Title
Enabling eyes-free interaction with tactile messages based on human experience

Permalink
https://escholarship.org/uc/item/26s131zr

Author
Li, Kevin Ansia

Publication Date
2009

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Enabling Eyes-free Interaction with Tactile Messages Based on Human Experience

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science (Computer Engineering)

by

Kevin Ansia Li

Committee in charge:

Professor William G. Griswold, Chair
Professor James D. Hollan, Co-Chair
Professor Patrick Baudisch
Professor Barry Brown
Professor Adriene Jenik
Professor Lawrence Saul

2009

The dissertation of Kevin Ansia Li is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California, San Diego

2009

DEDICATION

To my parents, Chin-Ho and Pone-Jane, and to my brother Allen.

TABLE OF CONTENTS

Signature Page iii

Dedication iv

Table of Contents v

List of Figures ix

List of Tables xii

Acknowledgements xiii

Vita and Publications xv

Abstract of the Dissertation xvi

Chapter 1 Introduction 1
- 1.1 The Need for Eyes-free Interaction 2
- 1.2 A Potential Solution: Auditory Feedback 2
- 1.3 Related Work 4
  - 1.3.1 Human Perception and Information Transmission via the Skin 5
  - 1.3.2 Virtual Reality and Surgical Simulations 6
  - 1.3.3 Tactile Feedback in Mobile Devices 7
  - 1.3.4 Tactile Communication 7
- 1.4 Creating Tactile Messages Based on Human Experience . . . 8
- 1.5 Overview 10

Chapter 2 Enabling Eyes-free Interaction with Auditory Feedback: A Motivating Example 11
- 2.1 Motivation 11
- 2.2 Related Work 13
  - 2.2.1 Auditory Feedback 13
  - 2.2.2 Mobile Input 14
- 2.3 Survey of Mobile Users 15
  - 2.3.1 Results 15
- 2.4 Blindsight’s Auditory Eyes-free Interaction 16
  - 2.4.1 Design Principles 17
  - 2.4.2 Menu Organization 18
  - 2.4.3 Walkthrough 20
  - 2.4.4 Implementation 23
- 2.5 Tactile Keypad Improving One-handed Use 23

2.6 Adding Tactile Features to the Phone Keypad 24
  2.6.1 Flipping the Phone to Help Users Access the Keypad 26
  2.6.2 Implications for the Design of the Auditory Menu . . 27
2.7 User Study 1: Phone Operation at the Ear 27
2.8 Interfaces 28
  2.8.1 Task 29
  2.8.2 Apparatus 29
  2.8.3 Participants 29
  2.8.4 Experimental Design 30
  2.8.5 Results 30
  2.8.6 Discussion 32
2.9 User Study 2: Blindsight vs. Smartphone 33
  2.9.1 Interfaces and Keypad Conditions 33
  2.9.2 Tasks 34
  2.9.3 Distraction 34
  2.9.4 Procedure 35
  2.9.5 Participants 35
  2.9.6 Hypotheses 35
  2.9.7 Results 35
2.10 Properties of Auditory Feedback 38
2.11 Acknowledgments 38

Chapter 3 Mapping Music and Sounds to Vibrations to Enable Buddy Proximity Cues on Mobile Phones 39
3.1 Motivation 40
3.2 Related Work 42
  3.2.1 Proximity Detection in the Wild 43
  3.2.2 Mobile Peripheral Cue Systems 44
  3.2.3 Auditory and Tactile Cues 44
3.3 Proximity Detection 45
  3.3.1 Proximity Detection Algorithm 46
  3.3.2 Evaluating Cell Tower Ratio Algorithm for Proximity Detection 47
  3.3.3 Sensor Noise 53
  3.3.4 Minimizing Power Consumption 55
3.4 Peripheral Cues in the Wild 56
  3.4.1 Auditory Cues 57
  3.4.2 Vibrotactile Cues 58
  3.4.3 Mapping Sounds to Vibrotactile Patterns 60
3.5 PeopleTones 64
3.6 User Study 66
3.7 Participants 66
  3.7.1 Methodology 67

3.8 Usage and Self-reported Data 68
3.9 Discussion 69
  3.9.1 Peripheral Cues are a Viable Communication Mechanism in the Wild 69
  3.9.2 Mobile Phones Are a Viable Platform for Context-Aware Peripheral Cue Applications 74
3.10 Conclusion 75
3.11 Acknowledgments 77

Chapter 4 Tapping and Rubbing: An Exploration in Creating Computer Mediated Human Touch 78
4.1 Introduction 79
4.2 Related Work 80
  4.2.1 Force Feedback 80
  4.2.2 Haptics in HCI 81
  4.2.3 Applications of Hard Drive Actuator Technology . . 82
4.3 SoundTouch 82
  4.3.1 Features 85
4.4 Tapping and Rubbing 86
  4.4.1 Tapping 86
  4.4.2 Rubbing 88
4.5 Scenarios 91
4.6 User Studies 92
4.7 Study 1: User Perceptions of Tapping 92
  4.7.1 Apparatus 92
  4.7.2 Independent Variables 93
  4.7.3 Task 1: Distinguish 93
  4.7.4 Task 2: Identify 94
  4.7.5 Questionnaire 95
  4.7.6 Participants 95
  4.7.7 Results: Distinguish Task 95
  4.7.8 Results: Identify Task 97
  4.7.9 Results of the Questionnaire 99
4.8 Study 2: User Perceptions of Rubbing 100
  4.8.1 Results: Distinguish Task 100
  4.8.2 Results: Identify Task 101
  4.8.3 Results of the Questionnaire 102
4.9 Discussion and Design Implications 103
  4.9.1 Design of Notifications 103
  4.9.2 Distinguishing and Identifying Taps and Rubs 104
  4.9.3 Locations for Tapping or Rubbing Apparatuses . . . 105
4.10 Conclusions and Future Work 105
4.11 Acknowledgments 106

Chapter 5 Bridging the Gap Between Semantics and Touch: Towards a Vibrotactile Encoding of Speech . . . 107
5.1 Introduction . . . 107
5.2 Related Works . . . 109
5.2.1 Vibrotactile Communication . . . 109
5.2.2 Linguistics . . . 110
5.3 What Elements of Linguistics are Relevant? . . . 111
5.3.1 Apparatus . . . 111
5.3.2 Task and Stimuli . . . 111
5.3.3 Participants . . . 112
5.3.4 Procedure and Design . . . 112
5.3.5 Discussion . . . 112
5.4 Vibe Messaging: A Tactile Messaging Backchannel Application . . . 114
5.4.1 Initial Feedback . . . 115
5.5 Future Work . . . 115
5.6 Conclusion . . . 115

Chapter 6 New Directions for Research on Tactile Communication . . . 117
6.1 Creation of New Tactile Languages . . . 118
6.2 Emotive Aspects of Computer Mediated Tactile Communication . . . 118
6.3 The Role of Tactile Communication in a Multimodal Communication Device . . . 120
6.4 Engineering New Input and Output Devices . . . 120

Chapter 7 Final Thoughts . . . 121
7.1 Methodology . . . 121
7.1.1 Surveys . . . 122
7.1.2 Quantitative Measurements . . . 122
7.1.3 Qualitative Measurements . . . 123
7.2 Contributions . . . 123
7.3 Outlook on Tactile Communication Research . . . 125

Bibliography . . . 126

LIST OF FIGURES

Figure 1.1: Taxonomy we used to guide our exploration of mapping stimuli that people are familiar with to the tactile channel. 9

Figure 2.1: To access phone information stored on the phone mid-conversation, users press buttons and receive auditory confirmation. This photo shows the *flipPhone* form factor. 12
Figure 2.2: Number of participants out of nine who rated the respective feature as “would like” or “very important” 16
Figure 2.3: BlindSight’s menus 19
Figure 2.4: (a) The menu mapping and (b) the iterator mapping. 20
Figure 2.5: Walkthrough of blindSight usage scenario. 22
Figure 2.6: (a) Accessing the built-in phone keypad using one hand and (b) two hands. 24
Figure 2.7: (a) The *Audiovox 5600* phone. (b) The *Red-E SC1100* phone offers more space between buttons. 24
Figure 2.8: The clay prototypes we used to determine minimum button spacing. 25
Figure 2.9: We modified this *Audiovox 5600* keypad by (a) enlarging the gaps between buttons and (b) adding epoxy dots on buttons 4 through 9. . 26
Figure 2.10: This typical phone holding posture places index and middle fingers on the back of the device. 27
Figure 2.11: Error rates for the Flip, Ear, and Visual conditions by block number. Error bars show 95% confidence intervals. 31
Figure 2.12: Average key press times for each interface condition by block. Error bars show 95% confidence intervals. 32
Figure 2.13: The number of participants (n=8) who preferred the blindSight or Smartphone conditions in the context of the respective statement. . 36
Figure 2.14: 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). 37

Figure 3.1: Equation used for calculating proximity ratio for two mobile phones where *a* and *b* are the sets of cell towers seen by each phone. 47
Figure 3.2: Initial cell ratio measurements taken from 3 different areas of varying population density. 47
Figure 3.3: Precision for different “nearby” distances in *Suburb*. 50
Figure 3.4: Precision for different “nearby” distances in *Downtown*. 51
Figure 3.5: Precision at different distances in *Suburb* for the entire data set. . . 52
Figure 3.6: Precision at different distances in *Downtown* for the entire data set. 52
Figure 3.7: Two-bit counter for eliminating noise in proximity detection. . . . 54
Figure 3.8: Code for generating a 20ms vibrotactile pulse. 59

Figure 3.9: Oscilloscope plots of voltage generated for vibrotactile actuator for pulses at levels (a) 1 (b) 3 (c) 5 and (d) 10. 61
Figure 3.10: Block diagram showing the process of converting a WAV file to a vibrotactile pattern. 63
Figure 3.11: (a) The main PeopleTones user interface showing buddies that are near and far, as well as the last time the system was updated. (b) Post-alert prompt asking participants how useful the alert was. . . . 65

Figure 4.1: Rubbing interaction implemented using our soundTouch prototype. 79
Figure 4.2: Our *soundTouch* prototype translates sound signals into tactile feedback. 83
Figure 4.3: Amplifier circuit schematic. 83
Figure 4.4: Voltage Divider to create $+6$V and -6V power rails for two amplifiers. 84
Figure 4.5: Voltage waveform produced by soundTouch to generate a tap with amplitude of level 7. 84
Figure 4.6: (a) Closeup of the voice coil motor in soundTouch (b) micro drive next to a US quarter. 85
Figure 4.7: Side view of a voice coil motor. 86
Figure 4.8: Tapping prototype: a hammer attached to the head of the soundTouch prototype taps on the user's finger. 87
Figure 4.9: Some of the materials we have used as hammer heads. 88
Figure 4.10: Rubbing prototype (cover removed). 89
Figure 4.11: (a) Rubbing prototype with cover; only the head shows through (b) earlier prototype. 90
Figure 4.12: (a) Bare head and (b) covered in Teflon tape. 90
Figure 4.13: Mean values of user reported levels for the *Identify* task for the *Amplitude* stimulus condition. Error bars show 95% confidence intervals. 98
Figure 4.14: Mean values of user reported levels for the *Identify* task for the *ConstantNumber* condition. Error bars show 95% confidence intervals. 98
Figure 4.15: Mean values of user reported levels for the Identify task for the *ConstantDuration* condition. Error bars show 95% confidence intervals. 99
Figure 4.16: Mean values of user reported levels for the *Identify* task for the *ConstantDuration* condition. Error bars show 95% confidence intervals. 101
Figure 4.17: Mean values of user reported levels for the *Identify* task for the *ConstantRubs* condition. Error bars show 95% confidence intervals. 102

Figure 5.1: Plots of the vibrotactile patterns that most participants associated with the different text phrases. 113

Figure 6.1: Past work has focused on how people *receive* messages, not how they can be *sent*. 118
Figure 6.2: Prototype for enabling end-user generated tapping and rubbing. . . . 119

LIST OF TABLES

Table 3.1: Recall for different ratios with a distance threshold of 0.1km when phones are at the same location. Precision is 99.9% since the subset falls within 0.1km. 49
Table 3.2: Recall rates at distances 0.1km to 1.0km for different ratios when phones are nearby. 50
Table 3.3: Recall rates for different ratios in the entire dataset. 53
Table 3.4: Group makeup for the three groups. 66
Table 3.5: Self-reported response to the cue. 68
Table 3.6: Self-reported identification of the cue’s information. 69

Table 4.1: Stimulus pairs differed by one or two levels. The resulting pairs are marked with an ×. Entries shaded in gray represent pairs that differed by two levels. Columns and rows denote level of first and second tap. 94
Table 4.2: Tapping *AmplitudeDistinguish* error rate in % collapsed across all participants for *Differentiate* task when being presented a stimulus pair of intensity (<row>, <column>). 96
Table 4.3: Tapping *ConstantDuration Distinguish* error rate in % collapsed across all participants. 96
Table 4.4: Tapping *ConstantNumber Distinguish* error rate in % collapsed across all participants. 97
Table 4.5: Rubbing *ConstantDuration Distinguish* error rate in % collapsed across all participants. 100
Table 4.6: Rubbing *ConstantNumber Distinguish* error rate in % collapsed across all participants. 101

ACKNOWLEDGEMENTS

To my Mom and Dad for always encouraging me to think like a scientist and for attempting to ground me in reality. To Allen for always listening to my ridiculous ideas.

To Jen for always believing in me even when I didn't believe in myself. And for bringing me snacks.

To my officemates, past and present, for bouncing ideas around, putting up with solder fumes and making 3150 quite possibly the best office to work in ever. To the folks in CSE for making grad school so much more than just doing research.

To my advisor Bill Griswold for teaching me to be optimistic and letting me work on research topics I knew nothing about. To my other advisor Jim Hollan for teaching me that at the end of the day, I should just do it my way. To Patrick Baudisch for pithy bits of wisdom that somehow stay etched in my mind.

To the other members of my committee: Barry Brown, Adriene Jenik and Lawrence Saul for their valuable feedback from countless discussions and for sitting through all of my horrible talks.

And last but not least, to Marti Hearst for introducing me to research and Human-Computer Interaction. None of this would have happened without you.

