BlindSight: Eyes-Free Access to Mobile Phones

Kevin A. Li
Computer Science and Engineering
University of California, San Diego
k2li@cs.ucsd.edu

Patrick Baudisch, Ken Hinckley
Microsoft Research
Redmond, WA
{baudisch, kenh}@microsoft.com

ABSTRACT

Many mobile phones integrate services such as personal calendars. Given the social nature of the stored data, however, users often need to access such information as part of a phone conversation. In typical non-headset use, this requires users to interrupt their conversations to look at the screen.

We investigate a counter-intuitive solution: to avoid the need for interruption we replace the visual interface with one based on auditory feedback. Surprisingly, this can be done without interfering with the phone conversation. We present *blindSight*, a prototype application that replaces the traditionally visual in-call menu of a mobile phone. Users interact using the phone keypad, without looking at the screen. BlindSight responds with auditory feedback. This feedback is heard only by the user, not by the person on the other end of the line.

We present the results of two user studies of our prototype. The first study verifies that useful keypress accuracy can be obtained for the phone-at-ear position. The second study compares the blindSight system against a visual baseline condition and finds a preference for blindSight.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. Input devices and strategies; B 4.2 Input Output devices

Keywords: mobile phone, devices, eyes-free, audio, non-speech audio, user interfaces.

INTRODUCTION

Many mobile devices now integrate functionality traditionally spread across multiple devices. These “smart” phones offer, for example, personal calendars in addition to contact lists and phone functionality. Since personal information is particularly important in social scenarios, users often need access while talking on the phone. This can impact phone conversations, as illustrated by the following scenario:

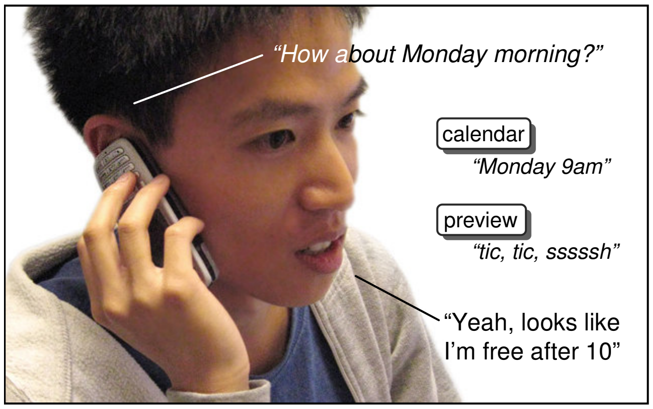
“How about Monday morning?”
calendar
“Monday 9am”
preview
“tic, tic, ssssh”
“Yeah, looks like I’m free after 10”

Figure 1. To access phone information stored on the phone mid-conversation, users press buttons and receive auditory confirmation. This photo shows the *flipPhone* form factor.

John: Hi Ami, can we meet sometime next week?

Ami: Let me check my calendar. Hold on.

Ami moves her phone away from her ear so she can look at it. She opens the calendar application and navigates to next week.

When did you have in mind?

John: How about Tuesday morning sometime?

Ami: Let me check. Hold on.

Ami looks at her phone again, navigates to Tuesday, inspects it, then she puts her phone back to her ear.

What did you say? Oh, yeah, no… I’m only free 3-4.

John: Sorry, I have meetings all afternoon. How does Wednesday afternoon look?

Ami: Hold on, let me see…

The traditional interaction model requires users to look at the screen, which is impossible while the phone is held against the user’s ear. Moving the phone back and forth to the ear interferes with the conversation.

Headsets offer one way to approach this problem. Although headsets are well entrenched in certain user groups and in some cultural settings, many users don’t use headsets because they interfere with real-world situational awareness and are often judged as uncomfortable, unattractive, or socially awkward [8,16]. Even with a headset, accessing visual information requires looking at the screen, which can interfere with other tasks requiring visual attention, such as walking or driving. Speakerphones are subject to the same limitations; in addition they can raise privacy concerns.

We present *blindSight*, a mobile application that provides users with access to personal information stored on their mobile phone while talking on the phone. Users control blindSight using the built-in phone keypad; information and confirmations are delivered via auditory feedback heard only by the user, not by the other person on the other end of the line.

A formative survey of nine users revealed that people need information access during phone conversations and find this situation problematic with current visually-driven phone interfaces. *Calendar* access and *Add Contact* were the most common in-conversation actions requested by survey participants, which informed the design of blindSight.

To provide a hardware basis for blindSight, we present a series of simple modifications to consumer phones that enable eyes-free, one-handed operation, including the configuration shown in Figure 1. We conducted an experiment that shows that this allows users to achieve eyes-free error-rates below 5%. The experiment also revealed that the overhead for eyes-free use is only 200ms per keystroke compared to sighted use.

In a final qualitative user study, 7 out of 8 of participants indicated a preference or strong preference for blindSight over a traditional smart mobile phone. Study tasks included negotiating meetings and managing contacts on the phone.

RELATED WORK

BlindSight builds on two main areas of research: auditory feedback and mobile input.

Auditory feedback

The strengths and weaknesses of auditory feedback have been studied extensively in the field of interactive voice response systems [21]. One of the main challenges is that audio prompting forces users to wait (resulting in "touch tone hell" [32]). Users should be able to "dial through" to interrupt prompts, or "dial ahead" to skip familiar prompts [1]. Perugini et al. propose dial ahead using speech [22]. *Skip and Scan* allows users to iterate through menu options on a telephone using forward and backwards keys, rather than having to listen to a prompt [23]. *Zap and Zoom* improves on *Skip and Scan* by allowing users to jump directly to a location using shortcuts [14]. Yin and Zhai proposed using a visual channel in parallel to using an interactive voice response system to inform users about their options [32], but this is counter to our design goal of eyes-free interaction.

While any human-human conversation contains a certain amount of redundancy [28, 7], weaving auditory information into the phone conversation risks interference. One approach to avoid interference is to time-compress utterances and then serialize them, as suggested by Dietz and Yerazunis who used this approach for recovering phone conversations after interruptions [7]. Tucker and Whittaker compare leaving out words with increasing playback speed [28]. Non-speech audio may be less distracting than speech audio, but can convey information such as navigational cues in hierarchical menus [6, 9, 15]. Zhao et al. explored eyes-free menus driven by auditory cures [33].

Tactile feedback offers another alternative. For example, Luk demonstrates piezoelectric-driven feedback for mobile devices [17].

