VibPress: Estimating Pressure Input Using Vibration Absorption on Mobile Devices

Sungjae Hwang
GSCT, KAIST
best@kaist.ac.kr

Andrea Bianchi
Computer Science,
Sungkyunkwan University
abianchi@skku.edu

Kwang-yun Wohn
GSCT, KAIST
wohn@kaist.ac.kr

ABSTRACT
This paper introduces *VibPress*, a software technique that enables pressure input interaction on mobile devices by measuring the level of vibration absorption with a built-in accelerometer when the device is in contact with a damping surface (e.g., user's hands). This is achieved using a real-time estimation algorithm running on the device. Through a user evaluation, we provide evidence that this system is faster than previous software-based approaches, and as accurate as hardware-augmented approaches (up to 99.7% accuracy). We also provide insight about the maximum number of pressure levels that users can reliably distinguish, reporting usability metrics (time, errors, and cognitive loads) for different pressure levels and types of gripping gestures (press and squeeze).

Author Keywords
Pressure Input; Haptics; Mobile; Accelerometer;

ACM Classification Keywords
H.5.2 [User Interfaces]: Haptic IO.

INTRODUCTION
Current mobile devices are equipped with a variety of sensors (e.g., capacitive touchscreens, magnetometers, microphones, accelerometers) offering numerous potential input channels for expressive interaction techniques [2, 8, 12]. Researchers have leveraged these capabilities to create new input systems that enhance user experience [e.g., 4, 6, 7]. More recently have suggested how mobile devices could also benefit from sensing pressure input (e.g., squeezing or tapping on a device with different strength levels). In fact, pressure input can free users from spatial restrictions and repetitive movements (e.g., flicking gestures could be replaced with pressure-sensitive interactions [10]). This is suitable for one-handed mobile interaction, adds a degree of freedom to touch locations [1], and can also provide for rich contextual selections [11]. Previous work on pressure input techniques for mobile devices either introduced specialized sensing hardware to exactly measure the force applied by

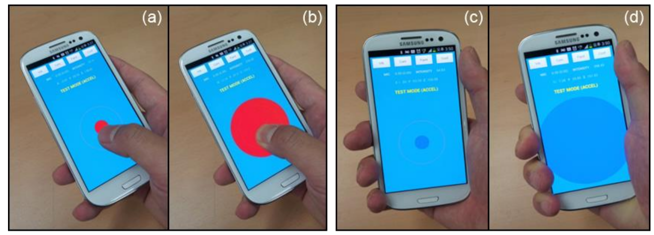

Figure 1. *VibPress* estimates light and heavy pressure inputs for the pressing gesture (a, b) and the squeezing gesture (c, d).

users [2, 3, 10, 11] or relied on software to estimate the pressure from sensors commonly available on mobile devices [1, 4, 6, 7, 8, 9, 12]. Following the hardware augmentation approach, GraspZoom [10] allows users to zoom and scroll with their mobile devices using a force-sensitive resistor (FSR) located on the back of the device. Brewster and Hughes [2] investigated the benefits of using a pressure-sensitive touchscreen for text entry. They used a pressure property to choose lower and upper case letters and showed the pressure-augmented text entry technique was faster and more accurate than the standard one. Cechanowicz et al. [3] showed the advantage of augmenting a computer mouse with two pressure sensors, allowing multiple levels of discrete selection modes in desktop applications. Similarly, Ramos et al. [11] showed the advantage of using pressure-sensitive styli on tablets to operate multi-state graphical widgets. They recommended that widgets should employ a maximum of six discrete levels for adequate visual feedback on pressure input. Specialized hardware offers reliable and fast (or real-time) input pressure readings, but the disadvantage is that such hardware is still not widely supported in current, commercially available mobile devices. It could also represent additional production costs for device vendors and additional maintenance costs for users.

