Voice Augmented Manipulation: Using Paralinguistic Information to Manipulate Mobile Devices

Daisuke Sakamoto[1], Takanori Komatsu[2] and Takeo Igarashi[1]
[1]The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo 113-0033 Japan
[2]Meiji University
4-21-1 Nakano, Tokyo 164-8525 Japan
{d.sakamoto, tkomat, takeo}@acm.org

ABSTRACT
We propose a technique called voice augmented manipulation (VAM) for augmenting user operations in a mobile environment. This technique augments user interactions on mobile devices, such as finger gestures and button pressing, with voice. For example, when a user makes a finger gesture on a mobile phone and voices a sound into it, the operation will continue until stops making the sound or makes another finger gesture. The VAM interface also provides a button-based interface, and the function connected to the button is augmented by voiced sounds. Two experiments verified the effectiveness of the VAM technique and showed that repeated finger gestures significantly decreased compared to current touch-input techniques, suggesting that VAM is useful in supporting user control in a mobile environment.

Author Keywords
Voice input; zooming; panning; navigation; mobile phone/device; tablet; input technique.

ACM Classification Keywords
H5.2 User Interfaces (D.2.2, H.1.2, I.3.6): Interaction styles (e.g., commands, menus, forms, direct manipulation), H5.2 User Interfaces (D.2.2, H.1.2, I.3.6): Voice I/O.

General Terms
Human Factors; Design.

INTRODUCTION

For many people, smartphones and other mobile devices have become indispensable in everyday life. Most of these devices are equipped with a touch screen, so users can directly intuitively manipulate a target appearing on the screen. For example, users can scroll through a long list by sliding a finger over the screen from bottom to top or vice versa ("sliding") and can zoom in and out of web pages and photos by placing two fingers on the screen and spreading them farther apart or bringing them closer together ("pinching").

MobileHCI 2013, Aug 27–30, 2013, Munich, Germany.

Figure 1: Voice augmented manipulation (VAM) accepts users' paralinguistic information to operate (zooming, panning, scrolling, etc.) mobile devices. The technique augments scrolling with user's scrolling gestures and voice (left), and augments zooming with a button press and user's breath (right).

However, it is usually necessary to repeat such actions on the screen because the screens are so small that only a portion of the content (e.g., map images or web pages) is visible. Another problem is that repeated finger gesturing or pressing of navigation buttons often hides the page content and decreases the visible area. It is then assumed that such gestures would lead to user fatigue or a feeling of being inconvenienced.

To resolve the above issues and avoid inconvenient repetition of finger gestures, our idea is to introduce an input method that involves the use of the phone's built-in microphone. Mr. Hugo Barra, while Director of Product Management, Mobile at Google, stated on 12 August 2010[1], *"One out of four queries on Android 2.0 devices in the United States are being made using voice search."* Likewise, the Wall Street Journal reported on 3 March 2012[2], *"87% of Apple's iPhone 4S users use at least one feature of Siri monthly."* We believe that these statements, which indicate an increasing tendency toward the use of voice input, lend support to our idea. As voice input, we are focusing on paralinguistic information, such as the pitch,

[1] http://techcrunch.com/2010/08/12/googles-hugo-barra-25-of-android-queries-are-voice-based/

[2] http://blogs.wsj.com/digits/2012/03/26/apple's-siri-gains-traction-for-some-things/

power, or tone of speech sounds, which well reflects the users' intuitive feelings or attitudes [7]. Since both paralinguistic information and finger gestures on a touch screen are suitable for expressing continuous/analogue input, paralinguistic information could support finger gesturing on the screen.

Here, we propose an intuitive manipulation technique for mobile devices called *Voice Augmented Manipulation (VAM)*. VAM uses both touch and voice input and augments user control in actions through voice input, such as zooming/panning over maps or scrolling through a web page. Zooming/panning on a mobile device requires repeated finger gestures. With the VAM technique, the user first presses a button or makes a finger gesture to manipulate something on the mobile device and voices a sound, such as *"Ahhh."* The operation will then continue until the user stops doing the action or making the sound. We implemented this technique on a smartphone and conducted two experiments to verify its usability and efficiency. The results show that our technique provides users with intuitive control and frees them from repeated finger gesturing on mobile devices.

RELATED WORK

Speech/Voice-based Operation Techniques

Extensive investigations of speech/voice-based interaction have been conducted throughout the history of computer research and numerous related products have been developed. In the context of human-computer interaction (HCI), speech-based I/O [26], speech-based cursor control [8,13,14], and a multimodal interface [3,24] are the main research topics related to speech/voice interaction. Speech/voice is sometimes used with a pen-based interface [21], web browsing interface [6], or an augmented reality system [18]. Speech-based dictation systems are becoming more practical [32]. A voice-augmented multi-touch interaction technique has also recently been proposed [31].

These studies focused on speech recognition for natural user input to computers and have already added great value to mobile devices. In this project, we focused on non-linguistic and paralinguistic user input for augmenting user control on mobile devices.

Paralinguistic Information

Human communication is explicitly achieved through verbal utterances, but paralinguistic information (e.g., the pitch, power, and tone of utterances) also plays an important role [19]. In general, verbal utterances can be regarded as digital information while paralinguistic information can be regarded as analogue information. This is because one's internal state, such as one's emotional state or ambiguous nuances in feeling, are deeply reflected in expressed paralinguistic information [7]. Most studies on paralinguistic information have focused on estimating people's emotional states from their speech sounds [29].

