

Regular Paper

BaroTouch: A Technique for Touch Force Sensing Using a Waterproof Device's Built-in Barometer

RYOSUKE TAKADA^{1,a)} TOSHIYUKI ANDO¹ BUNTAROU SHIZUKI¹ SHIN TAKAHASHI¹

Received: April 20, 2018, Accepted: November 7, 2018

Abstract: We present BaroTouch, a technique that leverages a waterproof mobile device's built-in barometer to measure the touch force. When an airtight waterproof device is touched, the distorted surface changes the air pressure inside that device and thus changes the built-in barometer value. BaroTouch estimates the touch force with a simple conversion from the air pressure to the touch force; it does not need machine learning. In addition, BaroTouch is a passive sensing technique. Therefore, BaroTouch could be lightweight. To investigate BaroTouch, we conducted two experiments. First, we investigated the relationship between the sensor value and the touch positions or forces using weights with three devices: two smartphones and a smartwatch. Second, in a controlled user study with 15 participants, we examined how the users can use BaroTouch. The results showed that the participants could exert 2–6 levels of the touch force with accuracies of over 96% accuracy under each device using BaroTouch.

Keywords: force input, pressure sensing, touch screen, smartphone, smartwatch

1. Introduction

A touchscreen, which detects a finger's touch along with its position, serves as a major input channel on a current mobile device, it still has a room for improvement in touch interactions considering how human touches objects. Specifically, a human touches, presses or taps objects in the real world with a different force. Due to this fact, if a touchscreen can measure this touch force, it is shown to enrich the input vocabulary for daily operations [28].

Currently, some commercial mobile devices such as iPhone 6S [2] measure the touch force using a built-in force sensor. Previous work (e.g., Refs. [26], [27]) extended input interfaces using additional sensors to estimate the touch force, which requires cost and space for sensors. To solve this problem, some other previous work (e.g., Refs. [14], [32]) attempted to estimate touch force with machine learning and/or FFT and built-in sensors of mobile devices, which requires an expensive calculation cost.

For this reason, we developed BaroTouch. BaroTouch estimates the user-exerted touch force by utilizing the changes in the internal air pressure of the device while the user touches as shown in **Fig. 1**. When an airtight waterproof device is touched, the distorted surface changes the air pressure inside that device and thus changes the built-in barometer value. BaroTouch estimates the touch force with a simple conversion from the air pressure to the touch force; it does not need machine learning. In addition, BaroTouch is a passive sensing technique. Therefore, BaroTouch could be lightweight.

We showed the idea to use a built-in barometer in a mobile device to estimate the touch force [31]. In contrast to the prior work, we designed new algorithms and implemented BaroTouch

on a smartwatch. Then, to investigate BaroTouch's characteristics, we conducted two experiments. First, we investigated the relationship between the sensor value and the touch positions or forces using weights with three devices: two smartphones (Xperia Z5 Compact and Xperia Z5) and a smartwatch (LG G Watch R). The results showed that the device's center part was more sensitive than the edge, and BaroTouch's estimated the touch force increases in proportion to the weight in a linear manner. Second, in a controlled user study with 15 participants, we examined how the users can use BaroTouch. The results showed that the participants could exert 2–6 levels of touch force with accuracies higher than 96.4% under each device using BaroTouch.

This paper makes the following four main contributions:

- Design of touch force sensing using a waterproof device's built-in barometer,
- Implementation of two BaroTouch's algorithms: one for smartphones and one for smartwatches,
- Investigation of the relationship between the sensor value and the touch positions or forces,
- Evaluation of how many levels of touch force a user can exert with BaroTouch.

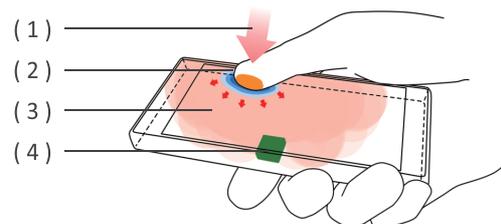


Fig. 1 BaroTouch estimates touch force from barometer sensor value. 1) When a user touches the touch screen, 2) the screen becomes distorted, and thus 3) internal air is compressed and 4) built-in barometer sensor value increases.

¹ University of Tsukuba, Tsukuba, Ibaraki 305–8577, Japan

^{a)} rtakada@iplab.cs.tsukuba.ac.jp

2. Related Work

Some commercially available smartphones (e.g., Refs. [2], [13]), smartwatch [3], and touchpad [4] measure the touch force with built-in force sensors to provide interactions involving the touch force. In addition, there are many research projects on the touch force sensing techniques which can be used without force sensors, as well as the ones that explored interaction designs involving the touch force.

2.1 Touch Force Sensing on Mobile Devices with Additional Equipment

Some techniques of the force sensing by an additional equipment have been suggested to expand touch interactions on a mobile device. For example, Nakai et al. [24] connected a smartphone with a smartphone case by gel and detected the touch force using the deformation of the gel induced when the touchscreen is touched. Acoustruments [19] can measure the touch force when a pipe, which connects a smartphone's microphone to its speaker, is touched. Force Gestures [11] attached force sensors to the case of a mobile device to measure the touch force on its touchscreen. Ono et al. [26] analyzed the resonant properties of a smartphone to recognize the touch force using a smartphone case with a vibration speaker and a piezoelectric microphone. In contrast to these approaches, BaroTouch realizes touch force sensing by only a built-in barometer.

2.2 Touch Force Sensing on Smartphones Using Built-in Sensors

Techniques that only use a smartphone's built-in sensors to measure the touch force on its touchscreen have been proposed. For example, GripSense [9] measures the touch force by vibrating the smartphone and observing the diminished gyroscope readings. ForceTap [12] is a technique for sensing the tapping force based on the acceleration value from the built-in accelerometer. ForcePhone [32] measures the touch force by using a smartphone's built-in microphone and speakers with ultrasonic sensing and machine learning. VibPress [14] measures the touch force by using a smartphone's built-in accelerometer and vibration motor. PseudoButton [16] emulates a force sensitive-button by using a built-in speaker and microphone. It emits sounds and analyzes the feedback sound when the user's finger covers on the pinhole of microphone and emulates the force sensitive-button. Low et al. [20] proposed the touch force estimating method which uses the smartphone's front camera and the flash to detect the touch force by measuring the distance between the smartphone and the user's palm.

In contrast to the above work, no prior work has investigated the use of a smartphone's built-in barometer for sensing the touch force. Moreover, BaroTouch does not need machine learning; it measures the touch force with a simple conversion from the air pressure to the touch force. In addition, BaroTouch is a passive sensing technique and thus could be lightweight, in contrast to active sensing techniques such as those in Refs. [14], [32].