Chapter 2, in part, is a reprint of the material as it appears in BlindSight: Eyes-free Access to Mobile Phones. Li, K.A., Baudisch, P., and Hinckley, K. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2008), pp. 1389–1398. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in PeopleTones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones. Li, K.A., Sohn, T.Y., Huang, S., and Griswold, W.G. In Proceedings of the International Conference on Mobile Systems, Applications and Services (Mobisys 2008), pp. 160–173. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in Li, K.A., Baudisch, P., Griswold, W.G., and Hollan, J.D. Tapping and Rubbing: Exploring New Dimensions of Tactile Feedback with Voice Coil Motors. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp.181–190. The dissertation author was the primary investigator and author of this paper.

VITA

2004	B. S. in Electrical Engineering and Computer Science, University of California, Berkeley
2007	M. S. in Computer Science, University of California, San Diego
2009	Ph. D. in Computer Engineering, University of California, San Diego

PUBLICATIONS

Li, K.A., Baudisch, P., Griswold, W.G., and Hollan, J.D. Tapping and Rubbing: Exploring New Dimensions of Tactile Feedback with Voice Coil Motors. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2008), pp. 181–190.

Li, K.A., Sohn, T.Y., Huang, S., and Griswold, W.G. PeopleTones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones. In Proceedings of the International Conference on Mobile Systems, Applications and Services (Mobisys 2008), pp. 160–173.

Li, K.A., Baudisch, P., and Hinckley, K. BlindSight: Eyes-free Access to Mobile Phones. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2008), pp. 1389–1398.

ABSTRACT OF THE DISSERTATION

Enabling Eyes-free Interaction with Tactile Messages Based on Human Experience

by

Kevin Ansia Li
Doctor of Philosophy in Computer Science (Computer Engineering)

University of California San Diego, 2009

Professor William G. Griswold, Chair

Professor James D. Hollan, Co-Chair

As computing moves towards mobile devices, new challenges emerge for Human-Computer Interaction. Although mobile phones have typically had visual interfaces, there are an increasing number of scenarios where users need to interact with their devices but cannot look at them due to situational factors. Furthermore, the visual and audio senses have already been overloaded by traditional user interface design.

Haptic feedback is a promising alternative for information delivery. Research in this domain typically takes an information theoretic approach towards increasing the bandwidth of information transfer through the skin. This approach often results in complex tactile patterns that can be difficult to learn. With the proliferation of mobile devices, and the shortcomings of visual and auditory channels of communication, there is tremendous opportunity for a tactile communication medium.

This dissertation breaks away from the traditional approach to haptic research, instead focusing on how human experience can be used to generatetactile messages that have pre-learned meaning. We have looked at how three different types of stimuli can be mapped to the tactile space: music, human touch, and speech. This set of projects acts as a proof of concept, demonstrating how the approach can be applied to a variety of different stimuli.

Chapter 1

Introduction

User interface research has traditionally focused on the desktop environment. The traditional desktop scenario assumes a user, fully attentive to a task, sitting in front of a desktop computer. Typically, a large display is available for viewing information. Users provide input using a keyboard and mouse, dedicated for that purpose. This setup has proven invaluable for knowledge workers who often have to examine large amounts of information. Advances in display technology have made information visualization more compelling than ever before. However, an inherent assumption in this setup is that the screen is always available to the user. As a result, traditional user interface design has focused on visual feedback as the main channel through which information is conveyed to the user.

Because traditional user interface design has informed the fundamental guidelines for user interface research, many of the design methodologies have carried over into mobile interface design. This is reflected by the highly visual nature of current mobile phone interfaces. When placed in a traditional desktop usage scenario, these phones reflect miniature computers with their visually appealing interfaces. However, mobile phone users often find themselves trying to access mobile phone content, while on-the-go, rather than when situated in an office environment. Although mobile phones were originally designed to support phone calls, they have become more advanced, providing access to an increasing amount of content. This stems from the fact that improvements in technology allow users to access information on their mobile phones in scenarios where it was previously not possible. Information that has traditionally been unavailable to users outside of the office is now available to users wherever they go. While this is a far cry from Mark Weiser's Ubiquitous Computing vision of computational power seamlessly embedded into the environment [Wei99], mobile phones have become the *de facto* standard for mobile computing today.

As technology advances, this disconnect between the needs of users and what mobile interface designers provide is only becoming greater. Tomorrow's user will be even more mobile. With advances in technology come new usage scenarios and users will need new ways of interacting with their devices.

1.1 The Need for Eyes-free Interaction

There are a number of scenarios where visual interaction with a mobile phone is undesirable. It is unsafe (and illegal in many states) to interact with the visual interface of mobile phones while driving. There are other situations where it can be disruptive to look at a mobile phone such as in the middle of a meeting. At other times, actively using a device can be considered socially unacceptable. Furthermore, there is an entire class of visually impaired users that cannot even see interfaces, much less use them. Additionally, as devices get smaller, many device manufacturers are removing the display entirely, resulting in an even smaller form factor. The most recent version of the Apple iPod Shuffle is an example of a screen-less portable MP3 player. These screen-less devices will require new ways of interaction that do not require the visual channel. There is a clear need to enable eyes-free interaction for scenarios where people cannot look at their devices.

1.2 A Potential Solution: Auditory Feedback

One potential approach to enabling eyes-free interaction is to use auditory feedback. Auditory feedback is easy to interpret; a system can simply read information to a user, allowing him to use it eyes-free. To explore the usage of auditory feedback for eyes-free interaction, we built a mobile application called *blindSight* that replaces the in-call menu of a mobile phone [LBH08]. Users provide input to the phone via the keypad, without looking at the screen. BlindSight responds with auditory feedback providing users with access to content stored on their mobile phones such as calendars and contact lists. This allows users to interact with their phone, without having to look at the screen. This is particularly useful for looking up information stored on the phone during phone conversations. A comparative user study showed that when users were involved in another task, the eyes-free interaction enabled by blindSight was preferred over the traditional visual interface of mobile devices. However, auditory feedback still has a number of drawbacks. Manipulating a device can be unsafe when driving or socially unacceptable when in a meeting. Furthermore, audio cannot always be heard because of environmental factors such as a busy road, or at a loud concert.

What if the pre-learned quality of auditory feedback could somehow be translated into the tactile channel?

At the most basic level, tactile messages conveying binary information are easy to interpret. Most mobile phone users of today are familiar with a vibrating mobile phone. Typically, this conveys to the user that someone is trying to contact them, but cannot provide additional information. Brown et. al explored how more complex messages could be conveyed using multiple vibrotactile actuators mounted on a user's arm [BBP06]. Geldard demonstrated how a vibrotactile mapping of written English (similar to Morse code) could be created by using five vibrotactile actuators mounted on a user's chest. He found that after 65 hours of training, users could receive messages at 38 words per minute with 90% accuracy. This is a valuable result because it demonstrates the high transmission capabilities of the skin as a channel for conveying information. This builds on the results from human perception and serves as a practical example of how the skin can be used as an information transmission channel.

These two projects exemplify a common approach in this line of research that focuses on creating a maximal number of distinguishable and identifiable tactile patterns, using skin as a transmission channel. The fundamental problem with these approaches is that they first focus on creating a maximal number of distinguishable tactile patterns, and then independently, try to associate semantics to these different patterns. When this type of approach is taken, researchers are essentially starting from a blank slate, making no assumptions about the user's experience. By ignoring human experience, users are forced to go through a steep learning curve when trying to associate semantics with the arbitrarily generated tactile patterns.

One of the main problems that has prevented tactile feedback from achieving widespread adoption has to do with bootstrapping. Tactile feedback would be desirable for users in a number of scenarios, but users lack the time or desire to spend hours training themselves on these tactile feedback mechanisms. What we need is something to bridge the gap between the theoretical limits of tactile feedback and what we have available to us today, namely a vibrating phone. This thesis hopes to address this problem by filling this much needed gap.

1.3 Related Work

To place this thesis in the context of the work that has been done by the haptic research community, this section provides a brief description of a number of key areas in this space.

Historically, haptic research has been dominated by psychologists and robotics and virtual reality research. Psychology researchers have been interested in studying the limitations of the skin from a human perception standpoint. In robotics and virtual reality scenarios, haptics has been particularly useful for creating tangible feeling for the user to make simulations seem more realistic [LTCK03]. This has been particularly useful for telemanipulation, allowing doctors to manipulate surgical devices remotely with precision. In virtual reality it has been particularly useful for helping the user become more immersed in their environment.

Because of the many benefits of haptics for Human-Computer Interaction, there has been a large body of work done in this space. A number of technologies have been proposed to enable tactile feedback, ranging from various types of motors to fabricated materials such as shape memory alloys [DMSW90] and piezoelectric tabs[LPL$^+$06]. Hayward provides a good overview of the different technologies used for enabling tactile feedback [HM07]. In this section, we focus instead on the different applications that haptic technologies have been proposed for. While the following categories are hardly exhaustive, they are intended to provide some structure to the incredibly large space of tactile feedback so that the research forming this dissertation can be placed in context with respect to what has already been done. The applications of haptic research can be thought of as focusing on increasing bandwidth, providing tactile feedback for surgical simulations, enabling tactile feedback in mobile devices, and providing a mechanism for tactile communication.

1.3.1 Human Perception and Information Transmission via the Skin

Psychologists have long been interested in human perception and in particular, cutaneous sensitivity or the skin's sensitivity. As it covers the entire human body, the skin is one of the largest senses in the human body. It should come as no surprise that this has been an interesting subject for researchers. Similar to the human senses of hearing and sight, tactile thresholds are often logarithmic in nature [WWB$^+$86]; just noticeable differences differ exponentially rather than linearly. Furthermore, a number of properties of cutaneous sensitivity make its study less than straightforward. First of all, sensitivity across the body varies with location. The fingertip is one of the most sensitive body locations for touch and is also easily accessible. As a result, most studies have been done targeting the fingertip. Secondly, it is difficult to isolate the skin's response to amplitude and frequency changes due to energy summation at the skin; increased frequency is often perceived as an increase in amplitude as well. Another effect of energy summation is temporal adaptation for just noticeable different thresholds. Like the other human senses, the difference threshold shifts temporarily higher when energy has been applied to the skin. A final factor in this space has to do with using multiple actuators placed at different loci near one another. If they are too near each other, they may be perceived as a single stimulation. On the other hand, when the actuators close to each other are vibrated at phase offsets, sensory saltation occurs whereby the user perceives the stimulus to be moving from one locus to the other [Gel75].

Because vibration is one of the easiest forms of tactile stimulus to generate it is the one that has been explored the most. Typical studies use a tactor where a small contact head extends into the skin that vibrates. These can be thought of as miniature speakers and offer higher fidelity than the typical offset motors found in commodity phones. However, because the usable frequency range for vibration ranges from 20–

400Hz, this has inherently constrained the range of frequency that has been explored [Sat61].

A number of researchers have built on the results of these low level studies constructing more complex signals, studying the problem from an information theory point of view, looking at how many different bits of information can be transmitted. Brown et al. examined the number of distinguishable vibrotactile patterns that could be generated by using multiple actuators mounted at different locations on the arm, varying intensity, amplitude as well as duration [BBP06]. Tan and colleagues have explored a number of other body locations for vibrotactile actuators, with a focus on the number of bits of information that could be conveyed [TP97]. Tan et al. also looked at using multiple voice coil motors to stimulate three of the user's fingertips, again with a focus on the number of bits of information that could be conveyed [TDRR99]. These types of actuators are capable of a higher range of stimulus but have not been explored as much because they require more complex setups than vibrotactile actuators.

The fundamental findings on both cutaneous sensitivity as well as the information theory applications have shown the potential of the skin as a transmission medium.

1.3.2 Virtual Reality and Surgical Simulations

A subset of haptic research applications have focused on virtual reality as well as surgical simulations. These projects attempt to provide tactile feedback for simulating textured surfaces, which can be useful in these scenarios. To date, one of the most successful devices in this space has been Massie and Salisbury's *Phantom* [MS94]. This device uses three pulleys attached to a pen-like rod to provide force feedback. Users hold the device like a pen and are able to experience the feeling of brushing the tip of the pen across a textured surface. Other, lower fidelity force-feedback devices have also been explored in this space. Force-feedback joysticks have been used for gaming [OTT$^+$95] as well as telemanipulation [Agr87]. Users can feel physical feedback associated with manipulating a remote object. Gloves worn on the user's hand can also recreate the feeling of grasping virtual objects [BBPB02]. These gloves are typically connected to a larger rig, actuated with pulleys. More compact versions also exist with linear actuators placed within the palm of the glove. Because work in these areas is focused on recreating the experience of touching a virtual object, their main goal is to mimic the exact feeling of an object. For these scenarios, bulky actuators and large equipment are acceptable. These devices can be powerful, but they trade off size and complexity for fidelity. As a result, they are unusable in mobile scenarios where miniaturization is important.

1.3.3 Tactile Feedback in Mobile Devices

Like visual and auditory feedback, tactile feedback on mobile devices is also constrained by size. Despite the lower fidelity tactile feedback in this space, they still provide many of the benefits. Typically they are used in multimodal interfaces in combination with visual and auditory feedback [MH08]. While some projects in this space have looked at increasing bandwidth [brown], many others have tried to provide additional experience to the devices. The *Haptic Pen* recreated the feeling of clicking a button with a stylus by adding a solenoid to the end of the stylus [LDL$^+$04]. Harrison and Hudson recreated the feeling of pressing a button on touchscreen devices by using a bladder filled with air [HH09]. *Ambient Touch* provides tactile feedback by vibrating the screen when it was pressed with a stylus [PMR02]. Although the entire screen was vibrating, it felt like only the contact point was vibrating because that is all users could feel. These devices attempt to recreate the feeling of pressing a button because it has a number of benefits such as being able to preview before committing to press [RISO]. Navigation has also been explored either by taking advantage of sensory saltation [ELW$^+$98]. Arrays of piezoelectric tabs have been used to make scrolling through lists feel more natural [LPL$^+$06]. The *Haptic Knob* proposed a force-feedback enhanced dial for enhancing the experience of going through multimedia [SMS$^+$01].

These projects attempt enhance a user's interaction by mimicking the exact feeling of interacting with a physical device. These solutions are typically focused on creating a more natural experience, rather than trying to convey tactile messages.