We suggest that paralinguistic information is important for not only understanding people but also for supporting them in computing. Igarashi et al. presented a voice-based interaction method [17] that uses the user's voice commands and paralinguistic information to control a web/document browser and game applications. In their interface, however, the system must use both voice commands and paralinguistic information. We suggest that speech recognition takes a significantly longer time than pressing a button or making a finger gesture in a mobile environment. Harada et al. presented a digital painting method that uses non-linguistic vocalization [11,12] for navigating a pointer and augmenting brush strokes; however, their method takes more time than pressing a button or making a finger gesture as well. Moreover, these previous studies did not conduct any formative user study to measure the task completion time.

Sporka et al. proposed a predictive text entry method for disabled people that use paralinguistic information [35]. Goto et al. used paralinguistic information as a cue in speech recognition [9]. Paralinguistic information has also been used for controlling a robot arm [16]. Studies similar to those for voice input have been conducted for blowing-input interfaces. Al-Hashimi presented a plotter controlled by a user's blowing onto the device [1]. Patel et al. proposed a blowable user interface, which uses the built-in microphone of a laptop computer [28]. The user puffs on the display, and the interface analyzes the user's blowing location. Sporka et al. proposed a method that navigates a pointer and emulates mouse clicks by having the user whistle [34].

We suggest that augmenting a user's input (a finger gesture or button press) with voice input is important in a mobile environment because manipulation with just voice, blowing, or whistling requires more time. We aim to increase the speed of users' interactions on a mobile device with voice input.

User Interface for Mobile Devices

User interfaces for mobile devices are currently a popular research topic in HCI. From the advent of the iPhone by Apple to the present, multi-touch gestures and actions have become increasingly common for end-users. Pinching for zooming in and out and swiping for browsing pictures, music, and video are well known gestures for manipulating objects displayed on mobile devices, but they are not always useful.

Extensive research has been conducted to develop more natural and intuitive interaction methods. Li presented a gesture-based searching technique on mobile devices [22]. Common gestures that can be used on mobile devices have been well researched in HCI [4]. Other important topics in HCI research are navigation and zooming methods on user interfaces for mobile devices. For example, Kratz et al. proposed a semi-automatic zooming method [20]. Baudisch

et al. proposed a zooming method in which irrelevant areas on a small-screen web browser are collapsed [2]. Rohs et al. proposed a map navigation method using a mobile device as a magic lens [30]. Sensors have also been attached to mobile devices to detect user behavior toward devices. Butler presented a sensing technique for detecting user gestures around a mobile device [5]. Miyaki presented pressure-based zooming control on a mobile device [25]. To address the shortcomings of mobile devices' small screens and bodies, Gunn et al. proposed a one-handed multiple-target selection technique [10]. However, the technique is limited to the target selection task.

We suggest that these interaction styles are not directly comparable with the VAM interface, but could be used simultaneously with it in a mobile environment. The VAM interface is simply another way to augment manipulation on mobile devices.

Zooming and Navigation Gestures

Many research groups have investigated gesture input methods for multi-touch displays. Olwal et al. proposed rubbing and tapping gestures to respectively zoom in on and select a target [27]. Malacria et al. proposed zooming and navigation gestures for a multi-touch display [23]. A multi-touch tabletop display with a depth camera has recently been proposed with the capability to sense a user's hands above the display [15]. These techniques might also be usable for mobile devices, but this has not been confirmed. The focus with our VAM technique is on navigation and zooming tasks, but the experiment results will contribute to broadening the range of HCI research.

USAGE SCENARIO AND REQUIREMENTS

To clarify our goal, we first discuss the basic attitudes of users towards voice input and then present usage scenarios that motivated our research.

When Do Users Prefer to Use Voice Control?

Shneiderman et al. suggested that users prefer to use voice control in four situations: when 1) the hands are busy, 2) the eyes are busy, 3) mobility is required, and 4) keyboards and screens would not be appropriate [33]. We suggest that the hands may frequently be busy and mobility will most likely occur in environments where users are not always stationary, i.e., where they can move around with their mobile devices. They can use both hands for interacting with their devices, but there are some difficulties (e.g., if the user looks solely at the device while walking, he or she risks an accident). The eyes would probably also be busy in a mobile environment.

Even if both hands are not busy in a mobile environment, voice control might be preferred by users. Mobile devices are small, but many user actions for control (zooming, pinching gestures for navigating maps, sliding tiny scroll bars, etc.) are required. Since voice input for searching and text input are already common among mobile phone users, such users may accept a voice control interface as a natural interaction format.

Below, we discuss three usage scenarios for the VAM interface.

Scenarios

Scenario A: A busy businessman, Jim, is heading to a client company by car. He uses voice search and the VAM interface to check the location of the company. He cannot operate the system with his hands continuously due to driving laws, so he just touches his mobile device for zooming in on or panning across the map with one hand. The manipulation will continue while he voices a sound. If at any time he stops voicing the sound, he can restart the manipulation just by restarting the voicing so that he does not need to touch the device continuously.

In this scenario, the system is supporting one-handed interaction. Users often use mobile devices while they are doing something else. However, sometime mobile applications expect two-handed interaction. This puts users' safety at risk. The VAM interface supports one-handed manipulation on and intuitive interaction with mobile devices.

Scenario B: Jim comes back home and lies on his back on a sofa to relax. He is browsing web pages and checking Facebook on his tablet by holding it above his face in both hands. He uses the VAM interface to scroll page contents and is freed from having to operate the tablet over and over again with finger gestures. A long, strong voice will scroll faster, and a short, strong voice will select individual items (e.g., selecting e-mail, checking Twitter updates or articles in an RSS reader). By making his voice stronger or weaker, he can control the scrolling speed.

In this scenario, holding a two-pound and ten-inch tablet device in both hands while lying on one's back and scrolling through a web page with many repeated finger gestures would quickly result in tired arms. The VAM interface is useful for repeating operations in this scenario, making it possible for the user to take any position he or she wants and still be able to use a mobile device. The VAM interface supports user manipulation (e.g., scrolling, zooming, etc.) with the voice.