2.3 Other Touch Force Sensing Techniques

Expressive Touch [27] is a method for measuring the tapping force on a tabletop touchscreen by using the peak amplitude of the sound waves generated by finger taps. Dietz et al. [8] measured the key pressing force by using a modified flexible membrane keyboard. Pressing the Flesh [22] measures the touch force on any surface by observing the color changes in the fingertips and nails using a camera. MicPen [15] is a force sensitive stylus by using a microphone. It analyzes the generated acoustic signal when the tip of the stylus is rubbed on the touchscreen and estimates the force. Expressive Typing [18] measures the typing pressure of keyboards using equipped accelerometers. PreSenseII [29] enables the bi-directional pressure based interactions using a touchpad with tactile feedback. Ramos et al. [28] proposed a UI framework of stylus input on tablets. They conducted target selection tasks by varying the stylus' pressure and proposed the framework based on these results.

In contrast to these studies, our method focuses on measuring the touch force on a mobile device by using its built-in barometer.

2.4 Design of Touch Force Usage

Various designs of the touch force usage are proposed. For example, Push-Push [10] proposes a drag-like operation which does not conflict with the page transition operation by using hover and pressed states in addition to the conventional touch. Pressures [30] examined pressure characteristics of multi-touch gestures and designed extended multi-touch gestures based on the examination. Arif et al. [5] proposed a touch force sensing method which is based on touch positions and lengths. Using the proposed method, they presented a software keyboard which enables users to bypass incorrect predictions by exerting an extra force. Brewster et al. [6] designed a force based software keyboard which can switch upper and lower case based on the force. ForceEdge [1] is a scroll technique based on the touch force by using smartphones and trackpads which have force sensors. GraspZoom [23] attached a force sensitive resistor on a backside of a smartphone and proposed a combination of a force gesture and a tiny thumb gesture of one hand. One-Press Control [7] proposed a tactile input method in a force measurable keyboard by classifying the touch force patterns of already pressed down keys. Mandalapu et al. [21] explored a force touch gesture as an alternative to a multi-touch gesture. In the task of zooming, the force touch gesture was faster than the two-finger pinch gesture.

As shown in the above studies, there are various useful gestures using the touch force. Therefore, it is important to enable smartphones/smartwatches which have no touch force sensors to provide these gestures.

2.5 Interactions Using a Barometer

Barometers are mainly used for context awareness and movement recognition in various works. Emoballoon [25] is a balloon that can recognize touch gestures on the balloon, which has a barometer and a microphone. Wu et al. [33] recognized doors' open or close using a built-in barometer in a smartphone. Ye et al. [34] suggested a floor localization system that can estimate the floor level of a multi-floor building using a barometer by utilizing

changes of atmospheric pressure depending on height. Hyuga et al. [17] proposed a localization method that can estimate the user's state in a subway only using a smartphone's built-in barometer.

BaroTouch uses the smartphone/smartwatch's built-in barometer to estimate the touch force.

3. BaroTouch

BaroTouch is a technique that leverages a waterproof device's built-in barometer to estimate the touch force a user exerts on the touchscreen. When the touchscreen is pressed, its distorted screen compresses the internal air. If the touch force is strong, the screen distorts largely and thus compresses the internal air heavily. Our idea is to continuously measure its pressure using the built-in barometer to estimate the touch force. Note that the airtightness of a waterproof device maintains the change; otherwise, the change instantly recovers. This allows us to measure the change caused by the distortion of the screen by the barometer of the commodity device. Since this change depends on the form factor, we built two algorithms for estimation: one for smartphones and one for smartwatches.

3.1 Change in the Internal Air Pressure Measured with the Barometer

We observed the change in the internal air pressure with the built-in barometers of different form factors. In this observation, we prepared the following three waterproof devices equipped with the same barometer (Alps Electric HSPPAD038) to control the experimental conditions: a small smartphone (S-SP, Fig. 2 a), a large smartphone (L-SP, Fig. 2 b), and a smartwatch (SW, Fig. 2 c). S-SP was SONY Xperia Z5 Compact (Waterproof rating: IPX5/8). L-SP was SONY Xperia Z5 (Waterproof rating: IPX5/8). SW was LG G Watch R (Waterproof rating: IPX7). We attached a force sensor (Interlink Electronics FSR402) to each touchscreen to estimate the finger's touch force as ground truth.

As a setup, we modified SW as shown in Fig. 3. SW has a small vent, which is a hole open on the smartwatch's surface to allow the barometer inside the smartwatch to measure the atmospheric pressure. We sealed the vent with adhesive tape to enclose the internal air so that the internal air pressure is kept increased while the device is pressed.

In this observation, we placed the devices on a desk and one author touched each touchscreen with the index finger two times:

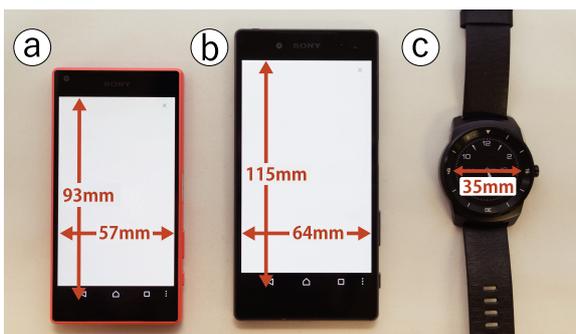


Fig. 2 Devices. Left: S-SP, center: L-SP, right: SW.

weakly once, then strongly once. Figure 4 shows the waveforms of the barometer and the force sensor values under S-SP. Under this condition, the barometer value greatly increased at first, then decreased, and finally recovered when the touchscreen was released. These shapes of waveforms were formed because the device's internal air flowed out/in when the screen distorted/recovered, as illustrated in Fig. 5. Specifically, the internal air pressure is usually the same as the atmospheric pressure (Fig. 5 a). Once the touchscreen is pressed, the distorted screen compresses the internal air and thus increases its pressure (Fig. 5 b). Immediately after this, the internal air begins to flow out from gaps of the device even while the finger continues to touch (Fig. 5 c). As a result, its pressure gradually drops to the same as the atmospheric pressure (Fig. 5 d). When the finger leaves the touchscreen, the distortion recovers. This decreases the internal air pressure below the atmospheric pressure (Fig. 5 e), which gradually returns to the same as the atmospheric pressure (Fig. 5 f).

Figure 6 shows the waveforms of the barometer value and the force sensor values under the SW condition. These waveforms show that the device's internal air did not flow out/in by adhe-

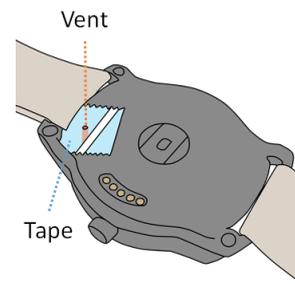


Fig. 3 Setup of the smartwatch: the vent was sealed with adhesive tape.

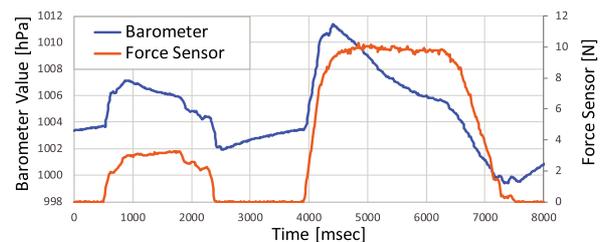


Fig. 4 Waveforms of the internal air pressure and touch force on the S-SP when it was touched.