Mobile input

BlindSight allows for one-handed input using a phone keypad. Keyboard-based entry with few buttons can be supported through iteration [20] or through chording (e.g. the *Twiddler* keypad [19]).

In some contexts, gestures can enable experts to perform eyes-free operations. For example, text entry based on *Unistroke* [10] or *EdgeWrite* [31] can become nearly eyes free, even with distractions [11].

One of the form factors we explore in this paper receives input on the back of the device. *BehindTouch* [13], *HybridTouch* [26], and the isometric joystick-based version of EdgeWrite [30] also explore using the back surface of mobile devices. *LucidTouch* [29] enhances back-of-device interaction by visualizing the user's hand position.

Mobile phone interaction and in-car navigation [1,2] can sometimes successfully employ speech recognition. In situations with a fixed and small vocabulary very good recognition rates have been achieved [12]. If used during a phone conversation, speech input can interfere with the conversation. Only in part can this be reduced by integrating speech commands meaningfully into the conversation (*dual-purpose speech* [18]).

SURVEY OF MOBILE USERS

To inform the design process, we interviewed 9 Smartphone users (2 female) ranging in age from 28 to 45 (median 36) about their usage habits. Our goal was to understand the tasks that users perform while talking on the phone. The resulting list of tasks informs the functionality required for blindSight (which tasks are needed, and how should they be organized) as well as the hardware design (how many buttons are needed).

Participants were recruited from within our institution via email. Interviews lasted approximately 30 minutes per participant. Participants owned a variety of phones; three of them used PDA phones. Average reported monthly talk time was approximately 400 minutes.

Results

Figure 2 summarizes our findings, highlighting the nine most desired tasks while talking on the phone. Access to the calendar was desired by all but two participants. Together, eight out of nine participants expressed that they would like support for these tasks, with seven rating this functionality as *very important*.

These findings suggest that our system should support at least *Add Contact, Find Contact,* and *Navigate Calendar.* While adding meetings and checking the calendar were listed as separate calendar tasks, several participants expressed that these tasks were often intertwined, which led us to combine them into a single task when designing blindSight.

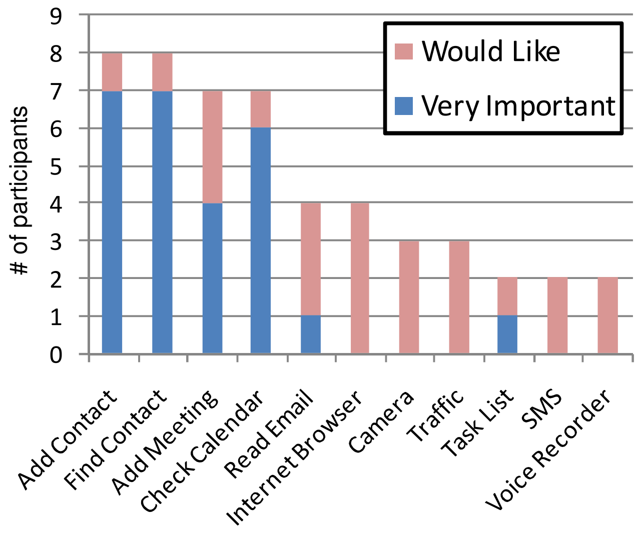

Figure 2. Number of participants out of nine who rated the respective feature as "would like" or "very important"

With respect to hardware design, the need for at least 10 buttons is suggested by the highly desired *Add Contact* task. This led us to use the 3x4 keypad found on traditional mobile phone form factors, rather than creating a custom key layout.

BLINDSIGHT'S AUDITORY EYES-FREE INTERACTION

BlindSight implements eyes-free access to the phone. Users control blindSight by pressing buttons on their phone and receive confirmation by means of auditory feedback. In this section we present the design rationale of the auditory menu, our menu organization, and then a walkthrough.

Design principles

The rationale behind using auditory feedback during a phone conversation is that any human-human conversation contains a certain amount of redundancy [7,28]. If part of the conversation is lost, e.g., because of drop-outs in the line or because a loud truck drove by, users can typically continue the conversation, as long as the inference is short and does not take place at a critical moment. To achieve this, we used the following 5 design guidelines:

1. Feedback only on-demand: blindSight plays auditory feedback *only* in immediate response to a user request. BlindSight never initiates auditory output. Putting timing under user control allows users to wait for an appropriate moment and to avoid moments where important information is communicated, such as a phone number.

2. Brevity: blindSight administers audio feedback in the very brief chunks—a single syllable whenever possible. This minimizes the risk of interference with the conversation.

3. Decomposition: To avoid long blocks of auditory feedback, blindSight breaks down composites, such as lists of menu items or appointments. Instead of presenting them all at once, users iterate though them separately initiating the playback of each item. When iterating through the calendar in 30min steps, for example, each step results in only 1-2 syllables conveying time and availability of the current time slot. Similarly, users block out a calendar items by repeatedly pressing a *block-and-advance* key (similar to the *toggle maps* calendar [4]), rather than entering start and end time.

4. Non-speech previews of composites: To give previews for the 3-hour and full-day calendar views, blindSight presents composites in their entirety. BlindSight creates these previews as a concatenation of discrete 40ms earcons (white noise for "available" and a buzzing sound for "blocked out") with 20ms spaces in-between. This use of non-speech audio minimizes feedback length.

5. Interruptability: By aiming for brevity and decomposition, most auditory elements in blindSight are only one or two syllables long. Exceptions are the task names forming the main menu (such as "hear text messages"). Full names are important here to allow for improved discoverability and learnability—essential in an eyes-free system. To minimize interference with the conversation, blindSight allows users to interrupt audio playback.

BlindSight's main menu combines several of the principles listed above. BlindSight's main menu is quiet when entered (*feedback only on-demand*). Hitting a button causes it to speak out only that button's functionality, such as "add contact" (*decomposition*, *discoverability*). Hitting a button again enters the menu for the respective function. Experienced users can preempt the announcement of the menu name by double-pressing in quick succession (*interruptability*), which turned out to be faster than the use of a separate confirm button.

Menu organization

Figure 3 shows blindSight's menu structure. All menus are based on the 3x4 key numeric portion of a traditional phone keypad, i.e., without additional buttons such as a directional-pad or soft keys. This was informed by our work on keypad form factors, which we present later in this paper.

Each menu is derived from one of the two patterns shown in Figure 4. The *menu* pattern offers fast access to menus containing a small number of choices, and also works for digits and T9 text entry. The *iterator* pattern, in contrast, allows users to traverse long lists using different step sizes or contents organized in a hierarchy.