Requirements

The requirements of the VAM interface are summarized below.

Voice Input

Voiced sound to augment interactions is the most important feature of the VAM technique and is becoming common in smartphones (Android by Google; iOS by iPhone). The user's voice operates the mobile device, but users may not always be able to voice a sound.

Commanding Gesture and Operation

The VAM technique augments user manipulations on mobile devices by using users' voice input, so we need to consider what operations are appropriate for the technique. As we mentioned in Related Work, touch-based mobile devices support basic gesture commands, such as tapping, pinching, swiping/panning, scrolling, and flicking. Tapping (pointing) and flicking are common gestures but very simple; most tasks operated with these gestures are single ones, and repeated gesturing is not required for task completion. For example, if a user selects several objects many times, the selection of one object is one task. A flick gesture also completes a single task. For these reasons, for the VAM interface, we selected the pinching, swiping/panning, and scrolling gestures, which are more complex than tapping and flicking.

Some gestures require two hands, such as pinching, so the VAM interface also has a button-based interaction format. The user selects a button for control and voices a sound over the mobile device, and the command that the user selects will be augmented.

VOICE AUGMENTED MANIPULATION (VAM)

VAM consists of two components: a sound recognition unit for capturing the user's paralinguistic information from a microphone and a controller unit for receiving the user-specific commands from a touch screen or keyboard. The sound recognition and controller units function in parallel or simultaneously. The proposed VAM technique has the following advantages compared to other input techniques.

- It focuses on users' intuitive input from paralinguistic information they can easily express and that deeply reflects their intuitive feelings or attitudes.
- It uses users' intuitive intentions from paralinguistic information as "an adverb" on the users' selected command in its role as "a verb." This means that the extracted intentions can be freely added to any kind of input, e.g., touch screen or keyboard.

We implement our technique on a smartphone for panning, zooming, and scrolling actions, which are original features of the device, and on direction buttons on the screen. Specifically, the power information in paralinguistic information is used and proportionally connected to the velocity of panning and zooming; that is, louder speech means faster actions, while softer speech means slower actions.

Our VAM technique is implemented for actions with a touch screen (which we call VAM+Touch) and actions with buttons (which we call VAM+Button).

VAM+Touch

This method uses basic finger gestures that users usually make on a touch interface, such as pinching and sliding, and users' paralinguistic information (Figure 2). The user first makes a gesture on the touch interface then voices a sound towards the smartphone. The chosen operation will continue until the user stops voicing the sound or starts new gesture.

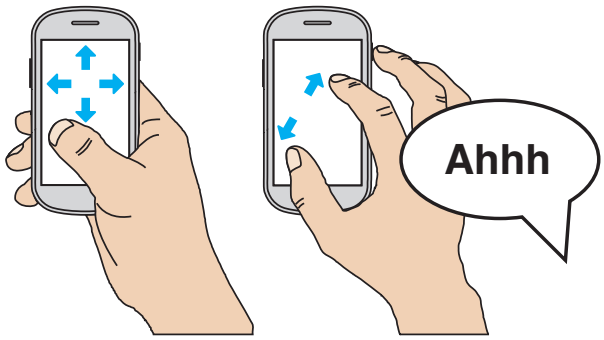

Figure 2: Vam+Touch: User first makes gesture on mobile device then voices sound. Command will then continue until user stops voicing sound: Scrolling/panning gesture (left); pinching gesture (right).

VAM+Button

This method uses buttons on the interface that are related to individual actions, such as "zoom in" and "scroll down", and users' paralinguistic information (Figure 3). The user first presses a button then voices a sound towards the smartphone. The chosen operation will continue until the user releases the button or stops voicing the sound.

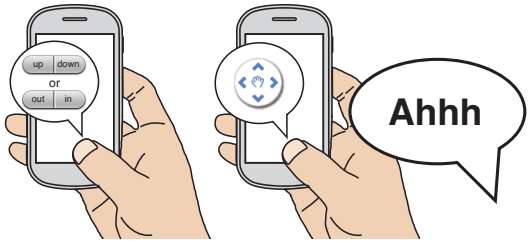

Figure 3: VAM+Button: User presses button on mobile device and voices sound. Command will then continue until button is released or user stops voicing sound Scrolling up/down or zooming in/out (left);navigating map with cross-key button (right).

Interaction Sequence

The basic interaction sequence is the same for both VAM+Touch and VAM+Button (Figure 4). First, the user selects an operation by pressing a button or making a finger gesture, and then voices a sound over the device. The operation will stop when the user 1) releases the button or touches the display again or 2) stops the sound. Even if the user is voicing a sound, the augmentation will stop if the user releases the button. However, in this case, if the user makes a new gesture on the device or presses another button, the new operation will be applied immediately. This also means that if the user voices a sound before pressing a button or making a finger gesture, the augmentation will be applied immediately.

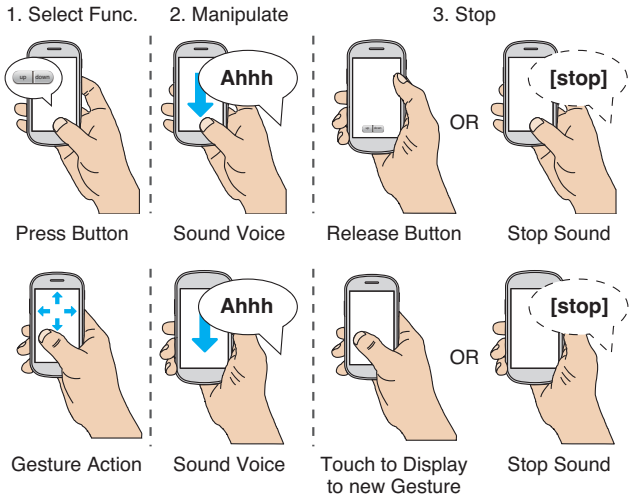

Figure 4: Interaction sequence: VAM+Button (top) and VAM+Touch (bottom).