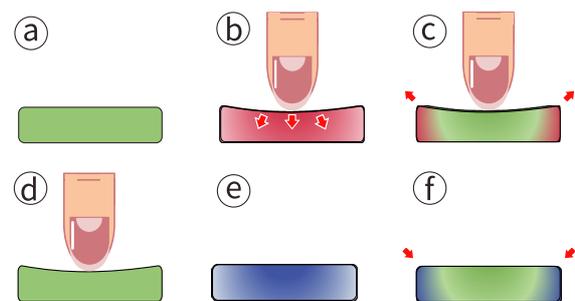


Fig. 5 Change of the internal air pressure of a waterproof device. a) Before touch. b) Immediately after touch. c) During touch. d) A period of time after touch. e) Immediately after the finger leaves. f) Long after the finger leaves.

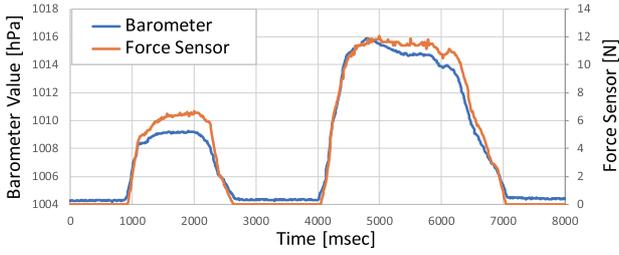


Fig. 6 Waveforms of the internal air pressure and touch force on the SW when it was touched.

sive tape when the screen distorted/recovered. In this case, the device's internal air changes in the order of Fig. 5 a \rightarrow Fig. 5 b \rightarrow Fig. 5 a.

3.2 Algorithms

Since the change in the barometer value differs under smartphones or a smartwatch (with adhesive-tape), as mentioned above, we built two algorithms to estimate the touch force during the touch: one for smartphones and one for smartwatches.

3.2.1 Algorithm for Smartphones

The algorithm for smartphones estimates the touch force f by low-pass filtering the raw barometer value b_{raw} using the following equations:

$$f[n] = b_{lp}[n] - b_{base} \quad (1)$$

$$b_{lp}[n] = b_{lp}[n-1] \times \alpha + b_{raw}[n] \times (1 - \alpha). \quad (2)$$

In these equations, b_{lp} is the low-passed value of the raw barometer value; b_{base} is the highest raw barometer value during the last 20 seconds while the touchscreen was not touched; n is a number of the barometer value; α is a constant equal to 0.99; b_{lp} is reset to b_{base} always when the user does not touch the touchscreen. Our current implementation runs this algorithm at the frequency of 90 Hz.

In this algorithm, we use a strong low-pass filter to estimate the touch force from the raw barometer values (i.e., the internal air pressure). This allows the algorithm to identify that the touch force is exerted while the internal air pressure gradually drops during the touch, and that the touch force is diminished when the finger leaves and thus the internal air pressure steeply drops.

Figure 7 and **Fig. 8** show the results of the algorithm under S-SP and L-SP conditions. Figure 7 (a) and Fig. 8 (a) show the touch force measured by the force sensors (i.e., the ground truth); Fig. 7 (b) and Fig. 8 (b) show the raw barometer value; Fig. 7 (d) and Fig. 8 (d) show the estimated force. As shown in Fig. 7 and Fig. 8, the estimated force is more similar to the ground truth than the raw barometer value. The correlation coefficient between the ground truth and the estimated force increased than the correlation coefficient between the ground truth and the raw barometer value (Fig. 7: 0.690 \rightarrow 0.814, Fig. 8: 0.829 \rightarrow 0.895).

3.2.2 Algorithm for Smartwatches

The algorithm for smartwatches simply uses the raw barometer value offset by b_{base} , which means the atmospheric pressure when the touchscreen is touched, as the touch force. Specifically, the algorithm is expressed by the following equation:

$$f[n] = b_{raw}[n] - b_{base} \quad (3)$$

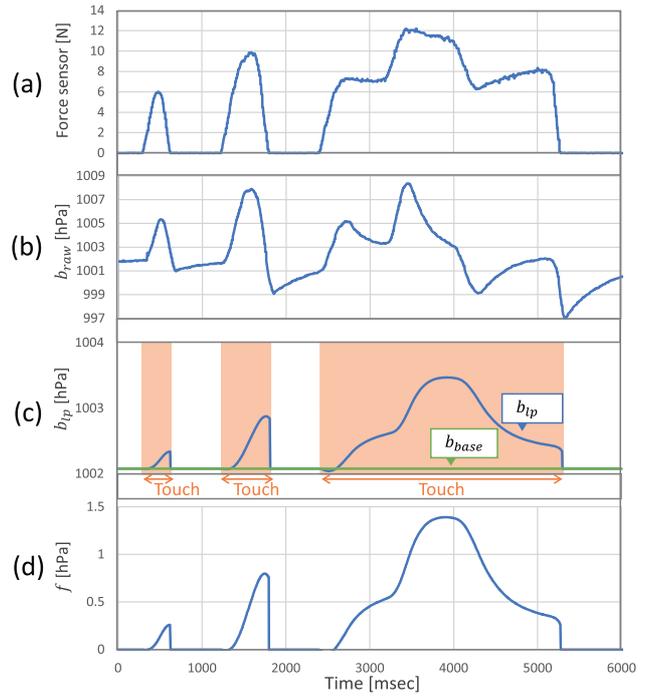


Fig. 7 Estimated touch force when S-SP is touched. Colored areas represent the period when the touchscreen was touched.

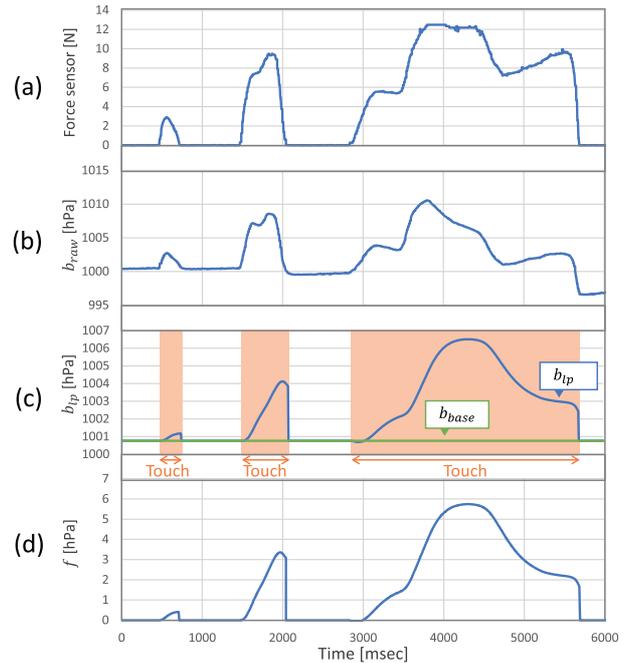


Fig. 8 Estimated touch force when L-SP is touched. Colored areas represent the period when the touchscreen was touched.