The *home*, *find contact*, and *add contact* menus (Figure 3a-c) follow the menu pattern; all other menus follow the iterator pattern. We considered implementing *find contact* using an iterator pattern, but opted for the faster and quite common approach of pre-filtering by typing part of the desired name or phone number using T9. To keep the responses short, blindSight responds with the number of matches rather than by spelling out matches. When users decide that the number of matches is small enough, they iterate through the remaining choices.

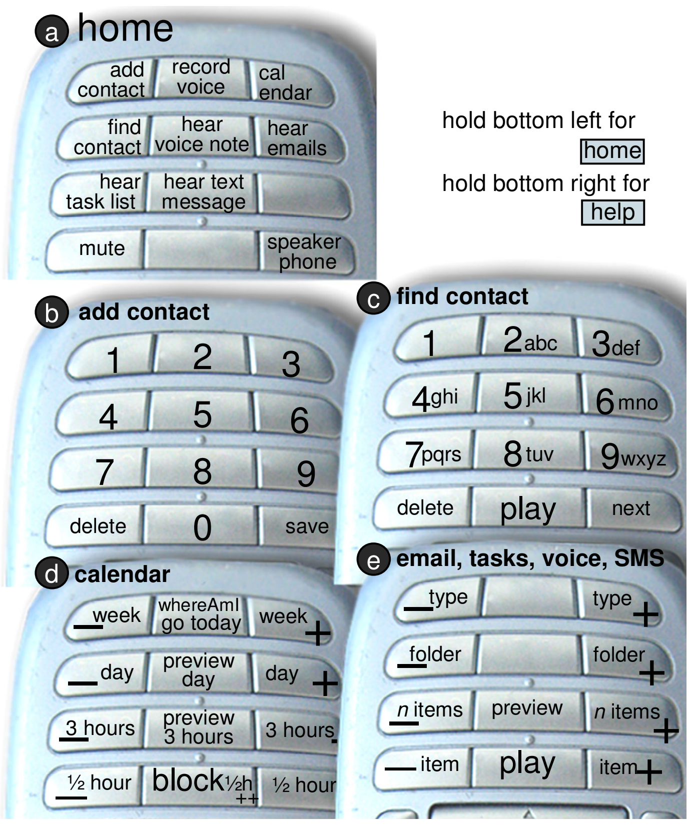

Figure 3. BlindSight's menus

Each submenu implements one of the tasks identified during the survey, with *Add Contact* and *Find Contact* as separate tasks, and *Calendar* as one task. *Add Contact* and *Calendar* are assigned to the prominent corner positions, because they were judged most relevant during our survey,

Mode switches are generally considered problematic [27], and are even more problematic for eyes-free applications. We minimized mode switching by avoiding multi-step menus or wizards. Our first calendar design used a two-menu sequence for picking a date and a time. We resolved this by deriving calendar from the *iterator* pattern instead. In the final design shown in Figure 3, each submenu holds the entire interface require for completing a task. The main menu functions *Mute*, *Speakerphone*, and *Record Voice* simply toggle the respective function, again avoiding mode switches.

BlindSight limits the information users can enter to what is crucial and defers the entry of all additional information until after the phone call. *Add Contact*, for example, allows users to add a phone number, but it does not allow entering a name for that number. Instead, the phone number is autofiled under ".blindSight filed <date><time>". The same holds for new appointments. Deferring the entry of less relevant data until after the call minimizes in-call interaction time and thus minimizes the impact on the conversation.

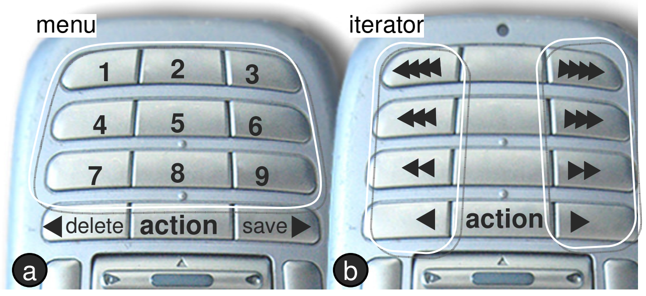

Figure 4. (a) The *menu* mapping affords the entry of a small number of choices, such as digits, characters, or menu functions. (b) The *iterator* mapping affords selection from a long or non-finite list of choices.

Walkthrough

We now revisit the scenario from the introduction section, this time using blindSight. We show blindSight interactions like this: $\boxed{\text{button pressed}}$ followed by the resulting *"audio response"*. While this presentation style suggests turn-taking between human-human and human-phone interactions, blindSight interactions typically take place *in parallel* with the spoken dialog, as discussed earlier. This often avoids wait times altogether. To convey a sense of the timing, we refer the reader to the accompanying video figure.

John: Hi Ami, this is John, can we meet sometime next week?

Ami: Oh, hi John. Yeah, sounds great. When did you have in mind?
$\boxed{\text{calendar}}$ *"calendar"*
$\boxed{\text{calendar}}$ **(enters calendar)** *"Monday 9am"*
$\boxed{\text{week +}}$ *"next Monday"*

John: How about Tuesday morning sometime?

Ami: $\boxed{\text{day +}}$ *"next Tuesday"*
$\boxed{\text{preview 3 hours}}$ *"tic, sssh, tic"* [tic="busy", sh="free"]

Ami realizes that noon is taken & looks for alternatives

$\boxed{\text{3 hours+}}$ *"noon"*
$\boxed{\text{preview 3 hours}}$ *"tic, tic, ssssh"*

I'm busy in the morning, but I am free in the early afternoon.

John: Sorry, I have meetings all afternoon. How does Wednesday afternoon look?

Ami: $\boxed{\text{day +}}$ *"Wednesday"*
$\boxed{\text{preview day}}$ *"tic,ssh,tic, ,ssssssssh"*

Yeah, Wednesday sounds good. I am free after 1.

John: Ok, let's make it one then. Call me on my mobile phone if anything comes up.

Ami: $\boxed{\text{½ hour +}}$ *"30"*
$\boxed{\text{½ hour +}}$ *"1"*
$\boxed{\text{block ½ hour++}}$ *"blocked"*

Will do, can you give me your number again?

$\boxed{\text{(hold) home}}$ **(returns home)** *"home"*
$\boxed{\text{add contact}}$ *"add contact"*

$\boxed{\text{add contact}}$ **(enters add contact)** *"enter number"*

John: Sure, do you have something to write with?

Ami: Yep!