Implementation

The VAM interface was implemented as an application with the Android Software Development Kit (SDK) 4.0.3 on a Google Nexus S smartphone; each interaction method was part of the application. The application uses microphone input. The user speech signal is sampled through pulse-code modulation at 16 bit/8 kHz. Then the short-time Fourier transform with a 2024-sample Hanning window is calculated using fast Fourier transformation (FFT). The FFT frame is shifted by 1024 samples, and power is measured with a frame size of 128 ms.

Power (sound volume) information is used for the VAM interface. This information is used in an appropriate way for each technique. Basically, increasing the power level will increase the augmentation level and decreasing it will decrease the augmentation level.

EXPERIMENT 1: PANNING

In experiment 1, we compared VAM+Touch and VAM+Button with the standard panning methodology of a "sliding" on mobile devices.

Equipment

The experiment was run on a smartphone, and VAM was implemented with Android SDK 4.0.3 and refreshed at 10 Hz. The display was a four-inch screen with WVGA (480 × 800 px) resolution. The participants were simply instructed to hold the smartphone in their hands; therefore, the distance between the device, specifically the microphone, and the participants was uncontrolled.

Task

We selected a continuous labyrinth task [23]. The participants were asked to follow a continuous labyrinth linking 16 consecutive nodes to a final target. This task mimics what users have to do when searching for a location on a map. The labyrinth that the participants were asked to follow consisted of 16 links that were in random combinations with eight directions (45°, 90°, ..., 360°) and two distances (200 and 800 px), so that the total distance to the goal was always 8,000 px. The system knows a link has been taken if an adjacent node comes into a circle (80 px in radius) located at the center of the interface. Paths and nodes that the user had already taken or visited were displayed in gray, and those not yet used were displayed in green (Figure 5). The circle is displayed in red.

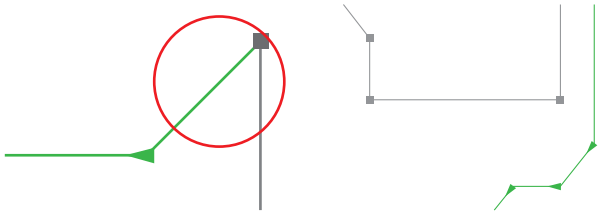

Figure 5: Labyrinth task; detail (left); map (right).

Implementation

We implemented the three interaction methods for the task (Figure 6). For VAM+Touch, we implemented an interface with sliding and voice input capability. Sliding is augmented by the user's voice and continues until the user stops voicing the sound or touches the mobile device again to make another gesture. For VAM+Button, we implemented an interface that displays a button at the center-bottom of the screen. By touching this button, the user specifies the direction of the navigation. Then, the user voices a sound. Navigation in the specified direction will continue until the user releases the button or stops the sound. Sliding was implemented as a basic action on the mobile device. By simply sliding a finger on the display, the user can navigate the map.

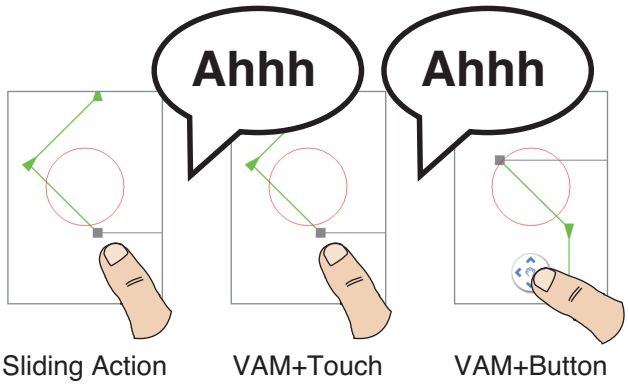

Figure 6: Interaction method for Experiment 1

All sliding actions are implemented as momentum-based sliding, where sliding should continue for some period and glide to a stop after the users stop their actions. Momentum-based sliding or scrolling is now a standard function in mobile devices, so it was used as the baseline condition for this experiment. In the VAM cases, sliding and VAM were

implemented with the momentum-based one to keep the usability the same as the baseline condition (sliding action).

Participants

Twenty university students majoring in chemistry, biology, and mechanical engineering (15 males and 5 females; 20 – 24 years old) participated in this experiment. Eighteen of the participants own a smartphone (90%), and use the smartphone for one to twelve hours a day (*Mean=3.444*, *SD=2.482*). Fourteen participants had experience in using voice search on mobile devices.

They were asked to use the three different input methods (VAM+Touch, VAM+Button, and sliding) three times each. The experiment consisted of three sets of tasks, with each set consisting of three labyrinths to be followed using each of the three input methods, and each participant completed all three sets (in total, nine labyrinths). We prepared the nine different labyrinths so that no participant experienced the same labyrinth within his or her nine trials, and the order of the labyrinths and the input methods were counterbalanced among the participants. Prior to the experiment, the participants completed two training sets per method.

The independent variables were the type of input method and the order in which the set was experienced (set order), while the dependent variable was the participants' performance. This performance was measured from the completion time of each labyrinth and the number of finger gestures needed to complete the task.