1.3.4 Tactile Communication

For the most part, tactile communication on mobile devices has been limited to the vibration alerts that tell a user someone is trying to contact them. Chang et al. looked at *Audio Haptics* where a single transducer could be used to generate both tactile as well as auditory feedback [CO05]. *ComTouch* looked at how users would communicate with unstructured vibrotactile sequences, without any prior training [COJ^{+}02]. At the other end of the spectrum, Geldard's *Vibratese Language* explored the limits of what could be learned with extensive training, creating a mapping of the English alphabet to vibration pulses, similar to Morse code [Gel60]. After 65 hours of training, users were able to receive messages with 90% accuracy at approximately 38 words per minute.

The focus of this dissertation is closest to this category. Most of the projects in this space are single projects focused on creating a more natural experience. In many ways this is one way of taking advantage of human experience but applied to a narrow application of tactile feedback rather than a tactile message.

Although this section has provided a general overview of the haptic space, this review has hardly been exhaustive. A more in-depth treatment of the work related to some of the projects associated with this dissertation appears in each of the subsequent chapters.

1.4 Creating Tactile Messages Based on Human Experience

The main problem this thesis focuses on is that users need to be able to interact with their mobile devices in eyes-free scenarios when they cannot look at their devices. Furthermore, they need to be able to do so in a way that is easily learnable. I propose that the way to do this is to take advantage of human experience. Humans already associate different stimuli with different pieces of information (e.g. a song reminds them of a friend). If we can map these stimuli to the tactile space, then we will gain the corresponding information associations for free.

To get an idea of the design space, consider how humans interact with the world around them eyes-free, in the absence of technology. They do so using their four remaining senses (the visual sense does not apply for obvious reasons) smell, taste, touch, and sound. Smell and taste do not have straightforward computational equivalents, although people are working on very interesting solutions for that problem. Instead, this

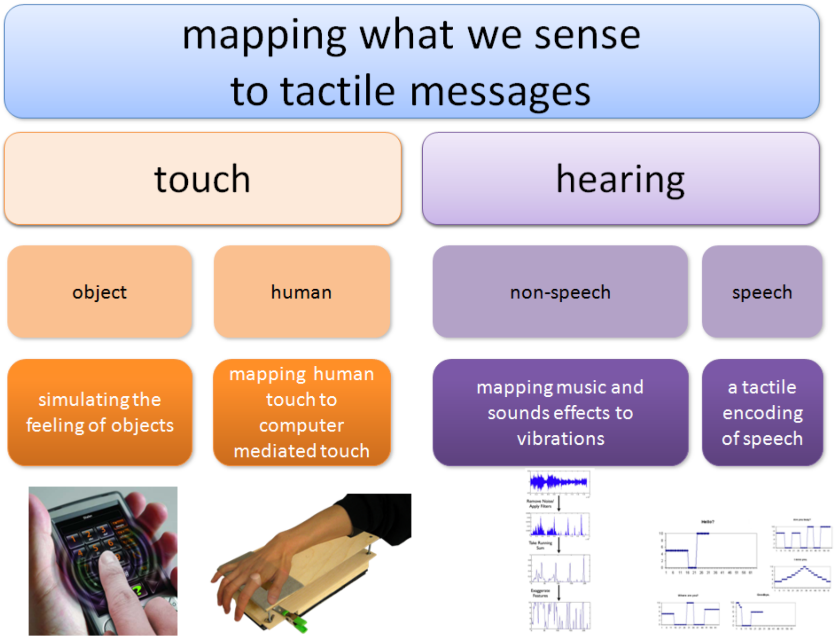


Figure 1.1: Taxonomy we used to guide our exploration of mapping stimuli that people are familiar with to the tactile channel.

thesis focuses on how stimuli that people sense with touch and sound can be mapped to the tactile channel. The main contribution of this thesis is a proof of concept that a user's information associations from other experiences can be carried over to the tactile channel, by mapping the appropriate stimuli.

Figure 1.1 illustrates a taxonomy that we have used to guide our exploration in this space. Within each category of stimuli, we have broken the different types of stimuli into subcategories. When people use their sense of touch, they can be touching other objects or other people. Similarly, when people use their sense of hearing, they can either be processing non-speech audio or speech. Within each of these subcategories, we have worked on a project that maps stimuli from that subcategory to the tactile channel. We have examined how music, human touch and speech can be mapped to the tactile channel.

1.5 Overview

Chapter 2 provides a motivating example, looking at how auditory feedback can be used to enable access to mobile phone content while talking on the phone. This chapter demonstrates many of the benefits of auditory feedback as well as how some of the problems of eyes-free interaction can be addressed.

Chapter 3 examines how music can be mapped to vibrotactile sequences. Using a buddy proximity application designed for mobile phones, we examine how people can use vibrotactile mappings of music for nice-to-know information such as buddy proximity. Although vibrotactile technology was successful for providing mappings of music cues, we were frustrated by the bandwidth limitations of this type of technology. Motivated by some of the shortcomings of the technology used for generating vibrations in commodity phones, we looked towards new types of actuators.

Chapter 4 looks at how voice coil motors can be used to create forms of tactile feedback similar to interpersonal touch. Unlike vibration, we were more interested in producing tactile feedback that was familiar, more closely resembling interpersonal touch. We present two prototypes for generating computer-mediated tapping and rubbing sensations. Our user studies demonstrate that we were successful in recreating experiences that people associate with tapping and rubbing from interpersonal communication. While interpersonal touch is a good mechanism for conveying intent, it does not convey semantics as well. In an effort to bridge this gap, we looked towards techniques for conveying semantic information via a tactile channel.

In Chapter 5 we present the initial results of trying to create a vibrotactile encoding of prosody. A series of user studies demonstrates how people perceive different linguistic aspects of speech to be mapped to the tactile channel. This approach has been applied to a messaging backchannel application for couples.

Chapter 6 examines future directions stemming from this research, looking at how the concept of leveraging human experience can be carried into the space of tactile message generation.

Finally, Chapter 7 concludes this thesis and describes potential areas for future work.

Chapter 2

Enabling Eyes-free Interaction with Auditory Feedback: A Motivating Example

Our first exploration into the space of enabling eyes-free interaction looked at using auditory feedback to provide information to users.[1]

2.1 Motivation

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:

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?

---

[1] Note that this chapter is a reprint with minor changes of *BlindSight: Eyes-free Access to Mobile Phones* co-authored by Kevin Li, Patrick Baudisch, and Ken Hinckley.

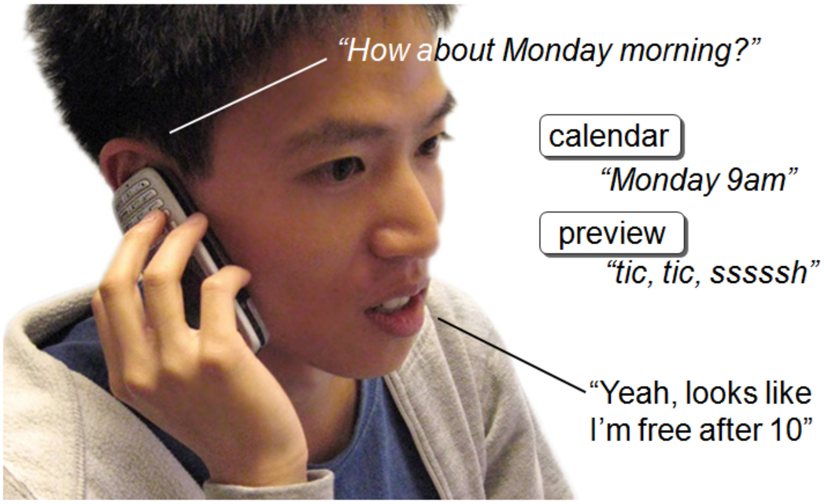


Figure 2.1: To access phone information stored on the phone mid-conversation, users press buttons and receive auditory confirmation. This photo shows the *flipPhone* form factor.

John: 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 do not use headsets because they interfere with real-world situational awareness and are often judged as uncomfortable, unattractive, or socially awkward [FT99, Ito05]. 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 2.2. 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.

2.2 Related Work

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

2.2.1 Auditory Feedback

The strengths and weaknesses of auditory feedback have been studied extensively in the field of interactive voice response systems [ME97]. One of the main challenges is that audio prompting forces users to wait (resulting in “touch tone hell”

[YZ06]). Users should be able to "dial through" to interrupt prompts, or "dial ahead" to skip familiar prompts [Win]. Perugini et al. propose dial ahead using speech [PAM07]. *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 [RV92]. *Zap and Zoom* improves on *Skip and Scan* by allowing users to jump directly to a location using shortcuts [Hor94]. Yin and Zhai proposed using a visual channel in parallel to using an interactive voice response system to inform users about their options [YZ06], but this is counter to our design goal of eyes-free interaction.

While any human-human conversation contains a certain amount of redundancy [TW06, DY01], 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 [DY01]. Tucker and Whittaker compare leaving out words with increasing playback speed [TW06]. Non-speech audio may be less distracting than speech audio, but can convey information such as navigational cues in hierarchical menus [Bre98, Gav89, HS96]. Zhao et al. explored eyes-free menus driven by auditory cures [ZDC$^+$07].

Tactile feedback offers another alternative. For example, Luk demonstrates piezoelectric-driven feedback for mobile devices [LPL$^+$06]. We defer the discussion of tactile feedback to later chapters.

2.2.2 Mobile Input

BlindSight allows for one-handed input using a phone keypad. Keyboard-based entry with few buttons can be supported through iteration [Mac02] or through chording (e.g. the *Twiddler* keypad [LSP$^+$04]).

In some contexts, gestures can enable experts to perform eyes-free operations. For example, text entry based on *Unistroke* [GR93] or *EdgeWrite* [MI08] can become nearly eyes free, even with distractions [GWC$^+$07].

One of the form factors we explore in this paper receives input on the back of the device. *BehindTouch* [HMT03], *HybridTouch* [SH06], and the isometric joystick-based version of EdgeWrite [WCM07] also explore using the back surface of mobile devices.

LucidTouch [WFB+07] enhances back-of-device interaction by visualizing the user's hand position.

Mobile phone interaction and in-car navigation [Win, Tel] can sometimes successfully employ speech recognition. In situations with a fixed and small vocabulary very good recognition rates have been achieved [HPG+00]. 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* [LSS+04]).

2.3 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.

2.3.1 Results

Figure 2.2 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.

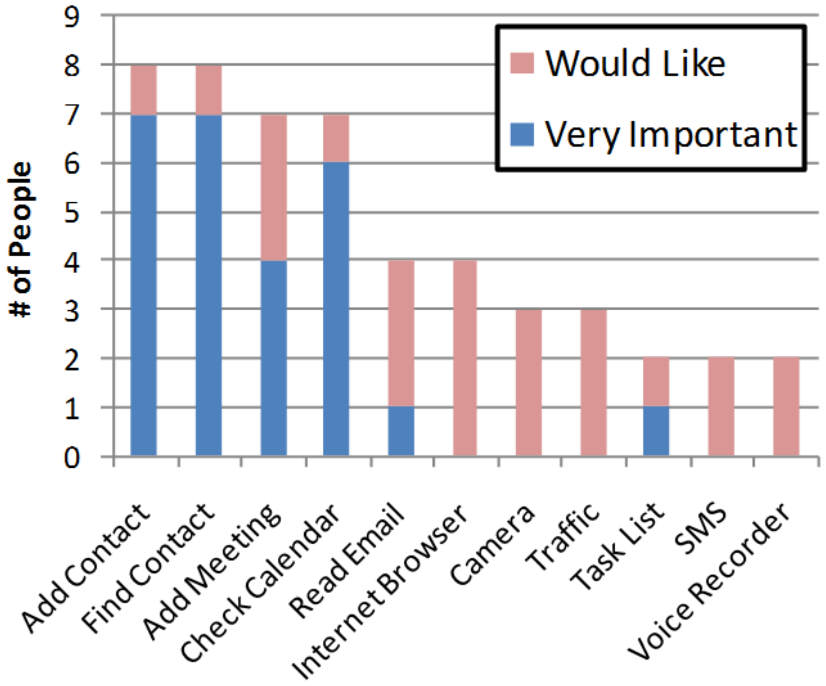


Figure 2.2: Number of participants out of nine who rated the respective feature as "would like" or "very important"

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 3×4 keypad found on traditional mobile phone form factors, rather than creating a custom key layout.

2.4 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.

2.4.1 Design Principles

The rationale behind using auditory feedback during a phone conversation is that any human-human conversation contains a certain amount of redundancy [DY01, TW06]. If part of the conversation is lost, e.g., because of dropouts 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 [Bau98]), 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. **Interruptibility:** 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 (*interruptibility*), which turned out to be faster than the use of a separate confirm button.

2.4.2 Menu Organization

Figure 2.3 shows blindSight's menu structure. All menus are based on the 3×4 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 2.4. 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 2.3a-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.

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 [Tes81], and are even more problematic for eyes-free applications. We minimized mode switching by avoiding

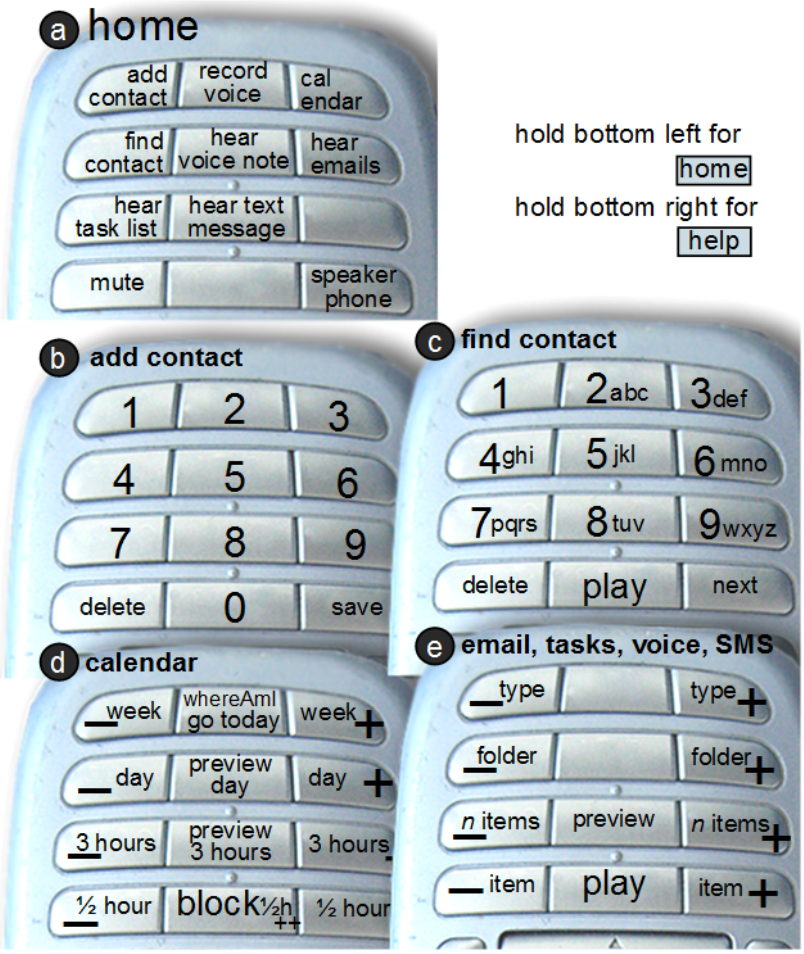


Figure 2.3: BlindSight's menus

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 2.3, 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 auto-filed 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.