As a response to these limitations, many authors have attempted to estimate input pressure with software, using readings from sensors already available on most mobile devices. Iwasaki et al. [9] used the accelerometer embedded in a notebook to detect the pressure with which users typed keys, creating a richer typing experience. ForceTap [6] uses patterns from accelerometer readings to estimate the force with which the screen of a mobile device is tapped, identifying strong or gentle taps. Hwang et al. [7] used a mobile device's internal microphone to estimate the pressure exerted with a stylus while drawing on the device

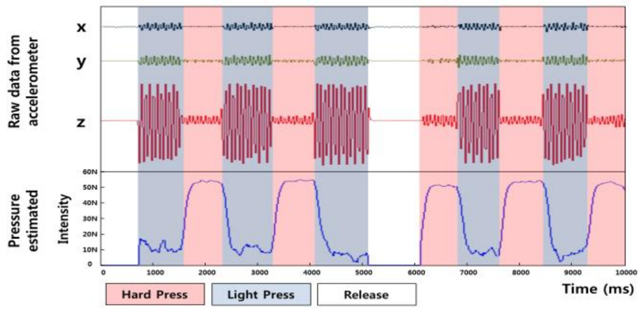

Figure 2. Accelerometer data alone can provide a reliable estimate for vibration absorption and input pressure.

screen and the strength with which an acoustic-based button (e.g., the microphone pinhole) is tapped [8]. Boring et al. [1] leveraged current capacitive screens on mobile devices to determine the contact size of human fingers on a screen, while Strachan et al. [12] used muscle tremors (subtle motions proportional to the force with which the user's finger presses the screen) as a coarse proxy of pressure. These approaches successfully estimated the input pressure using only inertial sensors on mobile devices. However, they do not continuously measure pressure applied by the user [6, 7, 9], the interaction area is fixed and limited by hardware settings (pinhole location) [8], and an additional calibration process is needed [1, 12]. Finally, Goel et al. [4] proposed GripSense, a pressure input system based on observing variations in the gyroscope readings when the device's inertial vibration motor is pulsed: This damping effect is used in combination with muscle tremors and the size of the fingers contacting the screen to estimate the input pressure. GripSense works by feeding a machine-learning algorithm with the readings from all of these variables and extracting an estimation of pressure and hand posture, assuming a maximum of three pressure levels. However, due to the computational and training phase requirements of this machine-learning technique, the estimation requires up to 10 seconds [4]. This paper builds on previous work by (1) suggesting a light-weight variation of Goel's method that somehow improves the time performance for analogous error rates [4, 6, 8]. Moreover, (2) we also present the results of a user study which reports insights on the maximum number of pressure input levels for *VibPress* that users can accurately distinguish using two different types of hand gestures (just noticeable differences). In the following sections, we introduce our technique, describe a prototype, and discuss the user study.

VIBRATION ABSORPTION AS PROXY FOR PRESSURE

This paper argues that the amount of pressure on a mobile device can be approximated by using an accelerometer to measure the spatial displacement generated when the internal vibration motor vibrates. In fact, depending on the pressure applied to the device, the amount of vibration absorbed by the user's hands varies, resulting in noticeable differences in accelerometer readings (Figure 2). Leveraging on this physical property, we can build software that measures the amount of vibration absorption to reliably and quickly estimate the input pressure on the device screen or external case, triggering different actions accordingly.

The main difference from similar previous work [4] is in the way pressure was estimated: While previous approach collected multiple features (muscle tremors, vibration damper, finger contact size on the screen) and fed them into a PC running a non-real time machine-learning algorithm, our approach relies on the accelerometer readings of the damped vibration as a proxy for real-time input pressure. The advantage of our approach, compared to systems that base their estimations on machine-learning techniques, is that it is lightweight (possibly could work on embedded low-powered devices with small screens, such as digital watches) and reduces recognition time. Our approach also works well with two types of gestures (Figure 1): with the *press* gesture, users tap with their fingers and exert pressure on the screen. With the *squeeze* gesture, the user holds the device in one hand and squeezes the sides, with no need to touch the capacitive screen anywhere avoiding occlusions.