Our current implementation runs this algorithm at the frequency of 90 Hz. **Figure 9** shows the results of the algorithm under the SW condition. The correlation coefficient between the ground truth and the estimated force increased than the correlation coefficient between the ground truth and the raw barometer value (Fig. 9: 0.972 \rightarrow 0.991).

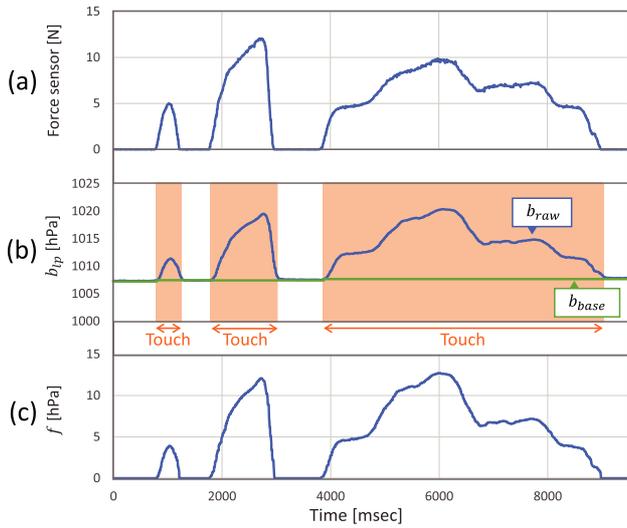


Fig. 9 Estimated touch force when SW is touched. Colored areas represent the period when the touchscreen was touched.

4. Force-Sensitivity of Different Touch Positions and Force

To clarify our algorithms' force-sensitivity, we investigated the relationship between touch positions and the estimated touch force, and between the real touch force and the estimated touch force. In these investigations, we used the same smartphones and smartwatch as mentioned above.

4.1 Procedure

These investigations were carried out in a room where the windows and doors were all closed. The atmospheric pressure of the room was 1,013.2 hPa at the start of the experiment.

In the force-sensitivity investigation of touch positions, we placed a weight on various positions of a touchscreen and recorded the estimated touch force (i.e., f). We divided the touchscreen into 5 rows \times 3 columns of areas (S-SP: 1.86 \times 1.90 cm in each area) and 6 rows \times 4 columns of areas (L-SP: 1.91 \times 1.60 cm in each area). Then, we placed a 100 g weight (radius = 1 cm) on a pedestal (radius = 1 cm, weight = 4.33 g, printed by 3D printer, ABS) as shown in Fig. 10 at the center of each area and recorded the largest f . We used only the two smartphones because the smartwatch's display is too small to be divided. We placed the weight 10 times on each position. As a result, the total number of trials was 420 ((S-LP: 18 areas + L-SP: 24 areas) \times 10 times) in this investigation.

In the force-sensitivity investigation of touch forces, we placed 20 g, 50 g, 100 g, 200 g, 500 g, and 1,000 g weights on a pedestal to unify the contact area sizes for all weights. We then placed the pedestal at the center and upper-left corner of the touchscreens of the two smartphones and recorded the largest f ; we placed the pedestal at the center of the touchscreen of the smartwatch and recorded the largest f . We placed the weights 300 times ((2 smartphones \times 2 locations + 1 smartwatch) \times 6 different weights \times 10 times) for the force-sensitivity investigation of touch force.

4.2 Results and Discussion

Figure 11(a) and Fig. 11(b) show the results of the force-

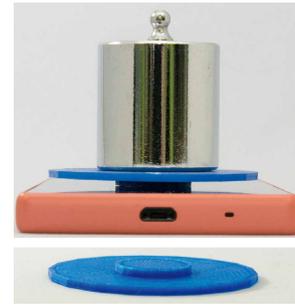


Fig. 10 Pedestal that we used in the investigation of our algorithm's force sensitivity.

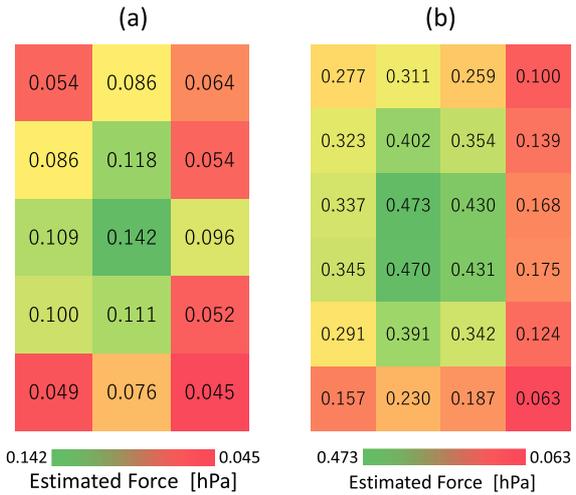


Fig. 11 Force-sensitivity characteristic under different position. a) S-SP. b) L-SP.

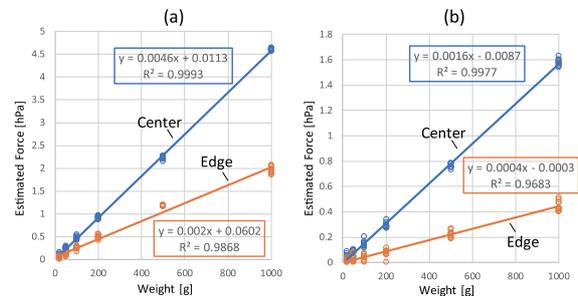


Fig. 12 Force-sensitivity of different force. a) S-SP. b) L-SP.

sensitivity investigation of touch positions with two smartphones. These figures show that the estimated force of the device's center part was larger than others for each smartphone. This would be because the center part of the device had greater distortion than the edge parts. In Fig. 11 (a), estimated force of the left side is higher than the right side. This would be due to the difference of the shape of internal space and the position of the sensor. By using these characteristics, it may be possible to estimate forces that do not depend on the touch positions.

Figure 12 (a) and Fig. 12 (b) show the results of the force-sensitivity investigation of touch force with two smartphones. They show that the value increases in proportion to the weight. We performed a linear regression between the weight and estimated force in each condition and obtained a high coefficient (Fig. 12 (a): $R^2 = 0.9993$ on the center, $R^2 = 0.9868$ on the upper-left corner; Fig. 12 (b): $R^2 = 0.9977$ on the center, $R^2 = 0.9683$

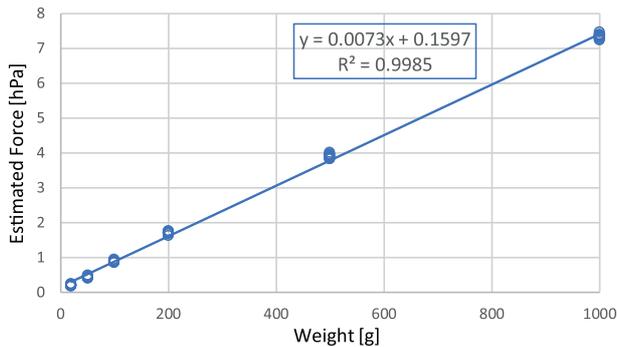


Fig. 13 Force-sensitivity of different force on SW. The weights are putted on the center of screen.

on the upper-left corner). Furthermore, since the same characteristic was observed at both the center and upper-left corner of the touchscreen, the entire touchscreen would share this characteristic.