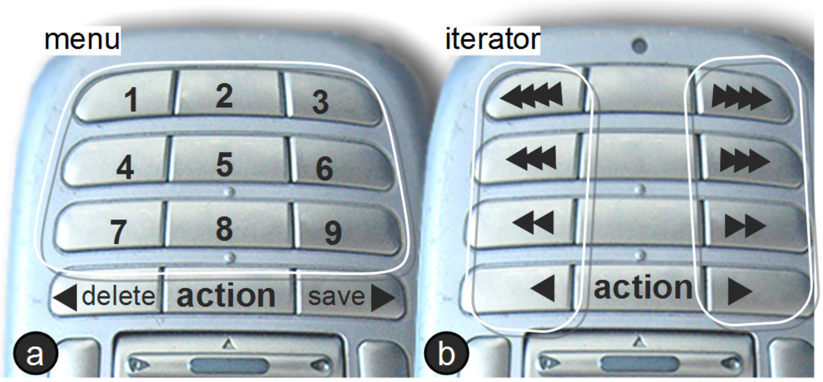


Figure 2.4: (a) The menu mapping and (b) the iterator mapping.

2.4.3 Walkthrough

We now revisit the scenario from the introduction section, this time using blindSight. We show blindSight interactions like this: 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. Figure 2.5 illustrates the walkthrough.

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?

calendar *“calendar”*
calendar **(enters calendar)** *“Monday 9am”*
week + *“next Monday”*

John: How about Tuesday morning sometime?

Ami: day + *“next Tuesday”*
preview 3 hours *“tic, sssh, tic”* [tic=“busy”, sh=”free”]

Ami realizes that noon is taken & looks for alternatives

3 hours+ *“noon”*
preview 3 hours *“tic, tic, ssssh”*

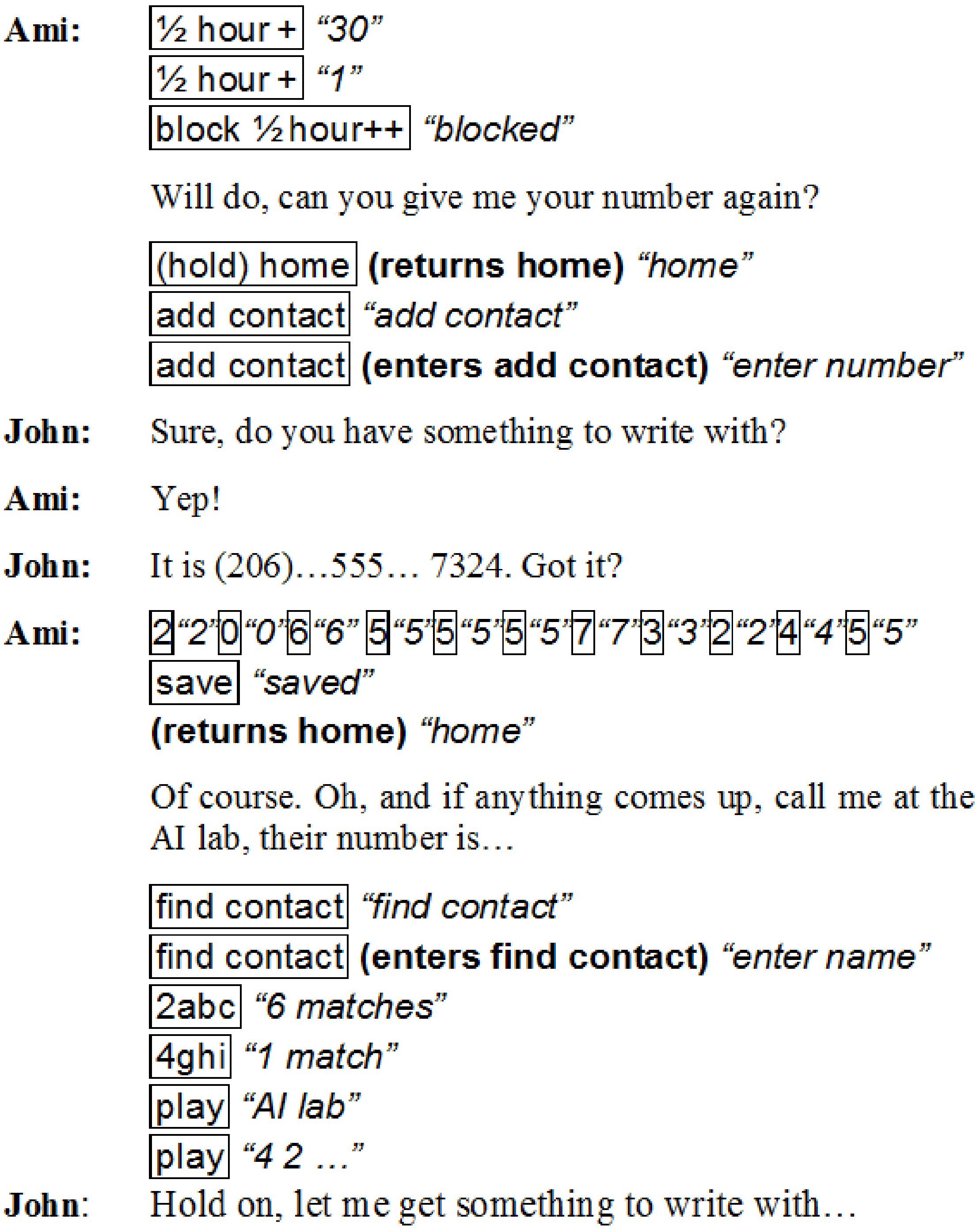
I’m busy in the morning, but I am free in the early afternoon.

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

Ami: day + *“Wednesday”*
preview day *“tic,ssh,tic, ,sssssssh”*

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: ½ hour + *“30”*
½ hour + *“1”*
block ½ hour++ *“blocked”*

Will do, can you give me your number again?

(hold) home **(returns home)** *“home”*
add contact *“add contact”*
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: 2*“2”*0*“0”*6*“6”* 5*“5”*5*“5”*5*“5”*7*“7”*3*“3”*2*“2”*4*“4”*5*“5”*
save *“saved”*
(returns home) *“home”*

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

find contact *“find contact”*
find contact **(enters find contact)** *“enter name”*
2abc *“6 matches”*
4ghi *“1 match”*
play *“AI lab”*
play *“4 2 …”*

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

Figure 2.5: Walkthrough of blindSight usage scenario.

2.4.4 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.

2.5 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 2.6 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 2.6a) requires index and middle fingers to impose a firm grip on the phone to hold it. Unfortunately, this posture causes the thumb to hit the keypad at an oblique angle, preventing users from feeling tactile features on the phone keypad. Thus, this posture makes keypad operation error prone. A pilot study of this posture during which participants entered random sequences of numbers eyes-free (similar to the study described in section 2.7 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 2.6b), but this may not always be possible or desirable.

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.

Figure 2.6: (a) Accessing the built-in phone keypad using one hand and (b) two hands.

2.6 Adding Tactile Features to the Phone Keypad

Figure 2.7a 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 2.7: (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 2.8). Larger gaps between buttons and rounded buttons seemed to address the problem

(Figure 2.8a). We found a Smartphone that possesses these characteristics, the *Red-E SC1100* shown in Figure 2.7b. Since this phone is no longer commercially available, we also modified our *Audiovox 5600* phone by cutting grooves between its keys (Figure 2.9a). 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 2.8: 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 2.7b), 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 2.9). 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.

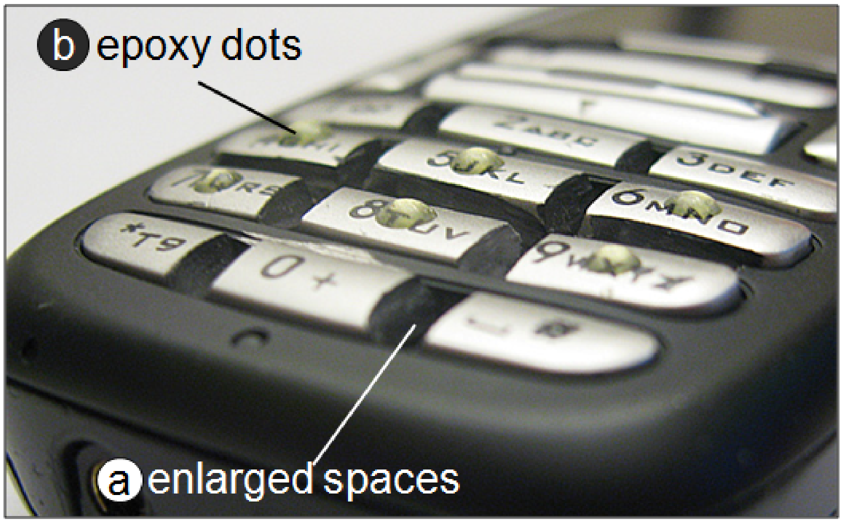


Figure 2.9: We modified this *Audiovox 5600* keypad by (a) enlarging the gaps between buttons and (b) adding epoxy dots on buttons 4 through 9.

2.6.1 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 2.6a. 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 2.10, 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 2.1. 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.

Flipping the phone meant changing the mapping of the buttons to the rotated mapping already illustrated in Figure 2.3 and Figure 2.4. Relabeling the keys was not necessary, because users do not see the keypad when it is flipped. Users feel the tactile

Figure 2.10: This typical phone holding posture places index and middle fingers on the back of the device.

features though, which is the reason for the double row of features shown in Figure 2.9; this arrangement was symmetric and therefore preserved meaning when rotated.

2.6.2 Implications for 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 3×4 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 flip-Phone-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.

2.7 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.

2.8 Interfaces

There were three interface conditions.

In the *Ear* condition, participants held the phone against their ear as shown in Figure 2.6a. 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 2.1. 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 2.7b. 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.

2.8.1 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 the moment a key was pressed.

2.8.2 Apparatus

The study was conducted using the *Red-E SC1100* phone shown in Figure 2.7b, with epoxy dots. It offered 16MB RAM and a 132MHZ Processor and ran the Microsoft Smartphone 2003 operating system.

2.8.3 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 message user, sending about 300 messages per month. The remaining participants reported sending less than 30 texts/month and less than a year of experience.

2.8.4 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.

2.8.5 Results

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

Error rates:

Errors rate is 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 2.11).

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,

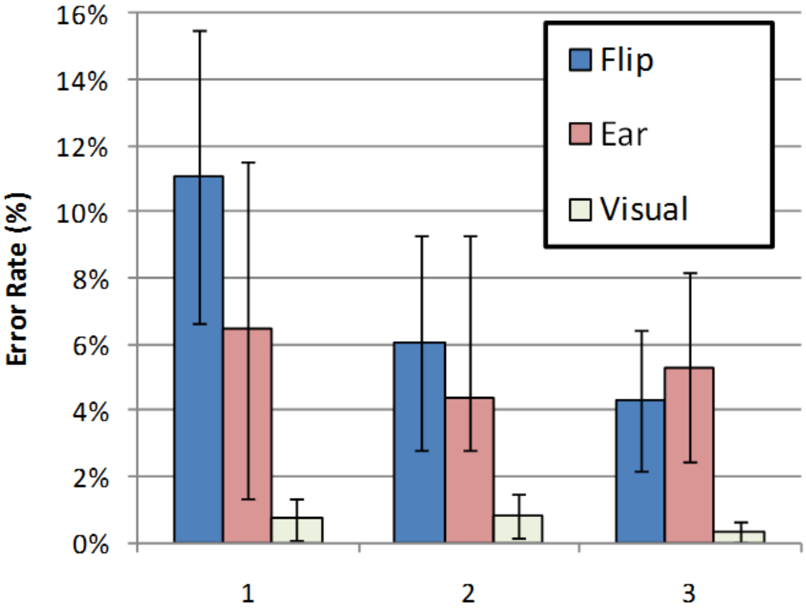


Figure 2.11: Error rates for the Flip, Ear, and Visual conditions by block number. Error bars show 95% confidence intervals.

medians were used as a measure of central tendency to reduce the effect of outliers [RG83]. 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.

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

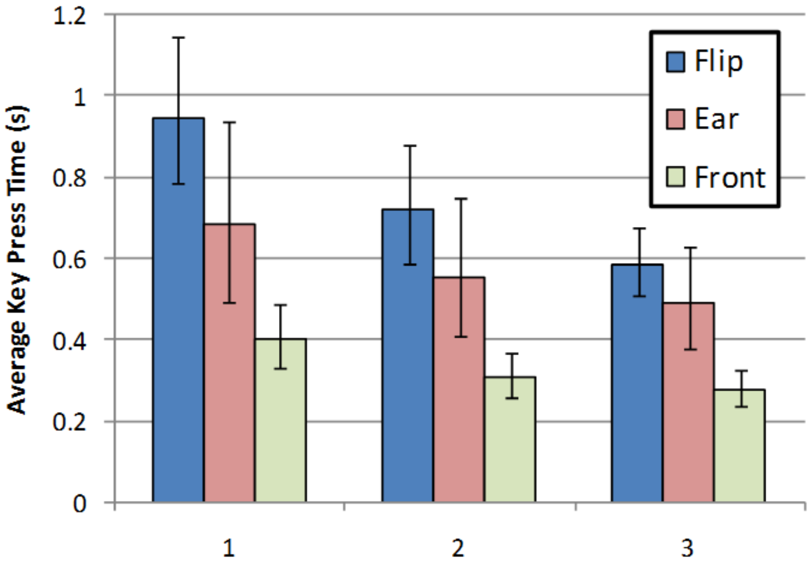


Figure 2.12: Average key press times for each interface condition by block. Error bars show 95% confidence intervals.

on the buttons for the two eyes-free conditions. Two participants commented on how the Flip condition felt similar to holding a mouse.