John: It is (206)…555… 7324. Got it?

Ami: $\boxed{2}$*"2"*$\boxed{0}$*"0"*$\boxed{6}$*"6"* $\boxed{5}$*"5"*$\boxed{5}$*"5"*$\boxed{5}$*"5"*$\boxed{7}$*"7"*$\boxed{3}$*"3"*$\boxed{2}$*"2"*$\boxed{4}$*"4"*$\boxed{5}$*"5"*
$\boxed{\text{save}}$ *"saved"*
(returns home) *"home"*

Of course. Oh, and if anything comes up, call me at the AI lab, their number is…

$\boxed{\text{find contact}}$ *"find contact"*
$\boxed{\text{find contact}}$ **(enters find contact)** *"enter name"*
$\boxed{\text{2abc}}$ *"6 matches"*
$\boxed{\text{4ghi}}$ *"1 match"*
$\boxed{\text{play}}$ *"AI lab"*
$\boxed{\text{play}}$ *"4 2 …"*

John: Hold on, let me get something to write with…

Implementation
We implemented a blindSight prototype on the *Windows Smartphone 2003* platform. BlindSight is invoked automatically when placing a call or when a call is received. It then allows accessing the user's calendar and contact list information using the interactions described in this paper.

BlindSight is written in C++ and C# using the .NET Compact Framework 1.0. The prototype uses the Pocket Outlook Object Model to access the user's contact list as well as the calendar. We used pre-recorded speech for auditory feedback.

TACTILE KEYPAD IMPROVING ONE-HANDED USE
BlindSight, as described above, is a complete and functional system. Yet, to operate blindSight successfully, users need to be able to operate buttons with sufficient reliability. This means that phone hardware plays an important role.

Many skilled users can operate their phone eyes-free if the phone is in its standard position in front of the user. Unfortunately, we found that these skills do not always transfer when the phone is held by the ear.

Figure 5 shows two postures we observed. While holding a phone in front of the user allows resting the phone loosely on the fingers, holding the phone up to the ear (Figure 5a) requires index and middle fingers to impose a firm grip on the phone to hold it. Unfortunately, posture 5a causes the thumb to hit the keypad at an oblique angle, preventing users from feeling tactile features on the phone keypad. Posture 5a thus makes keypad operation error prone. A pilot study of posture 5a during which participants entered random sequences of numbers eyes-free (similar to the study described in section "User Study 1") showed error rates as high as 20% for some participants.

The problem can be alleviated partially by supporting the phone using a second hand (Figure 5b), but this may not always be possible or desirable.

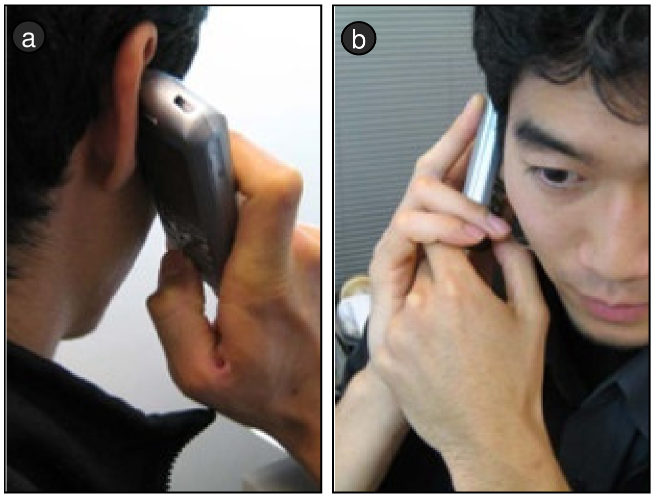

Figure 5. (a) Accessing the built-in phone keypad using one hand and (b) two hands.

To inform the design of future eyes-free phones we investigated tactile keypad features, produced several design prototypes, and conducted a series of pilot studies, as well as a user study.

Adding tactile features to the phone keypad
Figure 6a shows the Audiovox 5600 phone we started out with. In a pilot study it showed poor targeting performance; most participants reported difficulties distinguishing buttons.

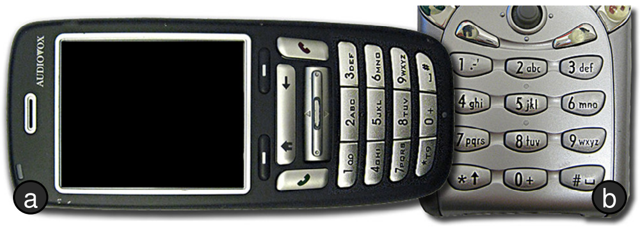

Figure 6. (a) The *Audiovox 5600* phone. (b) The *Red-E SC1100* phone offers more space between buttons.

We investigated the problem further using a series of clay prototypes (Figure 7). Larger gaps between buttons and rounded buttons seemed to address the problem (Figure 7a). We found a Smartphone that possesses these characteristics, the *Red-E SC1100* shown in Figure 6b. Since this phone is no longer commercially available, we also modified our Audiovox 5600 phone by cutting grooves between its keys (Figure 8a). The groves substantially decreased error rate for pilot participants.

However, for buttons located at center of the keypad, we still observed high targeting times and somewhat elevated error rates. During piloting, we observed that participants started all targeting from the corner positions because these were the only uniquely identifiable buttons. Users then traversed the keypad towards the desired target. This was slow and error prone. As a result the 5 and 8 keys were most troublesome to hit and users often confused them.

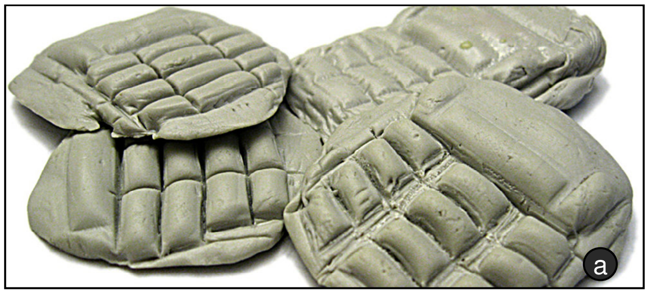

Figure 7. The clay prototypes we used to determine minimum button spacing

To address this confusion, we added tactile features to the keypad. We experimented with features *between* buttons, as already offered by some phones (Figure 6b), but even if we enlarged these features they remained all but imperceptible. We therefore added features *onto* the buttons, first on the 7-8-9 row and finally also on the 4-5-6 row (Figure 8). A final round of piloting showed that this dramatically reduced error rates and targeting time, resulting is roughly equivalent access times for buttons across the keypad.