Figure 13 shows the results of the force-sensitivity investigation of touch force on the smartwatch. We performed a linear regression between the weight and estimated force in each condition and obtained a high coefficient (Fig. 13: $R^2 = 0.9985$).

5. User Study of Different Levels of Touch Force

We conducted a controlled user study to evaluate how many levels of touch force a user can exert with BaroTouch. In this study, we used two applications shown in **Fig. 14**. We examined the accuracies of 2–6 force levels.

5.1 Participants

We recruited 15 participants (11 male and 4 female, 12 right-handed). Their ages ranged from 19–30 ($M=22.93$, $SD=2.35$). All the participants used their smartphones in their daily lives (usage time=41–102 months, $M=66.87$ months). No participants have used smartphones and smartwatches with force-sensitivity. After the study, we paid JPY 1,640 (USD 14.81) to each participant for their time.

5.2 Experimental Environment

The experiment was carried out in a room where the windows and doors were all closed. The atmospheric pressure of the room ranged from 995.50–1007.09 hPa ($M=1000.88$ hPa, $SD=5.31$ hPa) which is measured using the S-SP smartphone before the experiment. We used the same smartphones and smartwatch as during the first experiment.

5.3 Procedure

Each participant was asked to remain sitting on a chair during the experiment. Firstly, the experimenter informed the participant of the purpose of the study. After this, the participant gave informed consent and was instructed to answer a demographic questionnaire. Then, we measured the pinch (by a thumb and index finger) and push (by index finger) strength of the participant's right thumb using a pinch gauge (Baseline, Baseline 12-0228). The average finger pinch and push strengths were 7.60 kg ($SD=3.99$ kg) and 7.61 kg ($SD=1.66$ kg), respectively.

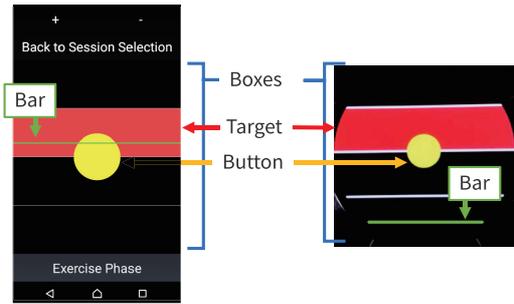


Fig. 14 Applications used to investigate how many levels of touch force a user can exert (left: for the smartphones, right: for the smartwatch).

The experimenter presented the application shown in **Fig. 14** to the participant and explained its usage. Then, the participant was asked to touch the button displayed at the center of the touchscreen 5 times weakly and 5 times strongly. The application recorded the largest f of each touch. Calibration was achieved by setting the average of the largest f s of strong touches as the upper limit, and those of the weak touches as the lower limit. The application divides the screen into 2–6 boxes (divide conditions) with an identical height; the top of the highest box corresponds to the upper limit; the bottom of the lowest box corresponds to the lower limit.

We divided the participants into three equally sized groups (Groups A, B, and C). Group A used the three devices in the order of S-SP → L-SP → SW; Group B used L-SP → SW → S-SP; Group C used SW → S-SP → L-SP. The five participants in a group were assigned all 5 divide conditions (i.e., 2–6 boxes) in one of the following orders: 2 → 3 → 4 → 5 → 6, 3 → 4 → 5 → 6 → 2, 4 → 5 → 6 → 2 → 3, 5 → 6 → 2 → 3 → 4, and 6 → 2 → 3 → 4 → 5. Under each divide condition, the participant performed a practice session and a test session. In the practice session, the participant completed 12 trials. In the test session, the participant completed 60 trials.

In each trial, the target was a red box displayed in a random order. The participant selected the target by adjusting the touch force. Specifically, the participant touched and then held a finger on the device screen. While holding, a bar moved up and down proportionally to the touch force exerted. The participant selected the target by keeping the bar within the target for one second. If a trial failed (i.e., finger released before the target was selected), the same target was displayed; the participant tried to select the target until succeeding the selection. The participant was instructed to perform the task as quickly and accurately as possible. The participant could take a break between trials; a long break (>3 minutes) was enforced between device conditions. Consequently, one participant completed at least 1080 trials (3 devices × 5 divides × (12 + 60 trials)) during the experiment.

In each trial, we recorded three measures. One was the correct rate of target selection. Next one was the target selection time of a successful trial, which is the time required to correctly input touch force including one second to trigger the selection. The final one was the number of crossings (NC), which is defined as the number of times the bar enters or leaves the target for a particular trial, minus 1 (e.g., $NC=2$ for a task where the participant overshoots and reacquires the target).

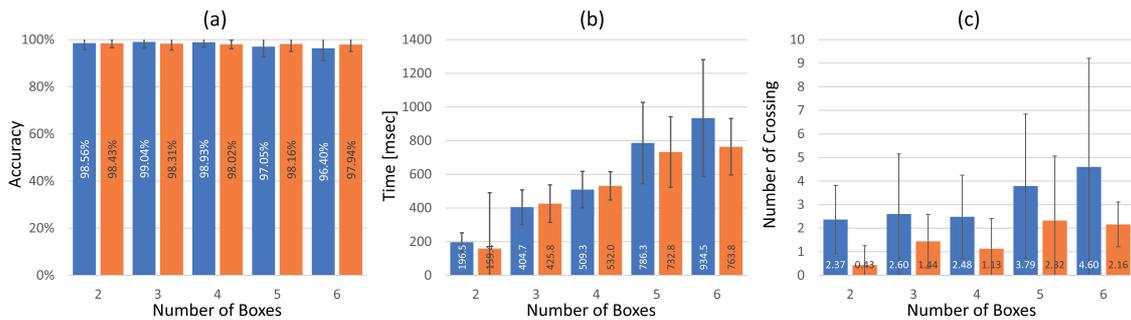


Fig. 15 Result of user study under S-SP (blue) and L-SP (orange). a) Correct rate of target selecting, b) Target selection time, c) Number of crossing.