2.8.6 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.

2.9 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.

2.9.1 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 2.9a. 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 on-screen 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 2.3. 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.

2.9.2 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.

2.9.3 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.

2.9.4 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.

2.9.5 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 talking on the phone while driving.

2.9.6 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.

2.9.7 Results

All participants completed both tasks successfully for all conditions. Figure 2.13 shows the tallied responses of the seven questions that required participants to choose between interfaces; Figure 2.14 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 2.13). 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."*

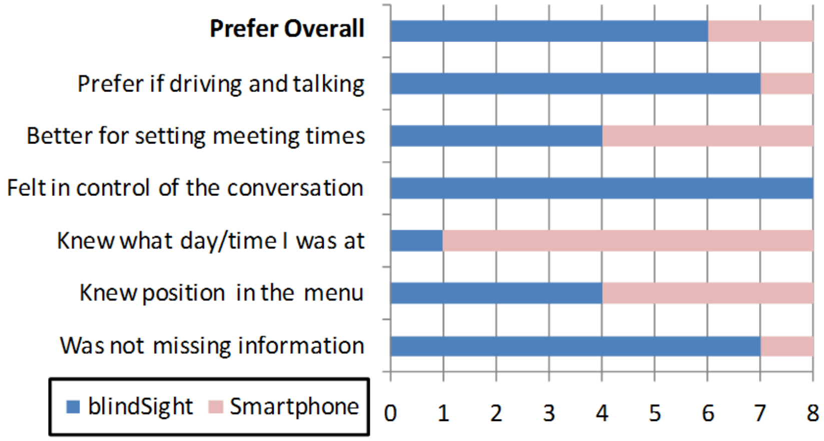


Figure 2.13: The number of participants (n=8) who preferred the blindSight or Smartphone conditions in the context of the respective statement.

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 2.14) 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.

The questionnaire identified calendar navigation as a weakness in the tested ver-

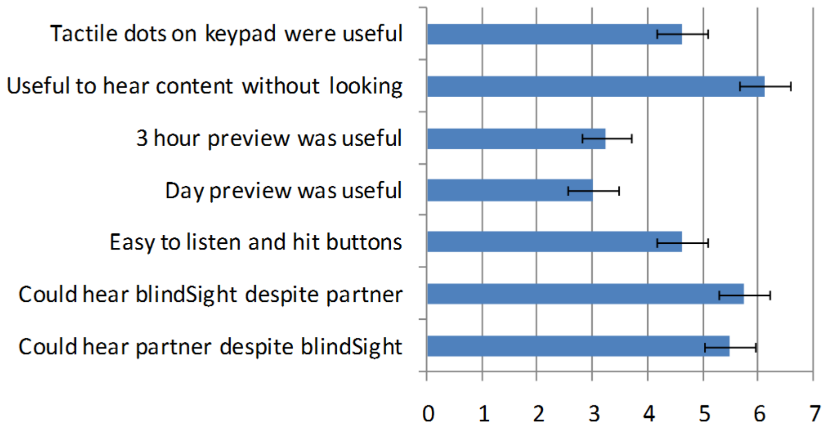


Figure 2.14: Likert responses (7-point scale) for questions regarding the eyes-free condition. Higher ratings are better. Error bars represent standard error of the mean (n=8).

sion 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.

2.10 Properties of Auditory Feedback

In this chapter 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.

Our results demonstrated that auditory feedback can be used to enable eyes-free interaction in scenarios where users are unable to look at the phones. Using an application such as blindSight allows users to access mobile phone content in situations where they otherwise would not be able to.

As far as enabling eyes-free scenarios, auditory feedback has a number of nice properties. It is easily learnable; any user that understands English can quickly learn how to use the system. However, auditory feedback is not always applicable due to situational factors. It would be difficult to use auditory feedback when in a loud environment. Additionally, there are a number of social scenarios where it might be unacceptable to use auditory feedback, such as in the middle of a meeting.

To address these scenarios where auditory feedback is unusable, we shifted our focus towards tactile solutions. The next chapter describes our examination of mapping music and sounds to vibrations.

2.11 Acknowledgments

This chapter, in part, is a reprint of the material as it appears in BlindSight: Eyes-free Access to Mobile Phones. Li, K.A., Baudisch, P., and Hinckley, K. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2008), pp. 1389–1398. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Mapping Music and Sounds to Vibrations to Enable Buddy Proximity Cues on Mobile Phones

Auditory feedback is useful for conveying information in a number of scenarios, such as when driving. However, there are a number of scenarios where auditory feedback is unusable due to environmental conditions. For example, it would be difficult to use auditory feedback in loud environments such as on a crowded subway. Additionally, there are a number of situations where it is socially unacceptable to use auditory feedback, such as when talking to someone. Motivated by these shortcomings of auditory feedback, we focused our exploration on tactile feedback. This chapter looks at how to map music to the tactile space. To ground this in a practical application, we built a buddy proximity application. Because we had to solve a number of system level problems in this process, we discuss those here as well.[1]

---

[1] Note that this chapter is a reprint with minor changes of *Peopletones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones*, a paper co-authored by Kevin Li, Timothy Sohn, Steven Huang, and William Griswold.

3.1 Motivation

One vision for ubiquitous computing is a context-aware infrastructure that can simplify and enrich our lives by helping us with tasks that might otherwise be out of our reach. For example, location-based services such as Loopt can detect the proximity of friends that are just out of sight or unnoticed [Loo]. Such applications can be useful for a variety of scenarios such as arranging ad-hoc meetings. To date such wide-scale applications have depended on specialized phone and carrier capabilities to detect proximity, both at a real cost to the user. Moreover, the user must make a conscious effort to look at the phone to learn of friends' proximity, lessening usefulness.

Realizing the ultimate vision depends on a ubiquitous mechanism for detecting such occurrences. For "nice to know" contextual information like the proximity of friends, we also need an unobtrusive mechanism for making us aware of them. To achieve true ubiquity—so that any two friends could be aware of their proximity—both must be achieved at little cost. In this chapter we explore the technologies of mobile phones and peripheral cues for the ubiquitous sensing and reporting of "nice to know" context through PeopleTones, an application for buddy proximity:

- Commodity mobile phones satisfy the ubiquity criterion (and by extension the cost criterion). As of 2007, there are 3.3 billion mobile phone subscribers worldwide [Bre98, BB04b]. Moreover, mobile phones possess both a number of sensors (e.g., microphone, camera, and GSM radio) and actuators (e.g., speaker and vibration motor), making phones a potentially ideal platform for ubiquitous computing. On the other hand, the sensors and actuators are of notoriously low quality, complicating precise sensing and high-fidelity actuation. Inference can be especially problematic when comparing readings between phones [CSC$^+$06].

- Peripheral cues, like those explored in office and home environments, are an attractive modality for "nice to know" information; they can apprise users of information without interrupting their current task. However, getting peripheral cues to work with commodity mobile phone actuators in the wild is an open challenge.

For detecting proximity on phones, our algorithm compares cell towers seen by the mobile phone clients to estimate proximity. This privacy-friendly approach does not require knowledge of actual location. However, GSM's long range and random characteristics means that a phone will, for example, occasionally detect cell towers that are miles away. We filter the proximity data using a simple state machine based on a 2-bit counter [HP97]. The state machine also helps to conserve power by sampling less frequently when two phones are considered near or far away. Power is further conserved by withholding reports when GSM signals are weak; proximity detection in this case is imprecise and extra power is required to report it.

Proximity can be reported by sounds, and past work has shown audio to be effective for delivering peripheral cues [MBWF97]. However, it is untenable to expect the use of headphones or similar devices to reduce the unobtrusiveness of cues or increase comprehension. We hypothesize that keeping audible cues short can improve their unobtrusiveness, but that may not be adequate for many uses. We propose using the vibrations provided by mobile phones, as they are private, subtle cues [HLR01], and likely to be etiquette-friendly. Vibrotactile cues that correspond to known audible cues (i.e., they "vibrate like the sound") can provide a parallel "private" vibration language without requiring the user to learn an arbitrary mapping.

However, the inexpensive vibrotactile actuators found on mobile phones today only have a binary on/off setting, severely limiting their communication abilities. To provide vibrotactile cues corresponding to the audio cues, we introduce an offline digital signal processing technique that captures the essence of audio cues. These patterns were realized on the mobile phone's limited vibrotactile actuator using our software algorithm. Using a technique similar to pulse width modulation, we can generate a range of amplitudes.

To explore these mechanisms we performed both controlled and *in situ* user studies of PeopleTones. First, we measured the precision and recall of our proximity detection algorithm using a large dataset collected from wardriving the Seattle area [CSC$^+$06]. Second, we lab-tested 17 users on their ability to identify how vibrations corresponded to music clips. Finally, we designed and deployed PeopleTones, a system for conveying buddy proximity via peripheral cues that are uniquely assigned to buddies. PeopleTones was deployed to three groups of friends (the same 17 users as above). Each group used a different cue-to-buddy mappings: nature sounds, music sounds picked by the buddies, and music sounds picked by the recipient of the cues. To uncover possible learning of the vibrotactile cues during the study, we repeated the lab test at the completion of the study.

For proximity detection, our findings show that our approach has excellent precision (few false positives) and fair recall (a split between true positives and false negatives). The two bit counter reduces false positives by up to 84.9% and increases precision to 99% at a threshold ratio of 0.4. The fair recall is adequate because proximity is most valuable for people who are lingering near each other (e.g., not driving), and such behavior provides many chances to produce a positive report. Such lingering also diminishes the effects of the cell towers that were cached in the phone upon arrival to the area, which is influenced by the towers seen along the users path to the destination. Managing power by dynamically adjusting the sampling rate enables a phone to run for one to two days, as opposed to 4–6 hours.

A general theme of the qualitative results is the importance of personal control for peripheral cues. Using ambient noise to detect social situations was explored as a way of choosing audio versus vibration cues, but most users opted to enforce explicit control of their cue delivery modality. Users who selected the cues they would hear found the system more useful and were also better at identifying corresponding vibrotactile patterns generated by our algorithms.

After a discussion of related work, Sections 3.2 and 3.3 detail our approaches to proximity detection and delivering peripheral cues on mobile phones. Section 3.5 introduces PeopleTones, and Sections 3.6–3.9 describe the results of our *in situ* user study.

3.2 Related Work

We build on work from three areas: proximity sensing, mobile peripheral cue systems, and auditory and tactile cues.

3.2.1 Proximity Detection in the Wild

A number of technologies have been proposed for proximity sensing. Infrared approaches such as those used by *Meme Tags* [BMV$^+$98] provide good accuracy but require line of sight. Ultrasound approaches such as *Activebadge* [WHFG92] also provide good accuracy but they require infrastructure support. *Hummingbird* uses short range radio which allowed Holmquist et al. to explore deployments in the wild [HFW99]. This approach provided good proximity detection but required specialized hardware, which created complications for in-the-wild deployment. *PlaceLab* uses estimates of cell tower positions to provide location [LCC$^+$05]. This approach provides excellent coverage and adequate accuracy for detecting something like buddy proximity (e.g., median accuracy of 94–196m and 90th percentile accuracy of 291–552m, using a single carrier's towers), but it requires a "wardriving" of the area to obtain location estimates for cell towers in the area. This can be quite costly, especially keeping the information up to date, as tower positions, etc. are updated on an annual basis.

One method of acquiring location on some phones is through the network carrier, but they often do not release the required APIs. *Loopt* is an example of a commercial system that enables sharing location with friends using both GPS and carrier based location [Loo]. Unfortunately GPS is not yet widespread, suffers from not working everywhere, i.e. urban canyons and would violate user's privacy by requiring location reports to a server. *Dodgeball* explores self-reporting location [Dod] but would require user's to pro-actively monitor the system, which is counter to our goal.

Another method is to use a location infrastructure. *Place Lab* [LCC$^+$05], *Active Campus* [GSB$^+$04], and *Plazes* [peo] are examples that offer both absolute and relative positioning. These infrastructures limit sensing to areas with pre-mapped access points.

Rather than calculate absolute location, NearMe explores a few algorithms for detecting proximity using Wi-Fi signatures, allowing it to work with no *a priori* setup [KH04]. We use a similar approach but for GSM readings.

3.2.2 Mobile Peripheral Cue Systems

Peripheral cues have been heavily examined in office settings. *Audio Aura* [MBWF97], *Live Wire* [WB96] and *ambientROOM* [Ish08] are systems that play auditory cues for conveying information in the background. Peripheral displays are known to be difficult to evaluate [MDH$^+$03, MDM$^+$04] and peripheral cues suffer from similar problems. The success of peripheral cues in home and office environments suggests that they may be useful in the wild. Deployments in the wild often reveal uses not found in laboratory studies. Examples of this include location-sharing [SCL$^+$05] and reminders [SLL$^+$05].

Mobile context-aware platforms have been proposed for aiding instant messaging, an intended, explicit interaction scenario. *WatchMe* [MSS04], *Hubbub* [IWR02], and *Connexus* [TYB$^+$01] are examples of such applications, supporting the initiation of a messaging session by providing cues of availability (e.g., not in a conversation). Studies with *Nomadic Radio* found that auditory communication is useful for mobile messaging but minimizing intrusiveness requires more than, for example, detecting breaks in conversation [SS00]. One possibility is detecting activity transitions with accelerometers placed in the seat of a chair or worn on the body [HI05]. Such approaches are less viable in the wild. Many of these systems have suggested that audio cues could be used to identify different users. Commercial ringtones are similar in that they map a person's identity to an audio cue, but they are little studied and most phone users don't consider an incoming call a “nice to know” condition (and hence worthy of a peripheral cue).

3.2.3 Auditory and Tactile Cues

Gaver's work with auditory icons revealed the effectiveness of using sounds that are semantically related to the objects they represent [Gav89]. Brewster's work with earcons found that music timbres are better at conveying information than unstructured sounds [BB04b] and that non-speech audio can be effective for navigation [Bre98]. We build off of these findings, using structured sounds for our auditory cues and exploring how different types of sounds affect user response.