Quantitative Results

Completion Time

The average completion times for the three input methods in each set are depicted in Figure 7. These average times were analyzed using a two-way analysis of variance (ANOVA) (within-subject plan, with type of input method and set order as independent variables and completion time as the dependent variable). The results showed no significant differences in interaction effects [$F(4,76)=1.83$, n.s.] and marginally significant differences in the main effect of the set order [$F(2,38)=2.95$, $p<.10$], but there were significant differences in the main effect of the input method [$F(2,38)=53.00$, $p<.01$].

A multiple comparison using a least significant difference (LSD) test on the simple main effect of the input method showed that the completion time for the VAM+Button method was significantly longer than for the other two (LSD=6163.2792, MSe=278068812.3, 5% level). A multiple comparison using an LSD test on the simple main effect of the set order showed that the completion time in the third set was significantly shorter than in the first set (LSD=3968.3394, MSe=115277941.1, 5% level).

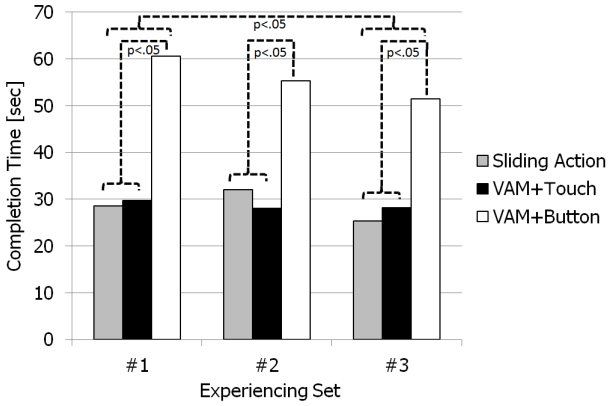

Figure 7: Average completion time of three input methodologies in each set in labyrinth task.

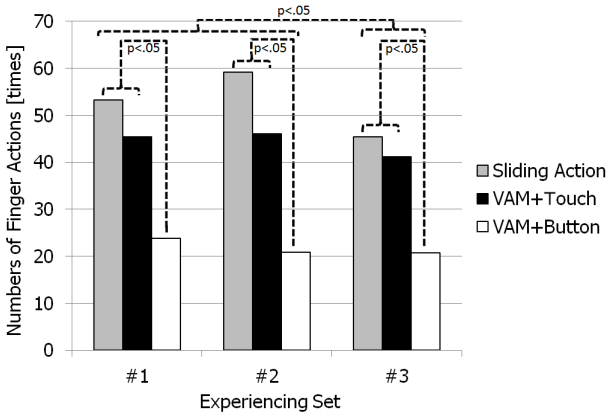

Figure 8: Average numbers of finger actions of three input methodologies in each set in labyrinth task.

Number of Finger Gestures

We recorded all users' activities on the mobile device during the experiment. For this analysis, we counted their finger gestures (sliding), which indicated how many times the participants touched the screen before finishing each labyrinth. In the case of VAM+Button, we counted the number of navigation button presses.

The average numbers of finger gestures for the three input methods in each set are depicted in Figure 8. These average times were analyzed using two-way ANOVA (within-subject plan, with type of input methodology and set order as independent variables and number of finger gestures as the dependent variable). The results showed no significant differences in the interaction effects [$F(4,76)=1.96$, n.s.] but significant differences in the main effect of the input method [$F(2,38)=27.70$, $p<.01$] and in the main effect of the set order [$F(2,38)=5.68$, $p<.01$].

A multiple comparison using an LSD test on the simple main effect of the input method showed that the number of finger gestures for the VAM+Button method was significantly smaller than for the other two (LSD=8.6765, MSe=551.0839, 5% level). A multiple comparison using an

LSD test on the simple main effect of the set order showed that the numbers in the third set was significantly smaller than those in the other two sets (LSD= 4.0043, MSe=117.3751, 5% level).

EXPERIMENT 2: ZOOMING

In experiment 2, we compared VAM+Touch and VAM+Button with the standard zooming method of "pinching" and with using a "sliding bar" on mobile devices. The experiment was run on the same smartphone as in experiment 1.

Task

We selected a multi-scale pointing task [23]; that is, the participants were asked to reach and validate a series of four circular targets of 40 px in diameter, each of which was 2,400 px away from the preceding one. The user first selected a point to zoom in on, and did an operation using the four manipulation techniques. Validation of the target required capturing at least 30 px of its diameter, so the participants were required to zoom in sufficiently on the target. Once the participants had validated a target, the target disappeared and a new one instantly appeared at another position 2,400 px away. This new target was not visible at first because it was far away enough that zooming out was required in order to see it. After zooming out and finding the target, the users were required to point to a certain position to decide the starting point for zooming and then start zooming in to validate the target. This task mimicks what users have to do when searching for a location on a map (zooming in and out).

Implementation

We implemented the four interaction methods accordingly for the task (Figure 9). For VAM+Touch, we implemented an interface with pinching and voice input capability. Pinching is augmented by the user's voice and continues until the user stops voicing the sound or touches the mobile device again to make another gesture. For VAM+Button, we implemented an interface that displays a button at the center-bottom of the screen. The button has two functions: zooming in and zooming out. By touching this button, the user instructs the system to zoom in or out. Then, the user voices a sound. Zooming in/out will continue until the user releases the button or stops voicing the sound. Pinching was implemented as a basic action on the mobile device. By simply making a pinching gesture on the display, the user can zoom in/out on the map. We also implemented a sliding bar, which is common for zooming in/out on maps. By sliding this bar, the user can zoom in or out on the map.