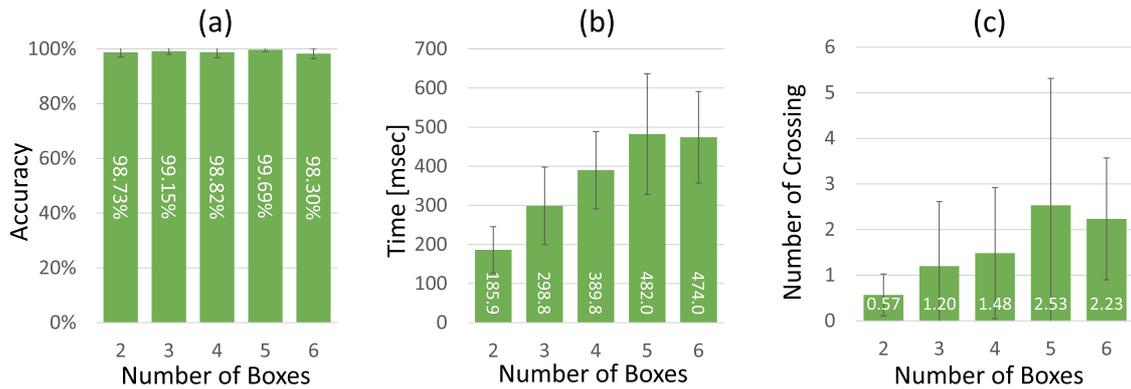


Fig. 16 Result of user study under SW. a) Correct rate of target selecting, b) Target selection time, c) Number of crossing.

After completing all the sessions, we asked the participant to fill out a questionnaire. The entire experiment, from the prior explanation to the last questionnaire, took approximately 95 minutes for each participant.

5.4 Results and Discussion

Figure 15 (a) and Fig. 16 (a) show the success rate for each device condition and divide condition. Figure 15 (b) and Fig. 16 (b) show the target selection time for each device condition and divide condition. Figure 15 (c) and Fig. 16 (c) show the NC for each device condition and divide condition.

A two-way ANOVA (two devices: S-SP and L-SP, and divide conditions) with a repeated measures on both factors revealed significant differences for the target selection time across the divide condition ($F(4,56)=51.437, p<0.01$), and NC across divide conditions ($F(4,48)=6.446, p<0.01$), but not for the different device condition, or for their interactions. A one-way ANOVA (divide conditions) with a repeated measure on both factors revealed significant differences for the target selection time ($F(4,56)=51.117, p<0.01$) and for NC ($F(4,56)=51.117, p<0.01$). Our findings for completion target selection time and NC significantly vary depending on the divide condition.

In the result of accuracy, there are no significant differences, but each result is highly accurate and it suggests that a BaroTouch user can input 2–6 force levels with over 96 % accuracy. Furthermore, under the 3 boxes divide condition, which is used in the peek and pop function with 3D Touch on an Apple iPhone6S, the S-SP, L-SP and SW conditions have a success rate of 99.04%, 98.31%, and 98.30%, respectively.

In the SW condition, the target selection time and NC of the

5 boxes divide condition are little higher than in the 6 boxes divide condition. We consider that the target width of each divide condition is almost the same size (5 boxes: 7.00 mm, 6 boxes 5.83 mm), and the participants input carefully in the 6 boxes divide condition.

In the questionnaire, we asked the participant that “Is it easy to distinguish touch force levels using BaroTouch?”, “Is it easy to distinguish touch force levels with S-SP in each divide condition (2→6)?”, “Is it easy to distinguish touch force levels with L-SP in each divide condition (2→6)?”, and “Is it easy to distinguish touch force levels with SW in each divide condition (2→6)?” with five-point Likert scales which translates into numbers between 1 (very negative) and 5 (very positive). In addition, each participant wrote comments in a free-form questionnaire. The result of the average score of the first questionnaire is 4.00 (SD = 0.76). The result of average scores of all divide conditions under S-SP, L-SP, and SW conditions were 4.05, 4.32, and 4.08, respectively. On the comments, 7 participants also commented that “I feel BaroTouch is easy to operate”.

6. Applications

6.1 Music Player

We propose a music player. Figure 17 shows an interaction flow of the music player. Tapping softly toggles between play/pause modes; tapping hardly shows a music list.

6.2 Recipe Application

We propose a recipe application. While the user cooks with a recipe application on the user’s smartphone, sometimes the user wants to measure the weight of seasoning. At that time, the user



Fig. 17 Example application: music player. a) Play mode, b) pause mode, c) music list.

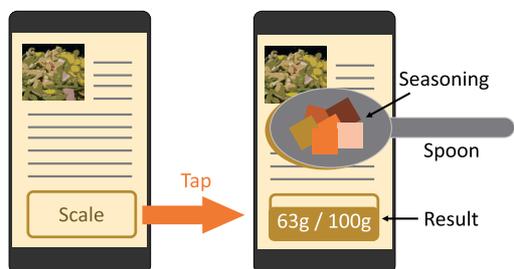


Fig. 18 Example application: recipe application.

can measure it on a spoon which is placed on the smartphone’s touchscreen as shown in Fig. 18. The application can measure a weight of seasoning using Fig. 12 and Fig. 13.

7. Discussion and Future Work

7.1 Finger Angle Estimation

Some smartphones (e.g., Xperia Z5) estimate the touch force based on the touch area size. Interestingly, this touch area size can also be changed by different angles of the finger. The touch area size and barometer value when the touchscreen is touched with a different force are shown in Fig. 19. Figure 19 also shows that the touch area size changes based on the touch force and the angle of the finger. On the other hand, the barometer value changes based only on the touch force. Therefore, the angle of the finger could be estimated by removing the touch force factor from the touch area size.

7.2 Influence by Environments

Although the results of the experiments show BaroTouch can estimate the touch force, the experiments were conducted only in the controlled environment. Therefore, our future work is to verify whether BaroTouch can be used in-the-wild. Some environmental conditions would have an influence on BaroTouch. For example, taking an elevator and boarding an airplane create a gradual change in the atmospheric pressure and thus in the internal air pressure of the device. Currently, this influence cannot be eliminated with our algorithms even if our algorithm for smartphones uses a low-pass filter; for example, our algorithms estimate that the touch force becomes stronger/weaker when the elevator starts to go up/down while the user touches the touch screen. To reduce such influences, we plan to use accelerometers to detect changes in environmental conditions to cancel their influences on the estimated touch force.

7.3 Airtightness

BaroTouch requires the airtightness of the smartphone to a certain degree. In the experiments, we used smartphones whose waterproof ratings are IPX5/8 and a smartwatch whose waterproof

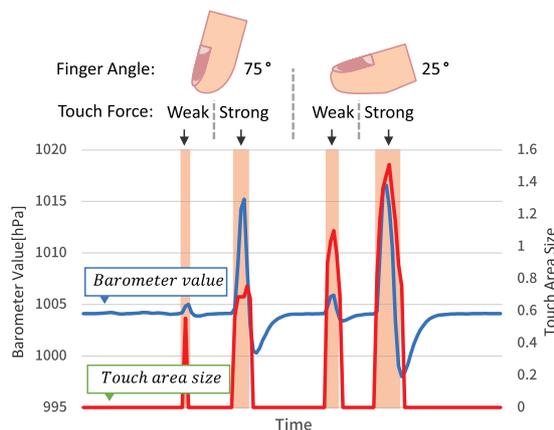


Fig. 19 The air pressure and touch size when the device is being touched.

rating is IPX7. To explore the applicability of our approach, we will verify BaroTouch’s characteristic under other waterproof ratings. Especially, we have to find the minimum airtightness where BaroTouch can work.