Tactile cues are subtle, private cues [HLR01] that have been suggested as a channel for ambient information delivery [PMR02]. Tactile perception cannot be fully utilized without a high-fidelity delivery channel [BLEW04]. Piezoelectrics have been proposed to convey information using touch, such as in Luk et. al's *Tactile Handheld Miniature Bimodal* [LPL$^+$06]. Vibrotactile cues have been proposed for a variety of uses such as for conveying information in a non-visual channel. Geldard's *Vibratese* language proposed a vibrotactile encoding of the English alphabet [GC56, Gel60] and *ComTouch* explored vibrotactile communication without learning (i.e., training) [CO05]. *Tactons* use specialized actuators similar to those found in mobile phones to generate distinct pulses, which have been shown to be effective for alerting users to message type as well as urgency [BBP06, BK06]. These works demonstrated that vibrotactile patterns can be differentiated. Multifunction transducers have been used to explore audio-haptics, playing vibration in conjunction with audio in mobile phones [COJ$^+$02]. Still, vibrotactile development on commodity phones is limited by APIs that provide only on/off functionality. The *VibeTonz* technology from Immersion supports richer, more complex vibrotactile pattern generation, but utilizes specialized hardware that is currently available on only a handful of commercially available handsets [Imm].

3.3 Proximity Detection

There were two design requirements we felt were necessary for a buddy proximity detection algorithm. First, it should be widely deployable in many environments with many phones, doing so in a privacy-aware manner. Secondly, since buddy proximity is "nice to know" information, it is important that when cues are delivered, friends are actually near one another. If too many cues are delivered when buddies are far away, users will stop using it. In the case of reporting when buddies are nearby, it is therefore important to maintain a high *precision*, even if this means lower *recall*.

Precision is defined as the number of near reports that are correct divided by the total number of near reports. High precision means that there are few false positives. *Recall* is the number of near reports that are correct divided by the total number of actual near occurrences. High recall means that most of the near occurrences have been detected.

PeopleTones does not need a person's geographic location to find the proximity of nearby buddies. Hence, we used a relative positioning method in the spirit of the *Nearme* server [KH04]. Nearme used a variety of metrics for comparing the distance between two wireless measurements, such as Euclidean distance, spearman rank correlation, and the ratio of common access points.

3.3.1 Proximity Detection Algorithm

To run controlled tests on a few different proximity detection approaches, we collected a small sample of cell tower readings from three regions with different population densities. These were obtained by sampling cell tower information from each of 3 mobile phones, all on the same carrier. Each phone recorded two samples while positioned each location. We took samples 5 minutes apart to approximate realistic behavior where users might linger at a particular location. To eliminate potential caching effects that may occur when reading cell tower information from the phone's memory, we reset all the phones in-between samples. One phone was kept stationary while the other two were moved away from the stationary one at 0.2mi intervals. The i-mate SP3i (HTC Tornado) phones we used are capable of reporting up to 7 towers at once. In summary, we used 2 samples per *phone* per *region*, 2 *phones*, 7 *distances*, and 3 *regions*, resulting in 84 readings. The purpose of gathering these readings was to test different algorithms for proximity detection on a realistic set of data. In our initial experiments, we found that computing the ratio of common GSM cell towers between two readings provided the best real-time proximity indicator. The intuition is that the closer two phones are, the more cell towers they will have in common. This ratio is simply the number of common towers between the two phones divided by the average number of towers seen. Figure 3.1 shows the equation we used for computing the ratio of cell towers between two phones, given two readings a and b, each consisting of a set of cell tower sector identifiers.

Figure 3.2 plots the averages of the *proximity-ratio* values for the three regions from which we collected data. From this plot we can see a clear trend for ratios to decrease as distance increases, although not consistently; there is a lot of noise. This suggested that the ratio approach would be promising for approximating distance between two stationary phones but we still needed to determine an appropriate ratio for a peripheral cue application’s needs.

$$proximity - ratio\ (a, b) = \frac{|a\ \cap b|}{\left(\frac{|a| + |b|}{2}\right)}$$

Figure 3.1: Equation used for calculating proximity ratio for two mobile phones where *a* and *b* are the sets of cell towers seen by each phone.

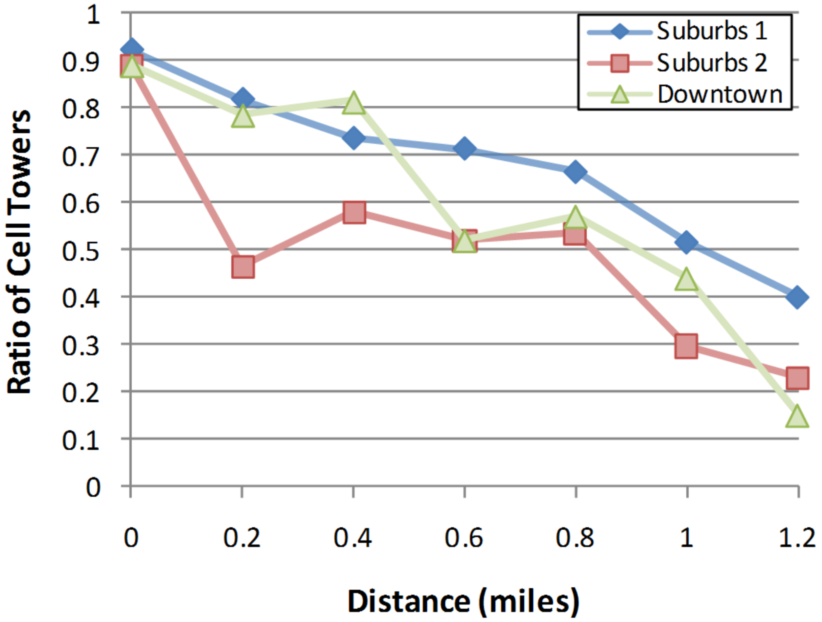


Figure 3.2: Initial cell ratio measurements taken from 3 different areas of varying population density.

3.3.2 Evaluating Cell Tower Ratio Algorithm for Proximity Detection

Evaluating our *proximity_ratio* algorithm was less than straightforward. It was difficult to obtain a suitably large and appropriate dataset for modeling two stationary phones at a variety of locations. Ideally, we would have simultaneously recorded readings from many stationary phones all at different locations with some ground truth measurement. However this would be hard if not impossible to achieve for a large number of phones. Instead, we used the dataset collected by Chen et al. from their wardriving of Seattle [CSC$^+$06]. During this process, they collected cell tower data along with GPS coordinates by driving around the greater Seattle area, equipped with a laptop, 2 mobile phones per carrier and a GPS device. They sampled the phones and GPS device approximately once per second to record cell towers seen by the phones and GPS coordinates. Since two phones were used per carrier, valid comparisons could be made between cell tower readings seen by the two different phones. We used readings from a *Downtown* area with an average cell tower density of 66 towers/km^2 and a *Suburban* area with an average cell tower density of 26 towers/km^2.

This dataset is not entirely applicable to buddy proximity detection. Because the data was collected from a moving vehicle, it only allows for modeling proximity situations where both mobile phones are moving quickly. We are most interested in scenarios when both phones are stationary or nearly so. Still, the data informs scenarios in which users may be moving. When people are driving, they are less likely to be interested in nearby buddies that are also moving, since it is unlikely that both will be available and have free time. In this case, lower recall rates are desirable. If two users are in actuality near each other, this is likely to be temporary and thus a system should not detect it.

Moreover, not all the points in the data set were collected at the same time, with some readings collected almost 5 hrs apart. Due to the load balancing employed by cell towers, comparing proximity between two phones seen at two different times is not an accurate model for a real-time application. To address this, we crosscut the dataset in different ways to approximate the precision and recall of the *proximity_ratio* algorithm for different scenarios. We then consider tradeoffs between precision and recall for different cutoff ratios in these scenarios. We were particularly interested in the behavior for two scenarios: when phones were at the same location, and when they were near each other. By breaking the analysis into these two different scenarios, we can use this dataset to evaluate our algorithm for a variety of distances.

Same Location

To analyze behavior when phones are in the same location (within 100m) and when participants are lingering in areas near each other, we extracted pairs of cell tower data where the readings were taken within 5s of each other. This yielded *28,625* pairs from Suburb and *19,087* pairs from *Downtown*. Analysis of GPS readings confirmed 99.9% of these points were within 100m of each other. We then calculated precision and recall numbers based on calculations of ratios for these comparisons. Table 3.1 shows recall values for different ratios in the *Downtown* and *Suburb* areas. Recall is higher for low ratios and tapers off for ratios between 0.3 and 0.4. Precision is 99.9% since the subset falls within 0.1km.

Ratio	Recall (Downtown)	Recall (Suburb)
0.1	0.96	0.96
0.2	0.84	0.85
0.3	0.83	0.83
0.4	0.57	0.58
0.5	0.44	0.44

Table 3.1: Recall for different ratios with a distance threshold of 0.1km when phones are at the same location. Precision is 99.9% since the subset falls within 0.1km.

Evaluating Near Each Other

We were also interested in situations where two mobile phones were near each other but not necessarily right next to each other. Since the data points collected from this set were taken from the same car, the phones were always next to each other at any particular time. Thus there was no way of getting same-time data from two phones that were far apart. To approximate situations where phones are near one another, we extracted pairs of readings taken within 90s of each other resulting in *569,264* pairs from the *Suburban* dataset and *379,285* pairs from the *Downtown* dataset. We then calculated the *proximity_ratio* for these pairs. Despite the higher recall rates for low ratios reported in the previous section, we knew from our initial studies that low ratios would have a much lower recall in a realistic setting since they detect phones that are miles away as "near" as well, so we focused on ratios higher than 0.3, since this is when recall rates were seen to decrease in the analysis of phones in the same location.

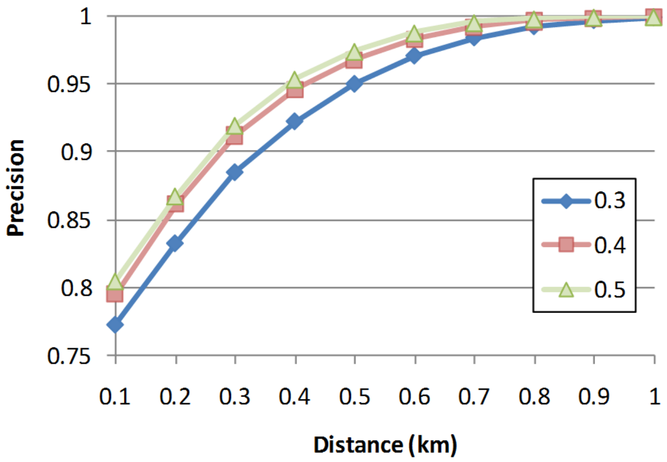


Figure 3.3: Precision for different "nearby" distances in *Suburb*.

Ratio	Recall (Downtown)		Recall (Suburb)	
	0.1km	**1.0 km**	**0.1 km**	**1.0 km**
0.3	0.74	0.66	0.67	0.66
0.4	0.50	0.41	0.42	0.40
0.5	0.39	0.30	0.30	0.29

Table 3.2: Recall rates at distances 0.1km to 1.0km for different ratios when phones are nearby.

Figure 3.3 and Figure 3.4 show the precision of the *proximity_ratio* algorithm when different threshold ratios are used for near/far determination. For phones in this *near each other* scenario, the lower the ratio, the lower the precision. The precision for *Downtown* is higher than that for *Suburb* which is not surprising considering the higher

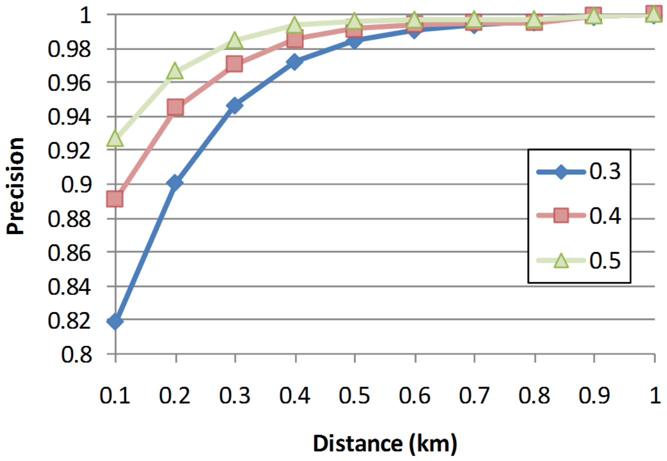


Figure 3.4: Precision for different “nearby” distances in *Downtown*.

cell tower density in this region. The range of recall rates observed over this area are shown in Table 3.2. While lower ratios still have higher recall rates than higher ratios, this comes at a cost of precision. Considering the design requirements for a peripheral cue application discussed earlier, precision is more important than recall. The issue of how close is close enough is addressed later in section 3.9. Precision increases as the distance threshold is increased since more false positive results become true positives. It should be noted that precision hits 99% at 0.5km in the *Downtown* case and at 0.7km in the *Suburb* case suggesting that even when false positives are delivered, these false positives are within 0.5km and 0.7km for *Downtown* and *Suburb* locations.

Far Apart

To validate *proximity_ratio* for phones that are far from each other, we decided to look at the entire dataset, even though we knew the temporal problems we described earlier would confound our analysis. For scenarios when phones are far apart, we were particularly interested in low recall while maintaining precision. Specifically, we wanted to make sure that increasing distance would not result in more false positives.

By comparing all of the pairings of readings from one phone to readings from the other phone, we obtained *55,181,015* Suburb pairs and *36,769,390 Downtown* pairs.

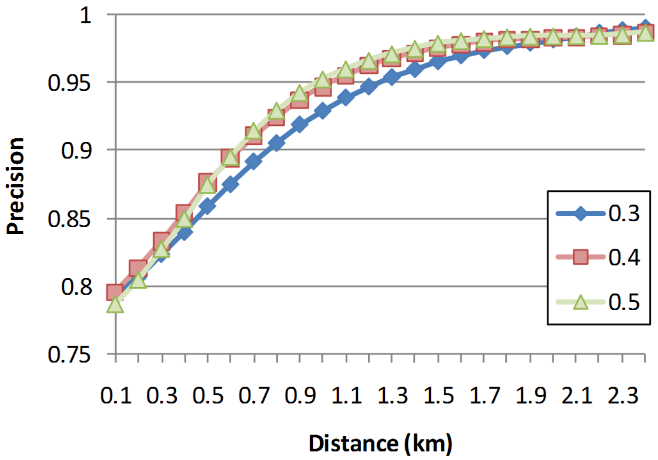


Figure 3.5: Precision at different distances in *Suburb* for the entire data set.

