importance, we can modify the presentation of text according to the typing intensity its source. The local system can then respond to messages or text typed more forcefully by prioritizing its delivery across the network (if applicable). Remote systems can choose to accentuate more urgent messages by presenting such messages using a larger font size or weight or to alert users to messages beyond a certain importance threshold.

FUTURE WORK

In the immediate future, comparing these interaction techniques against those without pressure is an obvious next step. We would also like to investigate what theoretical justification there might be for pressure-based approaches—does GOMS or other similar low-level human performance models indicate a significant advantage for pressure-based interactions over their non-pressure counterparts? We have also made a supposition that the data provided by Ramos, *et al.* for stylus pressure translates well to thumb- or finger-pressed buttons. Verification or refutation of this assumption is necessary.

On the hardware side, we are exploring a number of improvements to our current apparatus. The addition of haptic feedback (through short bursts of the phone's vibration motor) is likely to be especially beneficial for users. As we have also mentioned, our system is limited to processing button presses serially, and there is some latency introduced by the polling algorithm demanded by the limitations of the BS2p. Our expected transition to a more advanced processor that can operate on an interrupt-driven system will improve the latency and speed of our system.

CONCLUSIONS

Many approaches to improving input on mobile devices augment the bi-state norm for mobile keypad input buttons. We have suggested the use of pressure, a continuous-state input channel, as a means of enhancing mobile devices. It presents a number of useful characteristics over both tilting and simple touch sensing, two sensory paths commonly used in mobile input research.

Our prototype hardware equipment provides a platform on which to implement a number of pressure-based interaction techniques. We have specified three classes of such techniques, some of which borrow from existing research areas, and implemented a number of interactions to exemplify those classes and to use in future evaluations of the efficacy of pressure-based interaction on mobile devices.

ACKNOWLEDGMENTS

The authors thank Motorola, and in particular Joe Dvorak of iDEN Advancing Technologies Group, for the donation of the iDEN handsets and Nextel service for this research. We also thank Khai Truong for his insightful comments throughout the course of this work.

REFERENCES

1. Blaskó, G. and Feiner, S. Single-handed Interaction Techniques for Multiple Pressure-sensitive Strips. In *CHI '04 Extended Abstracts*, pp. 1461-1464.
2. Beverly, H.L., Fishkin, K.P., Gujar, A., Mochon, M., and Want, R. Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. In *Proc. of CHI '98*, pp. 17-24.
3. Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. Sensing Techniques for Mobile Interaction. In *Proc. of UIST '00*, pp. 91-100.
4. Hinkley, K., Horvitz, E. Toward More Sensitive Mobile Phones. In *Proc. of UIST '01*, pp. 191-192.
5. Partridge, K., *et al.* TiltType: Accelerometer-supported Text Entry for Very Small Devices. In *Proc. of UIST '02,* pp. 201-204.
6. Picard, R. *Affective Computing*. MIT Press, Cambridge, MA, 1997.
7. Rekimoto, J., Ishizawa, T., Schwesig, C. and Oba, H. PreSense: Interaction Techniques for Finger Sensing Input Devices. In *Proc of UIST '03,* pp.203-212.
8. Rekimoto, J. Tilting Operations for Small Screen Interfaces. In *Proc. of UIST '96,* pp. 167-168.
9. Ramos, G., Boulos, M. and Balakrishnan, R. Pressure Widgets. In *Proc. of CHI '04,* pp. 487-494.
10. Wigdor, D. and Balakrishnan, R. TiltText: Using Tilt for Text Input to Mobile Phones. In *Proc. of UIST '03,* pp. 81-90.
11. Zeleznik, R., Miller, T., and Forsberg, A. Pop Through Mouse Button Interaction. In *Proc. of UIST '01,* pp. 195-196.