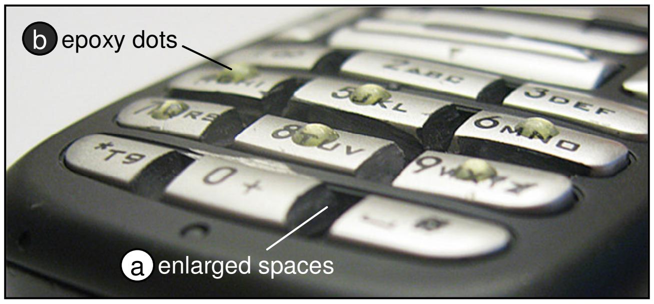

Figure 8. We modified this Audiovox 5600 keypad by (a) enlarging the gaps between buttons and (b) adding epoxy dots on buttons 4 through 9.

Flipping the phone to help users access the keypad

While the resulting keypad worked well, its operation remained cumbersome and tiring due to the odd angle of the hand as shown in Figure 5a. Rather than trying to make further improvements on targeting performance, we made one last design to improve on the ergonomics.

This form factor was inspired by how users hold the phone when talking. As shown in Figure 9, the typical grip holds the phone between the thumb on one side and little finger and ring finger on the other side. The index and middle fingers keep the phone in contact with the ear, but remain free to move around.

We considered creating a secondary keypad in this area on the back of the device, but it turned out that the design could be achieved with an existing keypad by flipping the phone around, as shown in Figure 1. We called this form factor *flipPhone*. Flipping the phone only requires replicating the speaker and microphone. This avoids problems that would likely result from a double keypad, such as unintentional button presses.

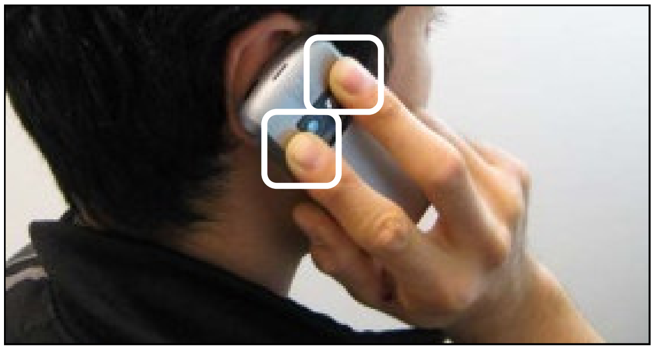

Figure 9. This typical phone holding posture places index and middle fingers on the back of the device.

Flipping the phone meant changing the mapping of the buttons to the rotated mapping already illustrated in Figure 3 and Figure 4. Relabeling the keys was not necessary, because users do not see the keypad when it is flipped. Users feel the tactile features though, which is the reason for the double row of features shown in Figure 8; this arrangement was symmetric and therefore preserved meaning when rotated.

Implications on the design of the auditory menu

Our work on phone keypads took place in parallel to our work on the auditory menu system and informed the design of the auditory menu system. Knowing that the 3x4 button numeric portion of the keypad could be made accessible led us to design for that keypad size, rather than for smaller keypad subsets we had considered earlier.

What remained were limitations on the overall size of the keypad. Holding the phone up to the ear, including the flipPhone-style grip, does impact the range of the fingers. We therefore opted to limit our designs to the 3x4 numeric portion of the phone keypad. This also preserved symmetry and thus kept the keypad layout consistent when users changed between regular and flipPhone orientation.

USER STUDY 1: PHONE OPERATION AT THE EAR

The first study examined the hardware designs presented in the previous section. The main purpose of this study was to verify that our modifications enabled users to operate the phone keypad in the ear position. In particular we wanted to verify reliability, i.e., whether error rates were in a range adequate for supporting the blindSight interaction model.

In addition, we measured task times of the eyes-free conditions, as these would eventually determine the maximum interaction speeds of blindSight. To put task times in perspective we added a *Visual* baseline condition.

Finally, we were interested in the relative performance of the two eyes-free form factors. We expected the less familiar flipped posture to require a longer learning period, but ultimately to perform better because of its two-finger use.

Interfaces

There were three interface conditions.

In the *Ear* condition, participants held the phone against their ear as shown in Figure 5a. They operated the phone keypad using the thumb of the hand holding the phone.

In the *Flip* condition, participants held the phone in the flipped position as shown in Figure 1. Participants were instructed to operate buttons with both their index finger and middle finger.

For the *Flip* and *Ear* conditions, we verified that participants kept the phone in contact with their ear at all times, which prevented them from looking at the phone screen.

In addition, we included a *Visual* condition as a baseline. In this condition, participants held the phone in front of them and operated the buttons with the thumb of the same hand. Participants were invited to look at the phone, which allowed them to visually verify targeting before pressing buttons.

All three conditions were implemented using the same *Red-E SC1100* phone shown in Figure 6b. The phone was enhanced with epoxy dots on the 4,5,6,7, 8, and 9 buttons. Participants operated the phone using their dominant hand.

Task

We measured keypad performance using a simple button pressing task. During each trial, participants entered the same 10-digit number. The number had been randomly generated once for the entire study and contained each digit from 0-9 exactly once. A sheet showing the number was kept in participants' sight throughout the study.

Correct input was acknowledged using a sound sample repeating the digit entered. If an incorrect digit was entered, an error sound was played in addition. Participants had to correct their input before proceeding. However, the correction procedure was simplified in that participants only had to re-enter the correct digit, rather than having to operate a backspace key.

To ensure participants could hear the auditory feedback also in the *Visual* condition, feedback for all three conditions was administered using a pair of speakers plugged into the headset jack of the phone.

Task time was measured on a per key basis from the beginning of the audio prompt to then moment a key was pressed.

Apparatus

The study was conducted using the *Red-E SC1100* phone shown in Figure 6b, with epoxy dots. It offered 16MB RAM and a 132MHZ Processor and ran the Microsoft Smartphone 2003 operating system.

Participants

Twelve volunteers (4 female) ranging in age from 24 to 31 years (median 26) were recruited from within our institution. Each received a lunch coupon for our cafeteria as a gratuity for their time. All participants owned a mobile phone. Only one was an experienced text messager, sending about 300 messages per month. The remaining participants reported sending less than 30 texts/month and less than a year of experience.

Experimental Design

We used a within-participants design, with presentation of *Ear, Flip, and Visual* counterbalanced across participants. Within each interface condition, participants performed 3 blocks separated by 1-minute breaks. Each block contained 30 trials, with each trial requiring them to enter the same 10-digit sequence.