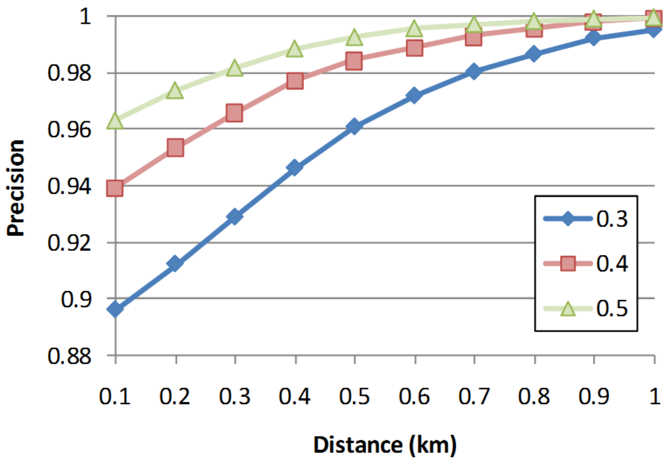
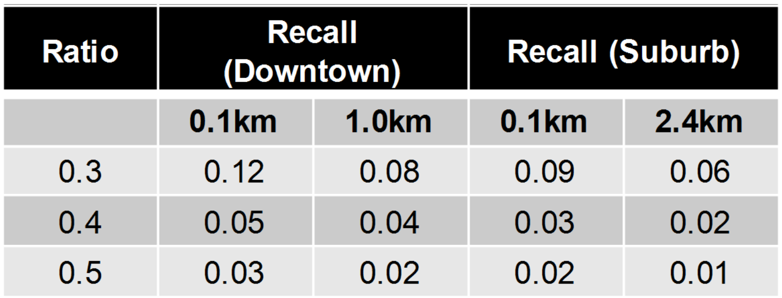


Figure 3.6: Precision at different distances in *Downtown* for the entire data set.

Figure 3.5 and Figure 3.6 show the precision at different distances for this set of comparisons. The precision values for this data follow the same trends observed before. Precision is higher for higher ratios and all ratios are able to obtain precision of 99% by some threshold (1.0km in *Downtown*, 2.4km in *Suburb*). These findings confirm that the *proximity_ratio* algorithm is effective at reducing false positives for moving phones that are far apart. However, recall rates are significantly lower (Table 3.3). The rates from the previous sections are much higher and based on more relevant data.

In our experience, the actual precision is lower than that calculated with this dataset and the recall is higher. For the reasons described earlier, it is difficult to gather the cell reading data from many phones simultaneously that we would need for a valid model of phones that are stationary at different distances. We defer qualitative analysis of *proximity_ratio* to section 3.9.



Ratio	Recall (Downtown)		Recall (Suburb)	
	0.1km	1.0km	0.1km	2.4km
0.3	0.12	0.08	0.09	0.06
0.4	0.05	0.04	0.03	0.02
0.5	0.03	0.02	0.02	0.01

Table 3.3: Recall rates for different ratios in the entire dataset.

3.3.3 Sensor Noise

GSM readings can vary widely from moment to moment in ways unrelated to the phone's proximity to the cell towers in the region. This creates the possibility of false proximity detection. Additionally, if buddies hover around a 0.2mi distance from each other for a prolonged period of time, multiple cues might be triggered, creating an annoyance. To mitigate such errors, we implemented a client-side filter for removing sensor noise.

We utilized a proximity reporting mechanism whereby a friend's nearby state is updated only after a number of consistent, consecutive readings. We originally considered using a straightforward approach of waiting until we detected 2 consistent consecutive readings (*2-same-filter*) or 3 consistent consecutive readings (*3-same-filter*) of near or far before updating a buddy's nearby state. In a pilot study, we found that *2-same-filter* helped reduce sensor noise, but still produced a number of false positives. In many situations, when buddies are near a distance corresponding to the ratio threshold of proximity, the ratio readings fluctuate between near and far quite a bit. As an improvement we decided to use a state machine approach, motivated by the 2-bit counters used by branch predictors in computer architecture [HP97]. Figure 3.7 illustrates the logic used for this approach. Buddies are initially reported as far away. Edge transitions represent a sensor sampling, yielding near or far. The state of a buddy is only updated to far or near when the states "Report Far" and "Report Near" are reached. This approach could potentially be applied to any binary decision-making sensor as long as it is accurate more than 50% of the time. Henceforth we call the 2-bit counter approach *2-bit-filter*.

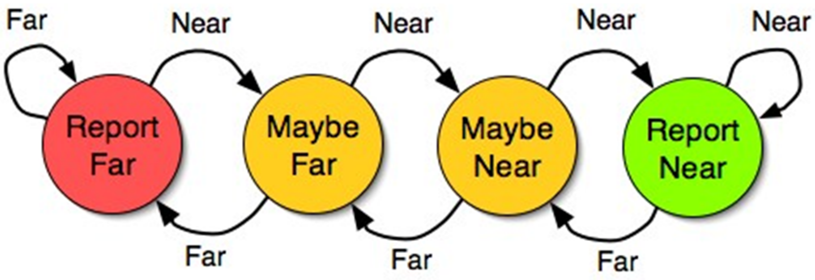


Figure 3.7: Two-bit counter for eliminating noise in proximity detection.

2-bit-filter attempts to improve upon these simpler algorithms. *3-same-filter* further reduces noise over *2-same-filter* but at the expense of added delay. In the worst case, *2-bit-filter* behaves like *3-same-filter*. However, *2-bit-filter* improves upon this approach because in all but the worst case, it has the responsiveness of a *2-same-filter* with the consistency of *3-same-filter*. As a result it is more robust than either of these two techniques.

To evaluate *2-bit-filter*, we compared its performance against *2-same-filter* and

3-same-filter. We also used a baseline condition whereby we would report near or far based on a single reading. For sensor noise filtering, we were interested in situations where users would be transitioning from far to near or vice versa. To extract these scenarios from the dataset, we extracted readings from the original dataset at 30s intervals. We then ran the three algorithms and the baseline on the resulting dataset.

Since all ratios showed a similar reduction in false positives, we report on the average reduction over the baseline. The usage of a *2-same-filter* was effective at reducing noise, reducing the average number of false positives by 53.8%. *3-same-filter* reduced false positives by 80.9%. *2-bit-filter* was most effective, reducing false positives by 84.9%. The reduced false positives translated into a higher precision, increasing by an average of 5% for a distance of 0.1km. This is quite significant considering the already high percentage of precision being reported in the previous section. Recall rates were improved by a negligible amount.

This analysis demonstrates that the *2-bit-filter* can be an effective technique for reducing sensor noise by maintaining recall while reducing the number of false positives. Not apparent in this analysis is the cost of using one of these filtering schemes, namely delayed proximity detection, which we address in the next section.

3.3.4 Minimizing Power Consumption

The limited power supply from mobile phone batteries requires careful consideration in a continuously running context-aware system. We addressed this by adjusting the sample rate and by minimizing unnecessary transmissions over the data network.

Sampling cell towers quickly often did not yield a change in seen cell towers, suggesting we could reduce the sample rate to save power, without dramatically affecting proximity detection response time. To get an idea of how sampling rate would affect power consumption, we originally chose a sample rate of 1 sample/20s. This caused the phone battery to discharge in less than a day, unfeasible for a study in the wild. To address this we decreased the sample rate to 1 sample/90s which turned out to be sufficient for a study in the wild, only requiring a recharge every other day. However, the usage of the 2-bit counter described in the previous section introduced a potential delay of 3 sample periods, 270s at a sample rate of 1 sample/90s.

To reduce the delay of proximity detection, an adaptive sampling rate was used. Initially, buddy proximity is sampled at a rate of 1 sample/90s. When the counter moves into a “maybe” state, the sampling rate is increased to 1 sample/20s, until steady state is reached (either “Report Far” or “Report Near”), at which point the sampling interval reverts to 1 sample/20s. This approach mitigates the delay of proximity detection reducing it to approximately 1.4 times the original sample rate. Initial data collection suggests *2-bit-filter* used in conjunction with an adaptive sampling rate provides good filtering of noisy data while reducing the delay of proximity detection. To avoid redundant notifications for buddies hovering around the near/far cutoff, the cues for a pair of buddies are delivered at least an hour and a half apart.

Our use of two sampling rates helps reduce power consumption to some degree, but measures are needed for situations with poor network signal. For one, sending data over a poor link tends to consume more battery power, in part because these data transmissions are more likely to fail, causing the underlying system to continue to attempt sending. Two, a poor link is indicative that there are no cell towers that are strongly suggestive of the phone’s relative location, so making a report provides no information about the phone’s whereabouts. These black hole situations are common in the USA, such as inside buildings with lots of metal or concrete. The PeopleTones client detects these situations by comparing the phone’s signal strength to a threshold. To compensate for when clients in these situations do not update, the server retains the last reported reading from the phone along with a timestamp, so that others can still make inferences about proximity for a while, assuming that the non-reporting of their buddies is caused by being in a building.

To measure power consumption, we timed how long it took to drain a fully charged iMate SP3i (HTC Tornado) running PeopleTones with the considerations described above. It took 2 days and 16 minutes to fully discharge the battery. This time period was deemed sufficient, precluding the need for daily recharges.

3.4 Peripheral Cues in the Wild

For the purposes of this study, we made three considerations to keep cues unobtrusive:

- First and foremost, the cue should not invade the periphery.
- Second, when the cue is perceived, it should not be seen as inappropriate in any way, most notably by those for whom the cue is not intended—a matter of etiquette.
- Third, because the periphery is constantly shifting with one's attention, perhaps as demanded by other changes in the environment (e.g., someone speaks to you, or shifting traffic conditions while driving), the cues, when perceived, should not be distracting—they should not impede shifts in attention or other natural changes to the periphery. In particular, people should not have to think about the cues that they are perceiving.

We refer to these three properties collectively as *unobtrusiveness*. With these issues in mind, the principal challenge with the use of peripheral cues in the mobile setting is resolving the tension between reliable receipt of cues and unobtrusiveness, without making unrealistic assumptions such as the required use of headsets. Peripheral cues can be overlooked without harm, and as designers we can err on the side of cues being missed.

With these considerations in mind, we decided to use a ratio threshold of 0.4 for our peripheral cue application. Based on our findings reported earlier, this provided good precision while maintaining fair recall. Our pilot studies confirmed this was an effective ratio for the deployment area.

We hoped to gain insight on these complex considerations over the course of our study, but we did have some initial hypotheses. One, short audio cues would be less invasive and more polite than long cues. Two, having corresponding vibration cues could be useful both for politeness and increasing chances of being perceived in noisy environments. Three, environment sensing could support the adaptation of the cues being played to ensure consistent maintenance of peripherality and politeness.

3.4.1 Auditory Cues

Since much past work with peripheral cue systems has used sound cues to deliver information, we followed in suit. Playing sound cues from a mobile phone is natural, but has potentially different requirements than environmental-based systems. Past work has found that short, rich auditory cues that build off of sounds users are accustomed to hearing in their normal lives can provide information to users serendipitously [MBWF97]. We explored a number of different types of sound cues. Soothing nature ecologies have often been used and so we created a set of nature cues of 3–5 seconds in duration. Music cues were also explored given that music timbres are effective for conveying information [BWE93]. Many mobile phones have the ability to map specific ringtones or music clips to different users on a contact list. While many people use these, the efficacy of mapping sound clips to identity is relatively unexplored. Yet, for a buddy proximity application, music clips seem promising for mapping the identity of a person to an audio cue, given the possibility of a semantic link [Gav89].

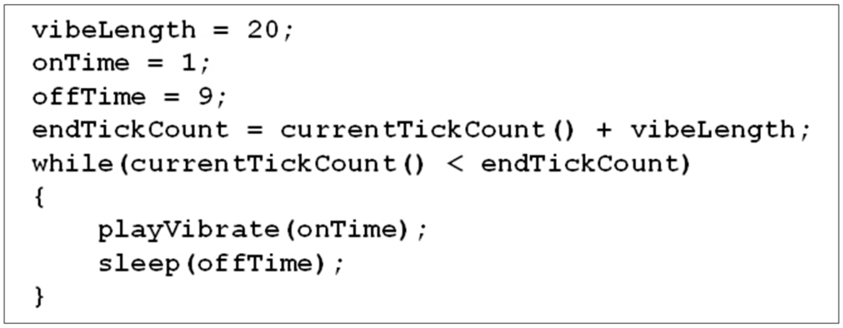
3.4.2 Vibrotactile Cues

In many office setting studies of peripheral cues, a headset or other wearable device is often the delivery mechanism used for delivering auditory cues in an etiquette-friendly manner. When delivering peripheral cues in the wild, where the user can be in a variety of social settings, it is unreasonable to require them to wear an additional device for receiving auditory cues. Mobile phones offer the ability to play sounds using their speakers, which can be effective for informal situations, but it is unlikely that this delivery channel will always be socially acceptable. Much like the silent or vibrate-only modes on mobile phones, peripheral cues delivered via these devices must also have a socially etiquette friendly mode [HLR01].

Motivated by haptics research suggestions to use vibrotactile cues for ambient information delivery [PMR02], we explored using vibrotactile patterns to convey ambient information on mobile phones with the actuator that commonly ships with these devices. Ideally, there would be a one-to-one mapping of sound cues to vibrotactile patterns, where a user could easily identify a vibrotactile cue and its respective auditory cue. However, generating a variety of distinguishable vibrotactile cues can be difficult on commodity mobile phones, given the limited API; most mobile phones only support the functionality of turning the actuator on or off. With the exception of phones with specialized built-in hardware [Imm], the API for most phones does not support playing vibrotactile pulses of different amplitudes nor do they provide any low-level functionality to specify the amount of current used to drive these actuators.