IMPLEMENTATION

VibPress was implemented as Java software for the Samsung Galaxy S3 smartphone, running Android OS 4.0.4. When a user touches a button on the capacitive screen, *VibPress* activates the internal linear vibration motor with a continuous pulse at maximum amplitude (a feedback mechanism similar to that used in smartphone keyboards) and simultaneously starts sampling values for each of the three axes at 100Hz using the internal accelerometer. The data buffer is then passed through a low-pass filter with a 50ms window size to reduce sampling noise, and the three-dimensional Euclidean distance between the last reading and its predecessor is computed and normalized. These three values reflect the amount of vibration along with three directions absorbed by the user's hand, and we used the sum of these values to reach a good estimation of the amount of pressure exerted on the device screen. With this setup, the performance of *VibPress* was not seriously degraded when users were walking or moving. We were able to empirically sense pressure up to a maximum of about 50N (with a lab scale), regardless of the type of gesture. Different levels of input pressure are graphically represented on the screen in various forms, depending on the application. Finally, our demo software (Figure 1) displays a red circle with a diameter proportional to the pressure applied.

EVALUATION

To investigate the feasibility and accuracy of *VibPress*, we carried out a pilot and a usability study. The goal of the pilot study was to establish the average minimum and maximum pressure that user could exert on the mobile device in order to derive an appropriate estimation of the maximum number of distinguishable pressure levels. The goal of the usability study was to evaluate how accurately and how fast people could select any of these pressure levels.

Pilot

4 participants (2 women) aged between 26 and 33 (average 29.5, SD 3.1) were asked to hold, in the palms of their dominant hands, a mobile device running *VibPress*, and sequentially apply maximum and minimum pressure to the center of the screen ten times with their thumbs. Every press event took 1.5 seconds, of which we discarded the first 500ms to allow for pressure stabilization. The accelerometer's output for vibration absorption was recorded in a log file stored on the device. Participants then repeated the experiment using the squeeze gesture. In this case, however, participants held the device with the dominant hand and activated the vibration motor for 1.5 seconds by clicking a button with the other hand. The entire experiment took about 10 minutes. Logged data were analyzed with a paired t-test and revealed statistical differences ($p<0.01$) across the two gesture configurations. In particular, the press gesture resulted in a uniform damping of all the vibrations resulting in robust input pressure, while the squeeze gesture resulted in slightly noisier data but a greater intensity range (higher maximum and lower minimum). We isolated a common area (the values between the common minimum and maximum intensities), which we divided by the average variance to estimate the theoretical number of distinguishable pressure levels for both the press and squeeze gestures. We divided this interval on a logarithmic scale, following the same method used in previous work [7] to determine intervals based on just noticeable differences across participants. *VibPress* allows up to six distinct pressure levels, regardless of the hand-gripping type. The following usability study aims to verify these results.

Usability Study

The goal for the usability study was to test the input performance (time, errors, and cognitive load) for different pressure levels and hand gripping gestures derived from the pilot. We tested *VibPress* for two conditions: pressure levels (2, 4 or 6 pressure levels, referred to as L2, L4, L6) and hand gestures (press and squeeze). Similarly to previous work [8, 11], the pressure levels for each group were equally sized, ranging from the minimum and maximum pressure values obtained from the pilot study (0–50N) and graphically represented on the device screen as columns with 2, 4 or 6 equally distanced, adjacent vertical boxes (Figure 3). At every input trial, the system presents a target level by graphically highlighting one of these boxes.

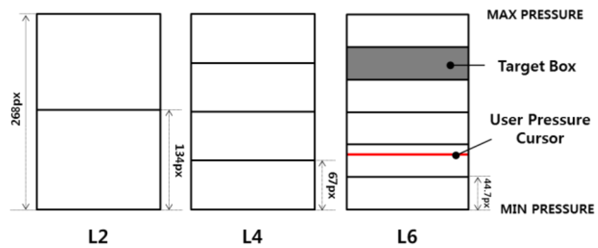

Figure 3. The graphical interface used for testing pressure input with 2 levels (L2), 4 levels (L4) and 6 levels (L6).