To allow us to investigate first time performance and learning curve, there were *no* practice trials. To minimize sequence effects across interface conditions, each participant performed each interface condition in 3 separate sessions with 1-12 hours between sessions. Each session took about 10 minutes, resulting in an overall duration of about 30 minutes per participant.

In summary, the experimental design was: 3 *Interfaces* (*Ear*, *Flip*, and *Visual*) × 3 blocks × 10 numbers × 10 digits per number = 900 key presses per participant.

Results

Repeated measures analyses of variance were used to assess the effects of interface (*Flip* vs. *Ear* vs. *Visual*) on error rate and selection time.

Error rates: Errors rates are the number of incorrect key presses per block divided by the number of required key presses per block (100). Repeated errors were counted only once, i.e., errors correcting an error were not counted. As expected, error rates for the eyes-free conditions were higher than the *Visual* baseline; *Flip* ($F_{1,11}$=25.32, p<.05) and *Ear* ($F_{1,11}$=36.17, p<.05). The difference in error rates between *Flip* and *Ear* was not statistically significant (p>.05).

For the last block of trials, error rates were 4.33% for *Flip*, 5.33% for *Ear* and 0.33% for *Visual* (Figure 9).

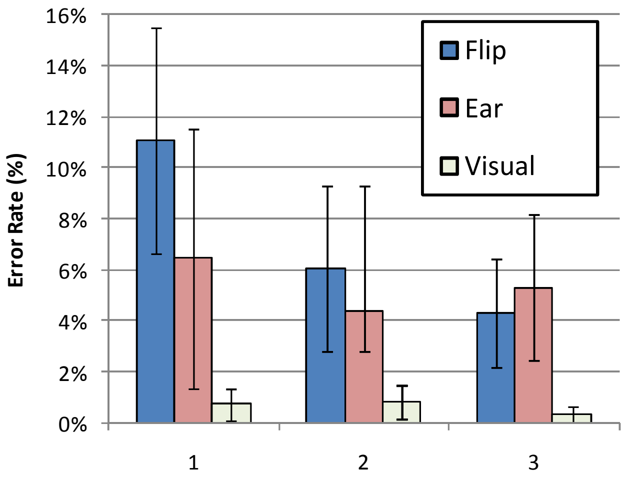

Figure 10. Error rates for the *Flip*, *Ear*, and *Visual* conditions by block number. Error bars show 95% confidence interval.

Task time: Task time for key presses was measured from when the audio prompt for that key started playing to when the key was pressed down. During aggregation of the data, medians were used as a measure of central tendency to reduce the effect of outliers [24]. For each block, we took the median of each participant's key press times as a representative measure of his/her performance for that block. We then averaged these representative measures across participants for each block.

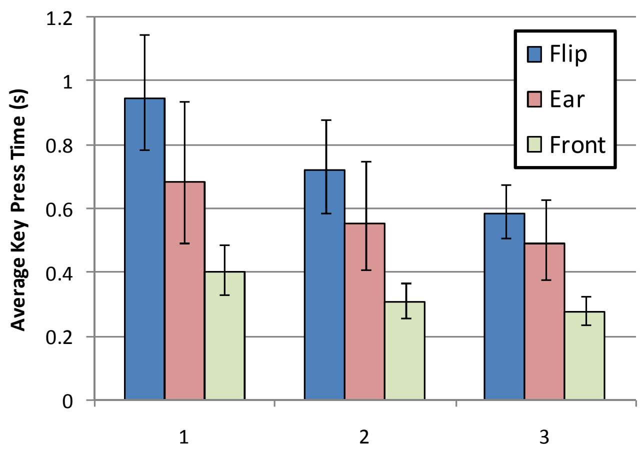

Figure 11. Average key press times for each interface condition by block. Error bars show 95% confidence intervals.

Again as expected the *Visual* baseline condition was faster than *Flip* ($F_{1,11}$=44.08, p<.001) and *Ear* ($F_{1,11}$=17.06, p=.003). Overall, *Ear* was faster than *Flip* ($F_{1,11}$=5.229, p<.05). However, in the last block, there was no significant difference in speed between *Flip* and *Ear* ($F_{1,11}$=2.66, p>.05).

Subjective Preference: When asked to compare the *Ear* and *Flip* techniques, 6 participants expressed a preference for *Ear* and 6 participants expressed a preference for *Flip*. Participants who preferred *Ear* cited the familiarity with using the thumb for number entry as the reason. Participants who preferred Flip cited its more comfortable ergonomics.

Ten of the twelve participants commented on the usefulness of the tactile features on the buttons for the two eyes-free conditions. Two participants commented on how the *Flip* condition felt similar to holding a mouse.

Discussion

With error rates around 5%, the *Ear* interface condition seems well-suited for use with blindSight. The *Flip* form factor seems promising, but its lack of familiarity made it require 200 key presses/10 minutes of practice time to reach an error level comparable to the *Ear* condition. Given the even split in preference between the two eyes-free interface conditions, however, both form factors seem worthy of further investigation. Our expectation that *Flip* would beat *Ear* in terms of task time was not fulfilled, a study with more than three blocks would be necessary to investigate this.

With respect to the visual baseline condition, the *Ear* and *Flip* conditions were about 200ms and 300ms slower per key press. This corresponds to 2-3 seconds for entering a phone number. In the context of blindSight, this seems like an acceptable cost, especially given that these numbers were obtained with users experienced with the visual control conditions, but new to the eyes-free conditions.

USER STUDY 2: BLINDSIGHT VS. SMARTPHONE

In this final study, we compared the eyes-free blindSight system with *Windows Smartphone 2003* as the visual baseline. During the study, participants scheduled calendar appointments and added contacts while engaged in a phone conversation with an experimenter. Participants performed these two tasks under two different levels of distraction.

One of the key hypotheses driving this system is that it is possible to overload the auditory channel with feedback even though that channel is already in use for human-human communication. We considered a range of more formal study designs, but all of them required us to decompose the system into techniques (e.g., studying the menu system). Such a quantitative study can provide valuable insights, but the qualitative study we employed allowed us to use an ecologically valid design with an actual conversation partner, a real world task, and a control condition that is not only visual, but also a complete commercial system. A study with these parameters would have been all but impossible had the goal been to obtain quantitative data.

Interfaces and keypad conditions

Participants were placed in a separate room and communicated to an experimenter over the phone running the respective interface. The phone was an *Audiovox 5600* mobile phone running Windows Smartphone 2003, with the epoxy dots shown in Figure 6a. Participants controlled the phone one-handed using their dominant hand.