7.4 Effect of Temperature

Changes in the temperature make the internal air pressure change and have influences on the touch force sensing of BaroTouch. According to Boyle-Charles’ law, as the temperature rises, the internal air pressure rises too. This phenomenon was observed with a smartwatch. It would be possible that the sealed condition, tiny volume of internal air, and the generation of heat are the reason. In our algorithm, this problem is not serious because the barometer value when the touch screen is touched serves as a base. However, to solve this problem completely, we will improve our algorithm to use Boyle-Charles’ law for coping with changes in the temperature.

7.5 Latency

We measured a latency between the time that a user touches the touch screen of each device, and the time that the evaluation application’s bar starts to move using a 960 fps camera (SONY DSC-RX100M5). The results of S-SP, L-SP, and SW were 0.096 sec, 0.076 sec, and 0.101 sec, respectively.

Moreover, our method has a latency due to the low-pass filter as shown in Fig. 7, Fig. 8, and Fig. 9. As shown in Fig. 15 and Fig. 16, the users succeeded to exert a touch force in about 0.2–0.6 sec. This duration includes a latency of the low-pass filter. While BaroTouch can estimate the touch force in a short time considering no additional sensors, developers should take this latency into account in designing their applications with our method.

7.6 Continuous Press

Since our method uses a low-pass filter, if the user continues to press down for a long time with a certain pressure, the estimated force converges to almost zero. Therefore, our method can be used in an application that uses touch with a touch force for a short time, such as the music application as shown in Section 6.1. Moreover, developers should consider this limitation in developing their applications with our method.

7.7 Modeling of The Internal Air Pressure while Pressing

In this paper, we use a simple low-pass filter for estimating the touch force, because it can be used in different airtightness conditions and is easy to implement. However, it is difficult to eliminate the latency. If we can construct a model of the internal air pressure change while a touchscreen is pressed, we could calculate the touch pressure with less latency and higher accuracy than the low-pass filter. However, it requires two relationships. One is the relationship between the touch pressure and the volume change due to the distortion of the device; the other is the relationship between the pressure difference of the device's inside/outside and the air mass flow. These relationships differ for each device. In the future, we will construct the model of each device and compare it with the BaroTouch's algorithm.

7.8 Touch Force Sensing on Back Side

Since BaroTouch estimates the touch force based on the distortion of the devices, the estimation is difficult on a device with a hard frame. On the contrary, the force touch on the back side of the device makes internal air pressure changes if the surface of the back side is soft enough. Using this, it would be possible to estimate the touch force on the back side with BaroTouch.

8. Conclusions

We showed BaroTouch, a technique for estimating the touch force from the change in the built-in barometer value due to a user's finger touching a waterproof device's touchscreen. We implemented two algorithms of BaroTouch: one for smartphones and one for smartwatches. Our evaluation has shown BaroTouch is easy to use with higher than 96.4% accuracy on 2–6 levels of touch force. Furthermore, under the 3 boxes divide condition, which is used in the peek and pop function with 3DTouch on an Apple iPhone6S, the results of our user study showed accuracies of 99.04% (S-SP), 98.31% (L-SP), and 98.03% (SW), respectively.

References

- [1] Antoine, A., Malacria, S. and Casiez, G.: ForceEdge: Controlling Autoscroll on Both Desktop and Mobile Computers Using the Force, *Proc. 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, pp.3281–3292, ACM (online), DOI: 10.1145/3025453.3025605 (2017).
- [2] Apple Inc.: Apple iPhone6S (2015), available from (<https://support.apple.com/kb/SP726>) (accessed 2018-11-12).
- [3] Apple Inc.: Apple Watch (2015), available from (<https://www.apple.com/watch/>) (accessed 2018-11-12).
- [4] Apple Inc.: Magic Trackpad 2 (2015), available from (<https://www.apple.com/shop/product/MJ2R2LL/A/magic-trackpad-2>) (accessed 2018-11-12).
- [5] Arif, A.S. and Stuerzlinger, W.: Pseudo-Pressure Detection and Its Use in Predictive Text Entry on Touchscreens, *Proc. 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, OzCHI '13*, pp.383–392, ACM (online), DOI: 10.1145/2541016.2541024 (2013).
- [6] Brewster, S.A. and Hughes, M.: Pressure-based Text Entry for Mobile Devices, *Proc. 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '09*, pp.9:1–9:4, ACM (online), DOI: 10.1145/1613858.1613870 (2009).
- [7] de Jong, S., Kirkali, D., Schraffenberger, H., Jilissen, J., de Rooij, A. and Terpstra, A.: One-Press Control: A Tactile Input Method for Pressure-sensitive Computer Keyboards, *CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10*, pp.4261–4266, ACM (online), DOI: 10.1145/1753846.1754136 (2010).
- [8] Dietz, P.H., Eidelson, B., Westhues, J. and Bathiche, S.: A Practical Pressure Sensitive Computer Keyboard, *Proc. 22nd Annual ACM Symposium on User Interface Software and Technology, UIST '09*, pp.55–58, ACM (online), DOI: 10.1145/1622176.1622187 (2009).
- [9] Goel, M., Wobbrock, J. and Patel, S.: GripSense: Using Built-in Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones, *Proc. 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12*, pp.545–554, ACM (online), DOI: 10.1145/2380116.2380184 (2012).
- [10] Han, J. and Lee, G.: Push-Push: A Drag-like Operation Overlapped with a Page Transition Operation on Touch Interfaces, *Proc. 28th Annual ACM Symposium on User Interface Software and Technology, UIST '15*, pp.313–322, ACM (online), DOI: 10.1145/2807442.2807457 (2015).
- [11] Heo, S. and Lee, G.: Force Gestures: Augmented Touch Screen Gestures using Normal and Tangential Force, *CHI '11 Extended Abstracts on Human Factors in Computing Systems, CHI EA '11*, pp.1909–1914, ACM (online), DOI: 10.1145/1979742.1979895 (2011).
- [12] Heo, S. and Lee, G.: ForceTap: Extending the Input Vocabulary of Mobile Touch Screens by Adding Tap Gestures, *Proc. 13th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI '11*, pp.113–122, ACM (online), DOI: 10.1145/2037373.2037393 (2011).
- [13] Huawei Technologies Co. Ltd.: HUAWEI Mate S (2015), available from (<https://consumer.huawei.com/jp/phones/mate-s/>) (accessed 2018-11-12).
- [14] Hwang, S., Bianchi, A. and Wohn, K.-Y.: VibPress: Estimating Pressure Input Using Vibration Absorption on Mobile Devices, *Proc. 15th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI '13*, pp.31–34, ACM (online), DOI: 10.1145/2493190.2493193 (2013).
- [15] Hwang, S., Bianchi, A. and Wohn, K.: MicPen: Pressure-sensitive Pen Interaction using Microphone with Standard Touchscreen, *CHI '12 Extended Abstracts on Human Factors in Computing Systems, CHI EA '12*, pp.1847–1852, ACM (online), DOI: 10.1145/2212776.2223717 (2012).
- [16] Hwang, S. and Wohn, K.Y.: PseudoButton: Enabling Pressure-sensitive Interaction by Repurposing Microphone on Mobile Device, *CHI '12 Extended Abstracts on Human Factors in Computing Systems, CHI EA '12*, pp.1565–1570, ACM (online), DOI: 10.1145/2212776.2223673 (2012).
- [17] Hyuga, S., Ito, M., Iwai, M. and Sezaki, K.: Estimate a User's Location using Smartphone's Barometer on a Subway, *Proc. 5th International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, MELT '15*, pp.2:1–2:4, ACM (online), DOI: 10.1145/2830571.2830576 (2015).
- [18] Iwasaki, K., Miyaki, T. and Rekimoto, J.: Expressive Typing: A New Way to Sense Typing Pressure and Its Applications, *CHI '09 Extended Abstracts on Human Factors in Computing Systems, CHI EA '09*, pp.4369–4374, ACM (online), DOI: 10.1145/1520340.1520668 (2009).
- [19] Laput, G., Brockmeyer, E., Hudson, S.E. and Harrison, C.: Acoustruments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices, *Proc. 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pp.2161–2170, ACM (online), DOI: 10.1145/2702123.2702414 (2015).
- [20] Low, S., Sugiura, Y., Lo, D. and Inami, M.: Pressure Detection on Mobile Phone by Camera and Flash, *Proc. 5th Augmented Human International Conference, AH '14*, pp.11:1–11:4, ACM (online), DOI: 10.1145/2582051.2582062 (2014).
- [21] Mandalapu, D. and Subramanian, S.: Exploring Pressure As an Alternative to Multi-touch Based Interaction, *Proc. 3rd International Conference on Human Computer Interaction, IndiaHCI '11*, pp.88–92, ACM (online), DOI: 10.1145/2407796.2407810 (2011).
- [22] Marshall, J., Pridmore, T., Pound, M., Benford, S. and Koleva, B.: Pressing the Flesh: Sensing Multiple Touch and Finger Pressure on Arbitrary Surfaces, *Pervasive Computing: Proc. 6th International Conference, Pervasive 2008 Sydney, Australia, May 19-22, 2008*, pp.38–55, Springer Berlin Heidelberg (online), DOI: 10.1007/978-3-540-79576-6_3 (2008).
- [23] Miyaki, T. and Rekimoto, J.: GraspZoom: Zooming and Scrolling Control Model for Single-handed Mobile Interaction, *Proc. 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '09*, pp.11:1–11:4, ACM (online), DOI: 10.1145/1613858.1613872 (2009).
- [24] Nakai, Y., Kudo, S., Okazaki, R. and Kajimoto, H.: Tangential Force Input for Touch Panels using Bezel-aligned Elastic Pillars and a Transparent Sheet, *Proc. Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14 Adjunct*, pp.111–112, ACM (online), DOI: 10.1145/2658779.2658801 (2014).
- [25] Nakajima, K., Itoh, Y., Hayashi, Y., Ikeda, K., Fujita, K. and Onoye, T.: Emoballoon A Balloon-Shaped Interface Recognizing Social Touch Interactions, *Proc. 10th International Conference on Advances*