The user then taps and holds a finger on the device screen, while a cursor moves up and down proportionally to the pressure exerted (position control). The bottom of the column is mapped to the minimum pressure and the top to the maximum. The user selects a specific target by keeping the cursor within the box for one second. We recruited 12 volunteers (6 women) among students and researchers aged between 24 and 33 (average 27.5, SD 2.5). 11 of the participants had a right-dominant hand, and 8 reported that they were familiar with haptic input interfaces. After we collected demographics, the participants were asked to familiarize themselves with the device for a maximum of five minutes. We then asked users to perform 36 successful targeting trials (the first 12 discarded as practice) for each pressure level (L2, L4, L6) using the press posture, followed by a NASA TLX [5] questionnaire for measuring cognitive load. We then repeated the same experiment using the squeeze gesture condition. In this case, the participants used a button to trigger the vibrations. The experiment was then concluded with an informal interview. The experiment took approximately one hour. It was balanced following a Latin square design, and we ensured that every target level was evenly represented, collecting data for a total of 288 trials per hand gesture type. For each pressure level, we recorded measures of performance for the selection time of successful trials — the time required for users to correctly input a pressure level inclusive of one second to trigger the selection. We also recorded the number of wrong selections and the variations of the pressure within the targets.

RESULTS AND DISCUSSION

Results for input and navigation time, errors, and cognitive load are shown in Figure 4. A 2 × 3 two-factor ANOVA with repeated measures on both factors revealed significant differences for input time across pressure levels ($F(2,11)=126.6$, $p<0.01$) and hand posture ($F(1,11)=6.3$, $p<0.05$), and an interaction was found ($F(2,11)=5.8$, $p<0.01$). Overall error rates for levels L2, L4, and L6 are,

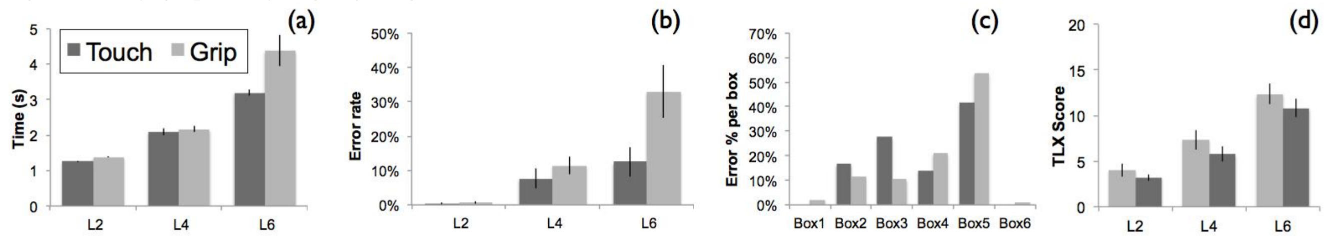

Figure 4. Metrics for selection times (a), error rates (b), percentage of error distribution among the selection levels for L6 (c) and users' cognitive workload with a TLX (d). Low is better.

respectively, 0.3%, 7.6% and 12.5% for the press gesture, and 0.7%, 11.5%, and 33% for the squeeze posture (Figure 4.b). A two-factor ANOVA revealed significant differences for errors across pressure levels ($F(2,11)=25.1$, $p<0.01$), hand postures ($F(1,11)=5.1$, $p<0.05$), and their interaction ($F(2,11)=3.6$, $P<0.05$). Finally, we found differences across pressure levels for the overall cognitive workload measured with the TLX ($F(2,11)=84.5$, $p<0.01$) but not for the different hand postures, or for their interactions.