Participants

The same twenty university students participated. . They were asked to use the four different input methods (VAM+Touch, VAM+Button, Pinching, and Sliding Bar) three times each. The positions of the four targets in the multi-scale task were randomly generated in each trial so

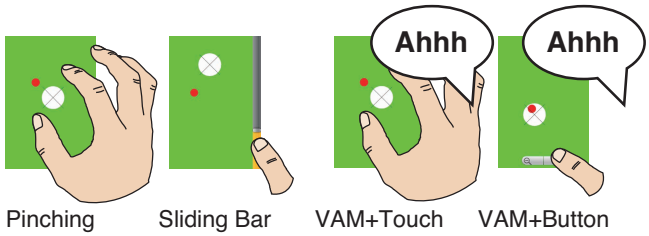

Figure 9: Interaction method for experiment 2.

that no participant experienced the same task setting within his or her twelve trials, and the order of the input methods was counterbalanced among the participants. Prior to the experiment, the participants completed two practice sets per method. The settings of the independent and dependent variables were the same as in experiment 1.

Quantitative Results

Due to technical problems during the experiment, the system failed to log four datasets. We excluded them for the quantitative analysis.

Task Completion Time

The average completion times for the four input methods in each set are shown in Figure 10. These average times were then analyzed using a two-way ANOVA (within-subject plan, with type of input methodology and set order as independent variables and completion time as the dependent variable:). The results showed no significant differences in the interaction effects [$F(6,90)=1.10$, n.s.] and the main effect of the set order [$F(2,30)=1.33$, n.s.], but there were significant differences in the main effect of the input method [$F(3,45)=33.84$, $p<.01$].

A multiple comparison using an LSD test on the simple main effect of the input method showed that the completion time for VAM+Touch was significantly shorter than that for Pinching and longer than those for Sliding Bar and VAM+Button. It also showed that the completion time for Sliding Bar was significantly shorter than that for Pinching

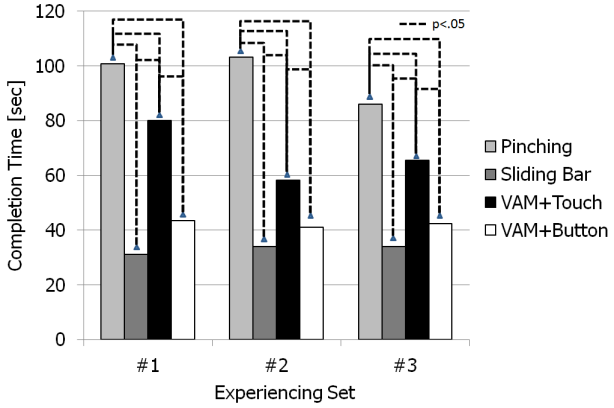

Figure 10: Average completion times of four input methods in each set in zooming task

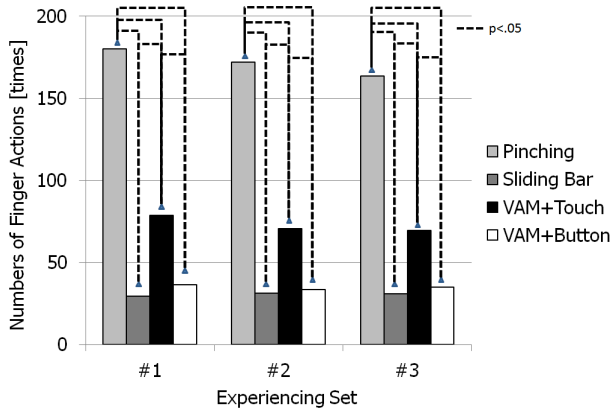

Figure 11: Average numbers of finger gestures of four input methods in each set in zooming task

and that that for Pinching was longer than that for VAM+Button. To summarize, the completion times for VAM+Button and Sliding Bar were significantly shorter than for VAM+Touch and Pinching (LSD=13982.0429, MSe=1156619319.0, 5% level).

Number of Repeated Finger Gestures

We recorded all users' activities on a mobile device during the experiment. We counted users' finger gestures (selecting a point to zoom in/out (Pointing), pressing a button (Pressing), pinching action (Pinching) and using a slider bar (Sliding)). For instance, the types of finger gestures in VAM+Touch, VAM+Button, Pinching and Sliding were pointing and pinching, pointing and pressing, pointing and pinching, and pointing and sliding, respectively.

The average numbers of finger gestures for the four input methods in each set are shown in Figure 11. These average times were analyzed using a two-way ANOVA (within-subject plan, with type of input methodology and set order as independent variables and number of finger actions as the dependent variable). The results showed no significant differences in the interaction effects [$F(6,90)=0.35$, n.s.] and the main effect of the set order [$F(2,30)=0.60$, n.s.], but there were significant differences in the main effect of the input method [$F(3,45)=85.74$, $p<.01$].

A multiple comparison using an LSD test on the simple main effect of the input methodology showed that the number of finger gestures for VAM+Touch was significantly larger than for Sliding Bar and VAM+Button and smaller than for Pinching, that for Sliding Bar was significantly smaller than that for Pinching, and that for Pinching was larger than that for VAM+Button. To summarize, the numbers of finger gestures for VAM+Button and Sliding Bar were significantly smaller than for VAM+Touch and Pinching (LSD=20.2000, MSe=2414.0921, 5% level).