In the *Smartphone* condition participants used the contact list and calendar functions that are part of the original Smartphone software. The phone allowed participants to launch the calendar and the contact list using two-key sequences. To add a contact, participants hit an “add” key, scrolled down to the phone number field and keyed in the number. To add an appointment, participants selected “add appointment” from a menu and filled in start and end times in an onscreen dialog. To operate these functions, participants looked at the screen of the device.

In the *blindSight* condition, the phone ran the blindSight prototype software described earlier which implemented the menu system shown in Figure 3. Unlike the Smartphone condition, participants navigated this interface eyes-free by means of auditory feedback. Half of the participants interacted with blindSight using the *Ear* posture while the other half used the *Flip* posture, each one identical to the respective interface in the previous study.

Due to a hardware bug at the day of the study, our prototype failed to run phone conversations while running blindSight. We created a work-around by injecting audio from a second phone routing the call into the speaker of the phone running blindSight.

Tasks

There were two tasks, both of which were administered by the experimenter who folded them into the phone conversation. During the conversation, the experimenter either gave participants a phone number to be recorded or negotiated an appointment with them. For the scheduling task, the experimenter proposed a day and time (morning, evening, etc.). Participants checked the proposed time against the pre-populated calendar on the phone, negotiated an alternative time slot in case of conflict, and entered the appointment.

Distraction

In the *Idle* condition, there were no additional stimuli.

In the *Driving* condition, participants performed the tasks and maintained the call while controlling an interactive driving game (*Moorhuhn Kart 2*). The game was operated with one hand using the four cursor keys. This condition allowed us to compare blindSight and Smartphone usage while involved in a cognitively loaded task such as driving.

To reflect the way many users operate phones while driving, we allowed participants to use a headset during the *Driving + Smartphone* condition. Six of our eight participants made use of this option.

Procedure

Participants received 10 minutes of training per interface condition. Participants were provided with printouts of the relevant parts of the menu structures of both interfaces and kept in sight throughout the study.

Participants performed 5 *Schedule Meeting* trials interlaced with 4 *Add Contact* trials with one interface × distraction condition. Then they repeated the block with new data on the remaining three interface × distraction conditions. The presentation order was counterbalanced.

Participants filled in a questionnaire and were interviewed regarding their experience. The study lasted approximately 60 minutes per participant.

Participants

We recruited 8 volunteers (2 female). Four participants had owned a Windows Smartphone for at least one year. Six participants reported talking while driving at least three times a week. Three participants reported using either a speakerphone or headset when on the phone while driving.

Hypotheses

Our main hypothesis was that we would see a subjective preference for blindSight. We expected to see a stronger preference for the driving condition because the competing visual task would interfere more with the visual Smartphone condition than with the eyes-free blindSight condition.

Results

All participants completed both tasks successfully for all conditions. Figure 12 shows the tallied responses of the seven questions that required participants to choose between interfaces; Figure 13 shows the results of the Likert scale questions referring to the blindSight condition.

Six of eight participants reported an overall preference for the blindSight interface over the Smartphone interface (Figure 12). Seven preferred blindSight for the driving condition, supporting our hypothesis. An experienced Smartphone user for over 5 years exclaimed *"If there were something like [blindSight], I would totally use it."*

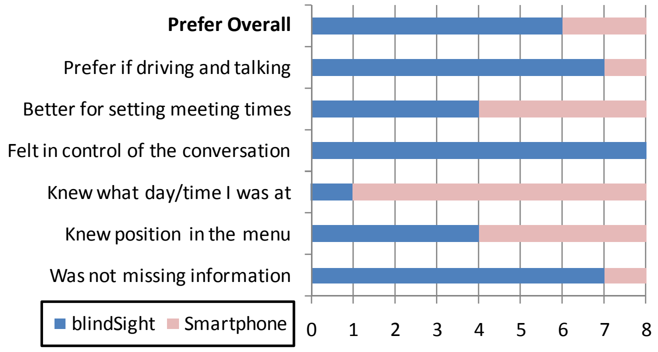

Figure 12. The number of participants (n=8) who preferred the blindSight/Smartphone conditions in the context of the respective statement.

The questionnaire results suggest explanations for this preference. While the functional parts of the systems receive balanced preference scores (blocking out meetings, menu orientation), the determining factor seemed to be that blindSight made participants feel in control of the conversation (7 out of 8 participants) and prevented them from missing information (7 out of 8). Participants found it useful to be able to hear content without having to move the phone away from their ear (Figure 13) and rated blindSight's eyes-free use as "very useful" (6.13/7).

Two people preferred Smartphone over blindSight. One explained *"I like having something to look at,"* but mentioned that if she were driving, she would prefer to use blindSight for safety reasons. The other participant who preferred using Smartphone had difficulties hitting the buttons eyes-free. He also expressed no preference for blindSight in a driving scenario; he talks about 20min a day while driving using his current (visual) phone.

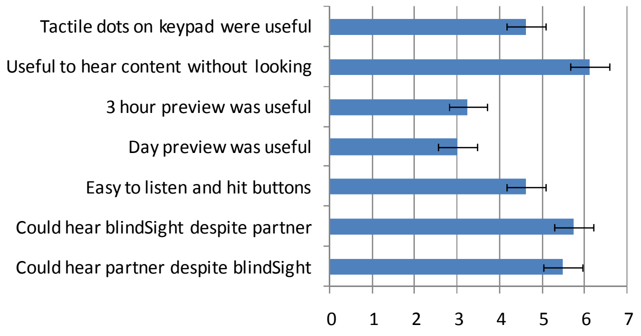

Figure 13. Likert responses (7-point scale) for questions regarding the eyes-free condition. Higher ratings are better. Error bars represent standard error of the mean (n=8).

The questionnaire identified calendar navigation as a weakness in the tested version of blindSight, giving blindSight low preference score for "I knew what date/time I was at". Participants expressed that they would have preferred blindSight to repeat the current date and time position more often. The non-speech calendar previews received mixed reviews. While participants liked the idea per se, five participants said they were unable to differentiate between the "free" and "busy" blips. One reported trouble parsing the blips into time segments. Three participants suggested replacing the non-speech previews with a spoken list of busy or free slots. All participants managed to use the iterative calendar exploration allowing them to succeed at the task. One participant commented "*It's more foolproof that way.*"

Apart from that, participants expressed enjoyment using the blindSight condition and found the system easy to use. One participant mentioned "*If I were using it eyes free [while driving] I wouldn't hold it by my ear—I would use it with a headset. I don't mind the manipulation so much as I mind the need to look at [the phone].*" This suggests that blindSight and the use of headsets are *complementary* and should not be considered competing approaches.