Our findings for completion time, errors, and cognitive load significantly vary depending on the number of pressure levels presented. When a binary input (L2, light or heavy press) was required, users were, unsurprisingly, the most accurate and fast, with an error rate of less than 1% and completion time of around one second. Any subsequent increment of the two additional pressure levels took at least 1.5 times longer and caused considerably more errors than in previous groups. The distribution of errors was, however, uneven, and most errors happened in the central targets (Figure 4c), rather than at the extremes: A user stated that “the second and third box from the bottom were especially challenging in L6.” This suggests that pressure levels could be subdivided unevenly, with different-sized ranges for targets. The error rate is also reflected in the results from the TLX (cognitive load is directly proportional to the number of pressure levels) and the post-hoc interviews, in which users expressed their frustration with using L6. When comparing these results with those from previous work for analogous dwell-based targeting tasks, *VibPress* shows selection times similar to those obtained with specialized hardware [3, 11], but faster [4] and with fewer errors than software pressure techniques [4, 6, 8]. A possible explanation for this improved performance is the simplicity of the pressure-estimation approach, a hypothesis corroborated by the user interviews (“*VibPress* is very intuitive”). In terms of usability, users responded positively to *VibPress*, expressing enjoyment and quick adoption, as also represented by the TLX findings. On the flip side, some participants reported that “continuous vibrations are somehow disturbing.” and mentioned “the need to put attention on the task when targeting small boxes.” Overall, participants agreed that for L2 and L4, the performance and the usability level were satisfactory, and despite the accuracy level, users preferred the squeeze gesture to the press one, as they felt it took less physical effort. Finally, many users spontaneously suggested possible application scenarios (e.g., rejecting phone calls or focusing and taking pictures by squeezing the phone) and emphasized the usefulness of putting *VibPress* in commercial applications.

CONCLUSION AND FUTUREWORK

This paper introduced *VibPress*, a lightweight interaction technique that enables pressure input interactions on mobile devices by using only the built-in vibration motor and accelerometer. We showed evidence that, with continuous visual feedback [11], users can reliably and quickly input using two to six pressure levels and different hand postures with accuracy ranging from 99.7% to 67% and from ~1 to ~4 seconds interaction time, depending on the hand gripping used. We also described how different users favorably judged the usability and potential applications of the system. Our work presents some limitations in terms of compatibility across devices (different motors or device cases propagate vibrations differently) and battery consumption, due to the continuous activation of the vibration motor. However, these issues could be alleviated by using frequent but short vibration patterns at lower intensities. Future work will aim to identify hand postures and specific pressure for selective portions of the screen. Finally, we also plan to measure pressure inputs using tangible objects (e.g., styli, tokens) rather than fingers.

REFERENCES

1. Boring, S., Ledo, D., Chen, X., Marquardt, N., Tang, A., Greenberg, S.,The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction, *In Proc. of MobileHCI*, (2012), 39-48.
2. Brewster, S.A., Hughes, M., Pressure-based text entry for mobile devices, *In Proc. of MobileHCI*, (2009).
3. Cechanowicz, J., Irani, P., Subramanian, S., Augmenting the mouse with pressure sensitive input*, In Proc. of CHI*, (2007), 1385-1394.
4. Goel, M., Wobbrock, J.O., Patel, S.N., GripSense: Using Built-In Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones, *In Proc. of UIST*, (2012), 545-554.
5. Hart, S.G., Staveland, L.E., Development of a multi-dimensional workload rating scale, *Human Mental Workload. Elsevier*, (1988), 139–183.
6. Heo, S., Lee, G., Forcetap: extending the input vocabulary of mobile touch screens by adding tap gestures, *In Proc. of MobileHCI*, (2011), 113-122.
7. Hwang,S., Bianchi,A., Wohn, K., MicPen: pressure-sensitive pen interaction using microphone with standard touchscreen, *In Proc. of CHI EA*, (2012), 1847-1852.
8. Hwang, S., Wohn, K., PseudoButton: enabling pressure-sensitive interaction by repurposing microphone on mobile device, *In Proc. of CHI EA*, (2012), 1565-1570.
9. Iwasaki,K., Miyaki,T., Rekimoto, J., Expressive typing: a new way to sense typing pressure and its applications, *In Proc. of CHI EA*, (2009), 4369-4374.
10. Miyaki, T., Rekimoto, J., GraspZoom: zooming and scrolling control model for single-handed mobile interaction, *In Proc. of MobileHCI*, (2009), 81-84
11. Ramos, G., Boulos, M., Balakrishnan, R., Pressure widgets, *In Proc. of CHI*, (2004), 487-494.
12. Strachan, S., Murray-Smith, R., Muscle Tremor as an Input Mechanism, *In Proc. of UIST*, (2004).