SUMMARY OF EXPERIMENTAL RESULTS

Overall, the participants successfully used the VAM interface. We confirmed the following points from the results of the two experiments

- In the labyrinth task (Exp. 1) for investigating the participants' behaviors during panning, VAM+Touch resulted in nearly the same performance (completion time) as standard sliding. In addition, even though there were no significant differences statistically, the average frequency of finger gestures in VAM+Touch was smaller than for Sliding Action. This means that the participants did not repeat sliding with VAM+Touch as much as they did with Sliding.
  VAM+Button was statistically slower than the other two methods (completion time). However, the average frequency of finger gesture was significantly smaller than the other two methods.
- In the pointing task (Exp. 2) for investigating the participants' zooming behaviors, VAM+Button resulted in significantly better performance (completion time and finger gestures) than Pinching and in nearly the same performance as the standard Sliding Bar. Additionally, the frequency of finger gestures was smaller than for Pinching.
  However, the standard Sliding Bar was faster than all other methods except VAM+Button. We consider this was because the zooming function was created as a non-step one and because the sliding action is a very simple one even in a mobile environment. But, we consider that the sliding action needs two-handed operation. On the other hand, VAM+Button allows the user to manipulate the device with just one hand.

Before each experiment started, we asked participants to answer a biographical questionnaire, which included a question about their openness in sounding a voice towards a mobile device when near other people. Eleven of the 20 participants answered that they were open to using voice input. We also asked them to answer a questionnaire after each experiment to collect their overall impression of voice control. Twelve of the 20 participants answered that they thought voice operation was useful. As we mentioned in the introduction, the number of users who use Google Voice Search is increasing; 87% of iPhone users use Siri at least once a month. We believe that voice input is becoming popular among smartphone users.

From these results, we conclude the following. 1) Both VAM methods have an advantage over pinching methods in zoom operations. 2) VAM+Touch (sliding and voice input) shows the almost same performance as general sliding input; however, the results suggest that users fewer finger gestures are required.

APPLICATIONS AND POSSIBLE EXTENSIONS

From the experimental results, we present possible applications and extensions for future implementation of the VAM interface.

Maps

Navigating maps is one of the main target interactions of the VAM interface. From the experiment results, the VAM interface can be used for navigation and zooming interactions in map applications. In particular, zooming by the pinching action was not efficient; a map application with navigation through VAM+Touch (sliding action) and VAM+Button is a feasible interaction. Interaction through VAM+Touch and VAM+Button is easily distinguished by the system and provides users a natural way to navigate the map.

Blowing Input

Even though people are becoming more open to using voice input, sometimes they may hesitate to use it to prevent disturbing surrounding people. In the experiment, six of the 20 participants tried to blow in the smartphone to augment their manipulation, instead of sounding a voice. The VAM technique just used the volume of the voice in this implementation, but the blowing input worked to manipulate the interface as well. This shows that the VAM technique is available not only for sounding a voice to a smartphone but also for blowing onto one. For blowing input, the user may blow like "Huuuu, Hu! Hu! Hu!" and the manipulation will be augmented.

LIMITATIONS

Our VAM technique is a proof-of-concept implementation for the experiments, so the accuracy of the voice inputs was not calibrated rigidly for various types of smartphones. We believe that the sound volume picked up by various mobile phone microphones will differ, and the application/interface should be calibrated in each mobile platform for practical use. Additionally, the volume may be changed by users. Automatic volume calibration will support practical use of the VAM interface.

The VAM technique only uses voice power information, but the system could include more degrees of freedom, such as the pitch, tone, rate of utterance, and voice quality. In practice, a simple interface and interaction instead of multiple controls would keep the interface robust and stable, because paralinguistic information contains complicated information that is difficult to separate. However, we believe that a system with multiple controls will support users much better than a single-control one. What information is effective for robust and useful control is still an open question and should be clarified in future research.

The experiments described in this paper were conducted in an experimental room; we ignored the effect of noise in a real-world environment. Currently, the VAM technique may erroneously accept noise as control input. There are many embeddable noise rejection techniques, and the VAM technique may include them. However, the effect of such a technique on voice control is unclear. To make the system practical, we have to first test these techniques and improve the quality and accuracy of recognition of voice input in various real-world situations.

Finally, in this paper, we did not examine the social acceptability of using paralinguistic sounds for the manipulation. Whether non-speech sound is acceptable or not is an open research question. . Though an investigation of social acceptability will be necessary for creating a commercial product, we believe that this paper shows the potential of the VAM technique on mobile devices.

CONCLUSION

We proposed the VAM technique, which augments user manipulation on mobile devices. There are two methods with VAM. One is VAM+Touch, which uses a user's voice and finger gestures (scrolling, pinching, etc.), and the other is VAM+Button, which uses a user's voice and a button interface. We conducted two experiments to evaluate these two methods. The results showed that VAM+Touch helps users navigate and zoom in on maps and that VAM+Button can be used for zooming. These methods can be used simultaneously to support users in operating mobile devices and their various applications.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 24700112 & 25330319.

REFERENCES

1. Al-Hashimi, S. Blowtter: A voice-controlled plotter. In *Proc. HCI 2006 Engage, The 20th BCS HCI Group conference in co-operation with ACM*, vol. 2, (2006), 41-44.
2. Baudisch, P., Xie, X., Wang, C., and Ma, W.-Y. Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. In *Proc. UIST 2004*, ACM Press (2004), 91-94.
3. Bolt, R.A. "Put-that-there": Voice and gesture at the graphics interface. In *Proc. SIGGRAPH Comput. Graph.* 14, 3 (1980), 262-270.
4. Bragdon, A., Nelson, E., Li, Y., and Hinckley, K. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proc. CHI2011*, ACM Press (2011), 403-412.
5. Butler, A., Izadi, S., and Hodges, S. SideSight: multi-"touch" interaction around small devices. In *Proc. UIST 2008*, ACM Press (2008), 201-204.
6. Christian, K., Kules, B., Shneiderman, B. and Youssef, A. A comparison of voice controlled and mouse controlled web browsing, In *Proc. ASSETS 2000*, ACM Press (2000), 72.