CONCLUSION

In this paper we presented blindSight, a system that enables eyes-free navigation by providing auditory feedback.

We made 3 main contributions. First we presented the design and implementation of a system that enables eyes-free access to phone content. Second, we investigated phone keypad interaction at the ear, presented several design improvements, and validated our designs using a user study. Third, we presented results from a user study comparing blindSight to a visual baseline condition that finds subjective preference for blindSight.

As future work we plan to explore the applicability of our findings to solutions for visually impaired users.

ACKNOWLEDGEMENTS

We thank members of the ASI group at Microsoft Research with special thanks to Ed Cutrell and David Thiel.

REFERENCES

1. microsoft.com/windowsmobile/voicecommand/default.mspx
2. www.tellme.com.
3. Aucella, A. F. and Ehrlich, S. F. Voice messaging enhancing the user interface design based on field performance. *Proc. CHI '86,* pp. 156–161.
4. Baudisch, P. Don't Click, Paint! Using Toggle Maps to Manipulate Sets of Toggle Switches. *Proc. UIST '98,* pp. 65–66.
5. Blanchard, H.E. and Lewis, S.H. (1999). Voice messaging user interface. In D. Gardner-Bonneau (Ed.), *Human Factors and Voice Interactive Systems*, Kluwer, pp. 257–284.
6. Brewster, S. (1998). Using nonspeech sounds to provide navigation cues. *ACM TOCHI* 5(3). pp. 224–259.
7. Dietz, P. H. and Yerazunis, W. S. Real-time audio buffering for telephone applications. *Proc. UIST'01, pp.* 193–194.
8. Fukumoto, M. and Tonomura, Y. Whisper: a wristwatch style wearable handset. *Proc. CHI '99*, pp. 112–119.
9. Gaver, W. The Sonic Finder: An interface that uses auditory icons. *Human Computer Interaction,* 4(1). pp. 67–94.
10. Goldberg, D., and Richardson, C., Touch-typing with a stylus. *Proc. INTERCHI'93,* pp. 80–87.
11. González, I. E., Wobbrock, J. O., Chau, D. H., Faulring, A., and Myers, B. A. Eyes on the road, hands on the wheel: thumb-based interaction techniques for input on steering wheels. *Proc. GI'07,* pp. 95–102.
12. Hansen, J.H.L., Plucienkowski, J., Gallant, S., Pellom, B.L., Ward, B. CU-Move: Robust Speech Processing for In-Vehicle Speech Systems. *Proc. ICSLP'00*, pp. 524–527.
13. Hiraoka, S., Miyamoto, I., Tomimatsu, K. (2003) Behind Touch, a Text Input Method for Mobile Phones by The Back and Tactile Sense Interface. *Information Processing Society of Japan, Interaction'03*. p. 131–138.
14. Hornstein, T. 1994. Telephone voice interfaces on the cheap. *UBILAB Rep,* Union Bank of Switzerland, Zurich.
15. Hudson, S. & I. Smith. Electronic Mail Previews Using Non-Speech Audio. *CHI'96 Extended Abstracts*, 237-238.
16. Ito, Mizuko, et. al., eds. Personal, Portable, Pedestrian. *Mobile Phones in Japanese Life*, MIT Press, Cambridge, Mass.
17. Luk, J., Pasquero, J., Little, J., MacLean, K., Levesque, V., Hayward, V. A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. *Proc. CHI'06*, pp. 171–180.
18. Lyons, K., Skeels, C., Starner, T., Snoeck, C.M., Wong, B.A., Ashbrook, D. Augmenting conversations using dual–purpose speech. *Proc. UIST'04,* pp.237–246.
19. Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., and Looney, E. W. Twiddler typing: one-handed chording text entry for mobile phones.*Proc.CHI'04,*671-678.
20. MacKenzie, S. Mobile text entry using three keys. *Proc. NordiCHI'02, pp*. 27–34.
21. Marics, M., and Engelbeck, G. (1997). Designing voice menu applications for telephones, in Handbook of Human-Computer Interaction, Helander, M., Landauer, T., and Prabhu, P. Editors. Elsevier, pp. 1085–1102.
22. Perugini, S., Anderson, T. J., and Moroney, W. F. A study of out-of-turn interaction in menu-based, IVR, voicemail systems. *Proc. CHI'07, pp*. 961–970.
23. Resnick, P. and Virzi, R. A. Skip and scan: cleaning up telephone interface. *Proc. CHI 1992*, pp. 419–426.
24. Rosenberger, J., and Gasco, M. (1983). Comparing location estimators: Trimmed means, medians, and trimean. in Understanding robust and exploratory data analysis, Hoagan, D., Mosteller, F., and Tukey, J., Editors. John Wiley: New York.
25. Starner, T. E., Snoeck, C. M., Wong, B. A., and McGuire, R. M. Use of mobile appointment scheduling devices. *CHI'04 Extended Abstracts,* pp.1501–1504.
26. Sugimoto, M. Hiroki, K. HybridTouch: an intuitive manipulation technique for PDAs using their front and rear surfaces. *Proc. MobileHCI'06, pp*. 137–140.
27. Teslar, L. The Smalltalk Environment, BYTE, 6,'81, 90-147.
28. Tucker, S., and Whittaker, S. Time is of the Essence: An Evaluation of Temporal Compression Algorithms. *Proc. CHI'06*, pp 329-338.
29. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., Shen, Chia. LucidTouch: A See-Through Mobile Device. *Proc. UIST'07,* pp.269-278.
30. Wobbrock, J. O., Chau, D. H., and Myers, B. A. An alternative to push, press, and tap-tap-tap: gesturing on an isometric joystick for mobile phone text entry. *Proc.CHI'07,* 667–676.
31. Wobbrock, J. O., Myers, B. A., and Kembel, J. A. Edge-Write: a stylus-based text entry method designed for high accuracy and stability of motion. *Proc. UIST'03,* pp. 61–70.
32. Yin, M, Zhai, S. The Benefits of Augmenting Telephone Voice Menu Navigation with Visual Browsing and Search. *Proc. CHI'06*, 319–328
33. Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. Earpod: eyes-free menu selection using touch input and reactive audio feedback. *Proc. CHI'07*, pp. 1395–1404.