in *Computer Entertainment - Volume 8253, ACE 2013*, pp.182–197, Springer-Verlag New York, Inc. (online), DOI: 10.1007/978-3-319-03161-3_13 (2013).

- [26] Ono, M., Shizuki, B. and Tanaka, J.: Sensing Touch Force using Active Acoustic Sensing, *Proc. 9th International Conference on Tangible, Embedded, and Embodied Interaction, TEI '15*, pp.355–358, ACM (online), DOI: 10.1145/2677199.2680585 (2015).
- [27] Pedersen, E.W. and Hornbæk, K.: Expressive Touch: Studying Tapping Force on Tabletops, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '14*, pp.421–430, ACM (online), DOI: 10.1145/2556288.2557019 (2014).
- [28] Ramos, G., Boulos, M. and Balakrishnan, R.: Pressure Widgets, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '04*, pp.487–494, ACM (online), DOI: 10.1145/985692.985754 (2004).
- [29] Rekimoto, J. and Schwesig, C.: PreSenseII: Bi-directional Touch and Pressure Sensing Interactions with Tactile Feedback, *CHI '06 Extended Abstracts on Human Factors in Computing Systems, CHI EA '06*, pp.1253–1258, ACM (online), DOI: 10.1145/1125451.1125685 (2006).
- [30] Rendl, C., Greindl, P., Probst, K., Behrens, M. and Haller, M.: Pressures: Exploring Pressure-sensitive Multi-touch Gestures on Trackpads, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '14*, pp.431–434, ACM (online), DOI: 10.1145/2556288.2557146 (2014).
- [31] Takada, R., Lin, W., Ando, T., Shizuki, B. and Takahashi, S.: A Technique for Touch Force Sensing Using a Waterproof Device's Built-in Barometer, *Proc. 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '17*, pp.2140–2146, ACM (online), DOI: 10.1145/3027063.3053130 (2017).
- [32] Tung, Y.-C. and Shin, K.G.: Expansion of Human-Phone Interface By Sensing Structure-Borne Sound Propagation, *Proc. 14th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '16*, pp.277–289, ACM (online), DOI: 10.1145/2906388.2906394 (2016).
- [33] Wu, M., Pathak, P.H. and Mohapatra, P.: Monitoring Building Door Events using Barometer Sensor in Smartphones, *Proc. 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15*, pp.319–323, ACM (online), DOI: 10.1145/2750858.2804257 (2015).
- [34] Ye, H., Gu, T., Tao, X. and Lu, J.: B-Loc: Scalable Floor Localization using Barometer on Smartphone, *11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS '14*, pp.127–135, IEEE Computer Society (online), DOI: 10.1109/MASS.2014.49 (2014).



Buntarou Shizuki is an associate professor of Department of Computer Science at University of Tsukuba. His research interest is in human-computer interaction. He received B.Sc., M.Sc., and D.Sc. from Tokyo Institute of Technology in 1994, 1996, and 2000, respectively.



Shin Takahashi is an associate professor of Department of Computer Science, University of Tsukuba. His research interests include user interface software and ubiquitous computing. He received B.Sc., M.Sc., and Ph.D. in information science from The University of Tokyo in 1991, 1993, and 2003. He is a member of ACM,

IPJSJ, and JSSST.



Ryosuke Takada is currently a Ph.D. candidate in the Department of Computer Science at University of Tsukuba. His research interests include human-computer interaction and human sensing. He received B.Eng. at Kobe City College of Technology in 2015 and M.Eng. at University of Tsukuba in 2017.



Toshiyuki Ando is currently a graduate student in the Department of Computer Science at University of Tsukuba. His research interests include human-computer interaction. He received a B.S. at University of Tsukuba in 2017.