7. Cohen, P. R., Morgen, J. and Pollack, M. E. Intentions in Communication, The MIT Press, MA, USA, 1990.
8. Dai, L., Goldman, R., Sears, A., and Lozier, J. Speech-based cursor control: a study of grid-based solutions. In *Proc. ASSETS 2004*, ACM (2004), 94-101.
9. Goto, M., Kitayama, K., Itou, K., and Kobayashi, T. Speech Spotter: On-demand Speech Recognition in Human-Human Conversation on the Telephone or in Face-to-Face Situations. In *Proc. INTERSPEECH 2004 - ICSLP*, ISCA Archive (2004), 1533-1536.
10. Gunn, T.J., Zhang, H., Mak, E., and Irani, P. An evaluation of one-handed techniques for multiple-target selection. In *Proc. CHI EA 2009*, ACM Press (2009), 4189.
11. Harada, S., Saponas, T.S., and Landay, J.A. VoicePen: augmenting pen input with simultaneous non-linguisitic vocalization. In *Proc. ICMI2007*, ACM Press (2007), 178-185.
12. Harada, S., Wobbrock, J.O., and Landay, J.A. Voicedraw: a hands-free voice-driven drawing application for people with motor impairments. In *Proc. ASSETS2007*, ACM Press (2007), 27-34.
13. Harada, S., Landay, J. A., Malkin, J., Li. X. and Bilmes, J. A. The vocal joystick: evaluation of voice-based cursor control techniques, In *Proc. ASSETS 2006*, ACM Press (2006), 174-204.
14. Harada, S., Wobbrock, J. O., Malkin, J., Bilmes, J. A. and Landay, J. A. Longitudinal study of people learning to use continuous voice-based cursor control, In *Proc. CHI 2009*, ACM Press (2009), 347-356.
15. Hilliges, O., Izadi, S., Wilson, A.D., Hodges, S., Garcia-Mendoza, A., and Butz, A. Interactions in the air: adding further depth to interactive tabletops. In *Proc. UIST 2009*, ACM Press (2009), 139-148.
16. House, B., Malkin, J. and Bilmes, J. The VoiceBot: a voice controlled robot arm, In *Proc. CHI 2009*, ACM Press (2009), 183-192.
17. Igarashi, T. and Hughes, J. F. Voice as sound: using non-verbal voice input for interactive control, In *Proc. UIST 2001*, ACM Press (2001), 155-156.
18. Irawati, S., Green, S., Billinghurst, M., Duenser, A., and Ko, H. "Move the couch where?": developing an augmented reality multimodal interface. In *Proc. ISMAR 2006*, (2006), 183-186.
19. Kendon, A. Do gestures communicate? A Review. *Research on Language and Social Interaction 27*, 3 (1994), 175-200.
20. Kratz, S., Brodien, I., and Rohs, M. Semi-automatic zooming for mobile map navigation. In *Proc. MobileHCI 2010*, ACM Press (2010), 63.
21. Kurihara, K., Goto, M., Ogata, J., and Igarashi, T. Speech pen: predictive handwriting based on ambient multimodal recognition. In *Proc. CHI 2006*, ACM Press (2006), 851-860.
22. Li, Y. Gesture search: a tool for fast mobile data access. In *Proc. UIST 2010*, ACM Press (2010), 87-96.
23. Malacria, S., Lecolinet, E. and Guiard, Y. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach, In *Proc. CHI 2010*, ACM Press (2010), 2615-2624.
24. Mitchell, C. and Forren, M. Multimodal User Input to Supervisory Control Systems: Voice-Augmented Keyboard. *IEEE Trans. Syst., Man, Cybern.,Syst 17*, 4 (1987), 594-607.
25. Miyaki, T. and Rekimoto, J. GraspZoom: zooming and scrolling control model for single-handed mobile interaction. In *Proc. MobileHCI*, ACM Press (2009), Article 11 , 4 pages.
26. Nass, C. and Brave, S. Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship, The MIT Press, MA, USA, 2005.
27. Olwal, A., Feiner, S. and Heyman, S. Rubbing and tapping for precise and rapid selection on touch-screen displays, In *Proc. CHI 2008*, ACM Press (2008), 295-304.
28. Patel, S.N. and Abowd, G.D. Blui: low-cost localized blowable user interfaces. In *Proc. UIST 2007*, ACM Press (2007), 217–220.
29. Pierre-Yves, O. The production and recognition of emotions in speech: features and algorithms. *INT J HUM-COMPUT ST 59*, 1-2 (2003), 157-183.
30. Rohs, M., Schöning, J., Raubal, M., Essl, G., and Krüger, A. Map navigation with mobile devices: virtual versus physical movement with and without visual context. In *Proc. ICMI 2007*, ACM Press (2007), 146-153.
31. Schnelle-Walka, D. and Döweling, S. Speech Augmented Multitouch Interaction Patterns. In *Proc. EuroPLOP 2011*, (2011).
32. Sears, A., Feng, J., Oseitutu, K., and Karat, C.-M. Hands-free, speech-based navigation during dictation: difficulties, consequences, and solutions. *Hum.-Comput. Interact. 18*, 3 (2003), 229-257.
33. Shneiderman, B., Plaisant, C., Cohen, M., and Jacobs, S. Designing the User Interface: Strategies for Effective Human-Computer Interaction. (2009).
34. Sporka, A., Kurniawan, S., and Slavik, P. Whistling User Interface (U^3I). In *Proc. ERCIM WG UI4ALL 2004*, LCNS 3196, Springer (2004), 472-478.
35. Sporka, A.J., Felzer, T., Kurniawan, S.H., Poláček, O., Haiduk, P., and MacKenzie, I.S. CHANTI: predictive text entry using non-verbal vocal input. In *Proc. CHI 2011*, ACM Press (2011), 2463-2472.