Generating Different Vibration Levels Using Mobile Phone Actuators

We present an algorithm to generate a wider range of vibrotactile sequences that circumvents API constraints on actuator functionality. While a full analysis of the capabilities of this approach is outside the scope of this paper, the basic algorithm for playing a pulse of varying amplitude is presented below for completeness. By changing the duty cycle[2] of the voltage sent to the motor, different speeds can be obtained. Similar techniques are used to reduce the power consumption of DC motors. This approach also reduces motor speed, making it useful for our goal of modulating the level of vibration. Our software approach repeatedly turns the actuator on for short periods of time, spinning between calls to the function that turns the actuator on. Timing is critical during this process, so the active thread is given the highest priority to avoid inopportune context-switches. By doing so, we demonstrate that we can achieve pulse-width modulation [3] via software. Different amplitudes can be generated by varying duty cycle.



```
vibeLength = 20;
onTime = 1;
offTime = 9;
endTickCount = currentTickCount() + vibeLength;
while(currentTickCount() < endTickCount)
{
    playVibrate(onTime);
    sleep(offTime);
}
```

Figure 3.8: Code for generating a 20ms vibrotactile pulse.

Figure 3.8 shows a code segment for this process. The `playVibrate` function represents the standard function for turning the actuator on, supplied by almost all mobile phone APIs. The variables `onTime` and `offTime` control the amount of time the actuator is turned on and off respectively. We demonstrate that by changing the values

---

[2] Duty cycle refers to the proportion of time that the device is turned on.

[3] Pulse-width modulation refers to the modulation of duty cycle.

of these, we can adjust the duty cycle of the vibrotactile actuator, changing the level of vibration generated.

A series of pilot studies found that people could not detect pulses played for less than 20ms in this manner and suggested the operating range could be divided into 10 differentiable *levels*, sufficient for this study. To generate a 20ms pulse level of 1, values `onTime=1`, `offTime=9` are used. To generate a 20ms pulse level of 9, `onTime=1`, `offTime=1` values are used. A pulse level of 10 is generated by calling the `playVibrate` function for the desired pulse length. Using this approach, a vibrotactile pattern can be defined as a sequence of such pulses of varying level.

To examine the effects of our software approach on the actual hardware, we opened a phone and measured the voltage levels produced by our software using an oscilloscope. Figure 3.9 shows sample oscilloscope plots measured for 4 different amplitudes. These plots confirm that our algorithm successfully achieves pulse-width modulation and that the different levels of vibration produced are the result of this.

3.4.3 Mapping Sounds to Vibrotactile Patterns

With peripheral cues deployed in the wild, a number of situations will arise where auditory cues will be socially disruptive (e.g. during a meeting) or might not be heard over ambient noise (e.g. walking by a busy street). We generated vibrotactile cues as a complement to auditory ones, hoping to leverage the association of auditory cue and buddy identity. If a vibrotactile pattern can be generated such that users can match it to its corresponding audio cue, then users can map the vibrotactile pattern to the buddy cue as well. This would reduce the need for learning a vibrotactile language.

Mapping auditory cues to vibrotactile sequences is challenging. On the one hand, there are difficulties associated with trying to map from an auditory system to a tactile one, where different receptors are being used to receive information [WWB+86]. This issue becomes further complicated by the significant differences in sample rates. Our pilot studies found that participants had difficulty differentiating between vibrations separated by less than 20ms. This generates a signal with fidelity equivalent to a signal sampled at 50Hz. A typical music file is sampled at 44.1kHz, a full three orders of magnitude greater, capable of capturing far more fidelity. To address this gross level

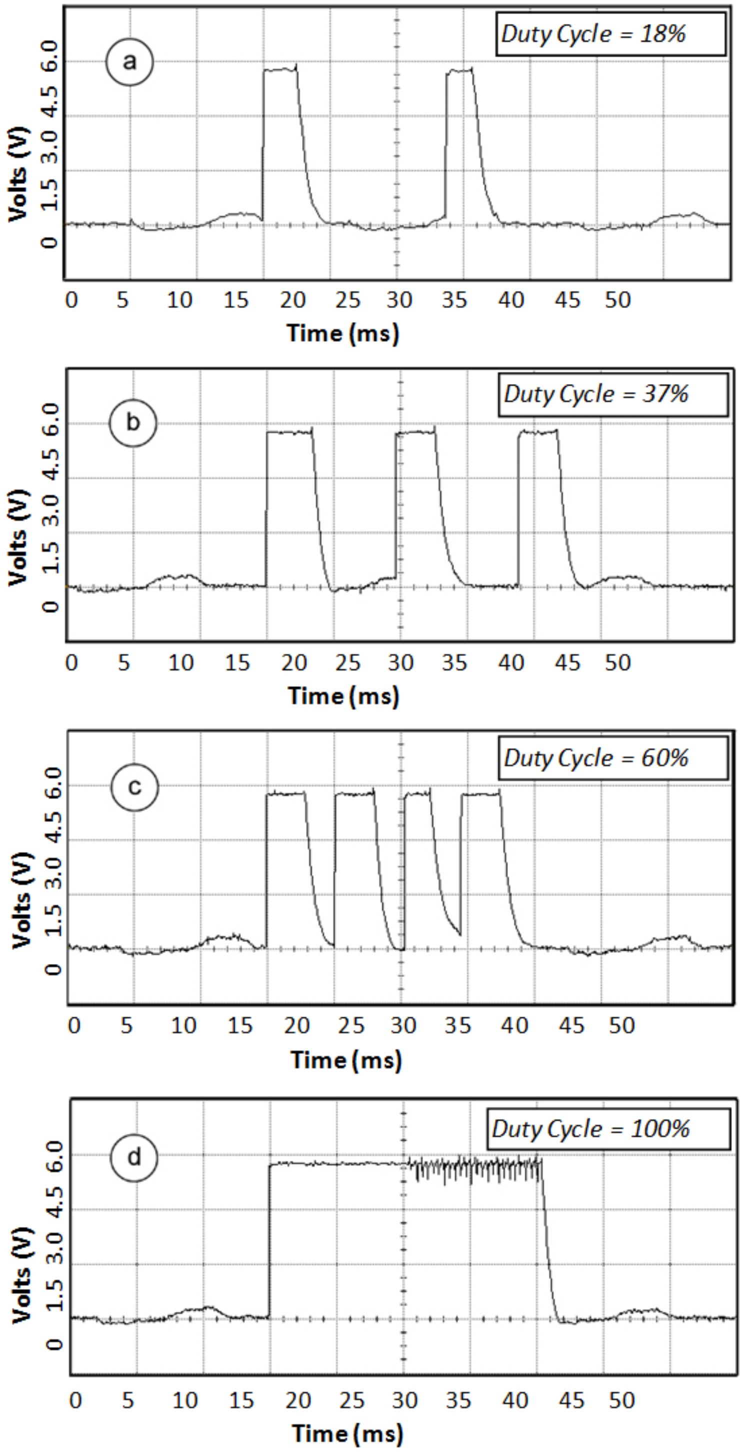


Figure 3.9: Oscilloscope plots of voltage generated for vibrotactile actuator for pulses at levels (a) 1 (b) 3 (c) 5 and (d) 10.

of under sampling, we utilized a number of digital signal processing techniques as part of the encoding process to try to capture the essence of the sound. We used a semi-automated method for converting a song to its vibrotactile equivalent using Matlab on a desktop PC.

Capturing the essence of a song is a known hard problem [BLEW04]. Initially, we considered the beat of the sound by examining lower frequency components of the clip. While this can be effective for certain sound clips, our experience suggests that it is because of the higher amplitudes of the low frequency components in those sequences. The lyrics of the song chorus were also thought to be important to characterize, given their use in identifying songs. However, in practice, lyrics are difficult to map to our vibrotactile language due its lower fidelity.

Pilot studies suggested that a combination of amplitude thresholding and band-pass filters would be the most promising approach. While lyrics are important in recognizing songs aurally, they are in practice difficult to map to vibrations. Instead, we aimed at mapping the beat of the song to vibrotactile patterns. We also found that by exaggerating the difference between loud and quiet sounds, the song was better characterized. The general process can be thought of as trying to create a humming sequence for the audio clip. Figure 3.10 outlines the general steps of this process.

The first step in converting a sound file into a vibrotactile pattern is to remove noise from the original signal. In this context, we consider “noise” to be elements of the sound that are not significant to the vibrotactile encoding of the sound, in addition to the traditional definition of the term. Our pilot studies found that components of the signal falling between the frequencies 6.6kHz to 17.6kHz were a good balance between noise reduction and keeping the original signal. We used an 8^{th} order implementation of the Butterworth Filter (a commonly used filter for bandpass filtering [HVV98]) to isolate the components of the signal in this frequency range (Figure 3.10-Remove Noise/Apply Filters). Additionally, we use an amplitude threshold to remove components from the output of the bandpass-filtering step. We only keep components that are greater than the average of the output.

The next step in the process is to try to characterize the resulting processed signal in a way that preserves the characteristics of the sound file. To do so, we take a running

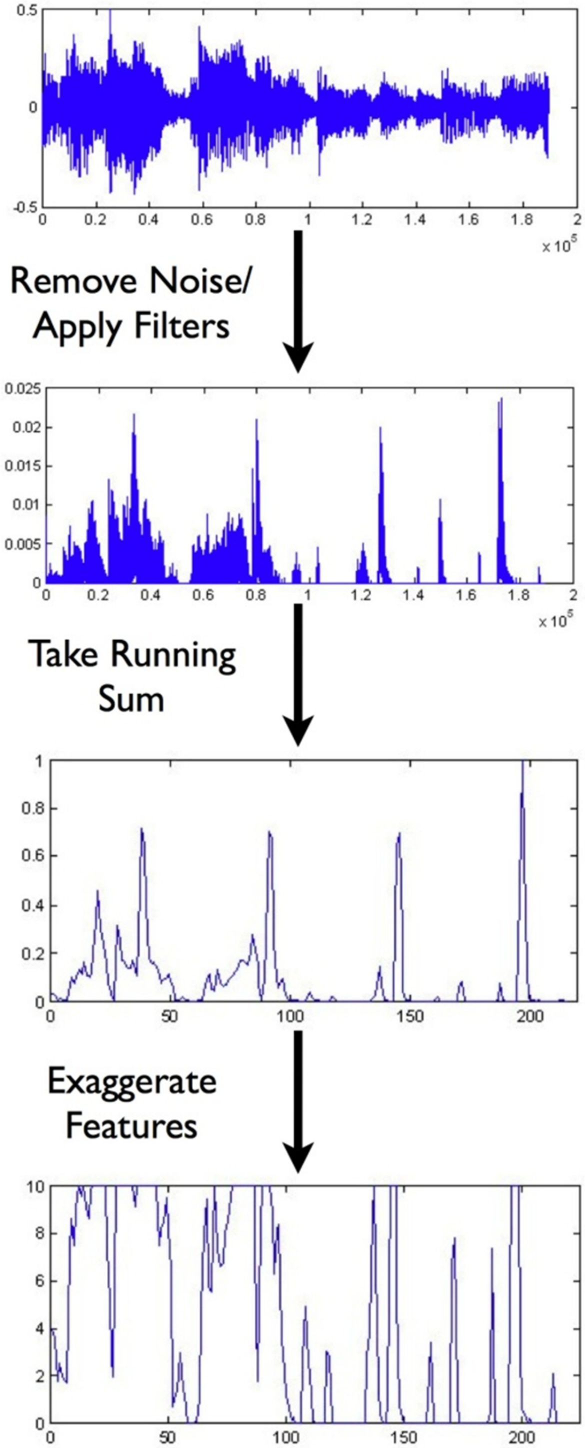


Figure 3.10: Block diagram showing the process of converting a WAV file to a vibrotactile pattern.

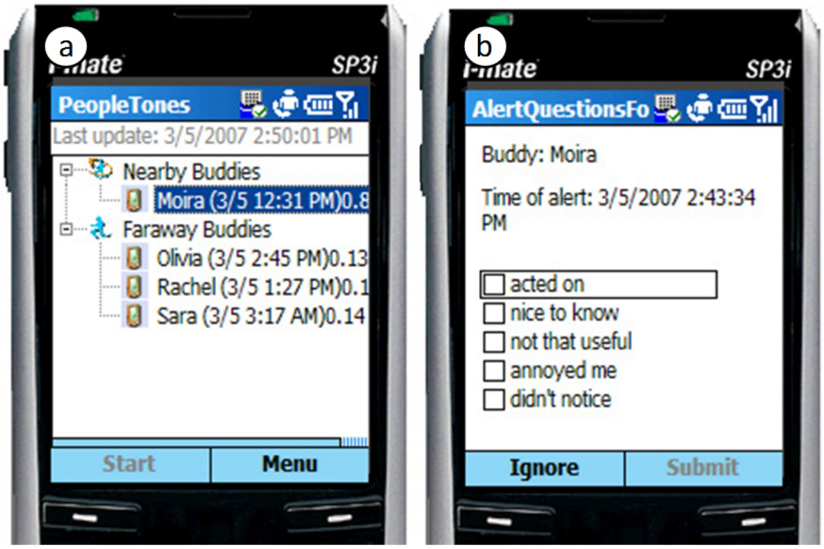
sum of the absolute values from the output from the previous step, generating 1 value for every 20ms (Figure 3.10-Take Running Sum). Each sample now represents a value that can be played for 20ms while keeping length of vibration and length of sound clip consistent. Finally, the differences between loud and quiet components of the signal need to be exaggerated (Figure 3.10-Exaggerate Features). We do this by composing the output from the previous step with a power function of the form Ax^n where x is the sample value and A and n are constants in the ranges: $10 \leq A \geq 15$, $1 \leq n \leq 2$. Part of the reason this is currently semi-automated is because we used different constants for different songs. Generally speaking, we used larger values of n when there was a larger range of frequencies in the original sound, and smaller values of A when the signal was louder. The result is a sequence of values representing a vibrotactile pattern that preserves many of the characteristics of the original sound signal.

3.5 PeopleTones

To validate the system level components we described above and to explore peripheral cues in the wild, we developed PeopleTones, an application for informing users of buddy proximity via peripheral cues from their mobile phones. A sound clip and corresponding vibrotactile pattern is associated with each buddy.

To inform the user of a buddy's proximity, the user can specify to have only vibrotactile cues, only audio cues, or both be played by selecting the appropriate phone profile. Alternatively, the user can enable an automatic noise detection mode to select an appropriate form of delivery. On the one hand, sound cues delivered in the middle of a meeting can be disruptive; on the other hand a sound cue delivered in the middle of a loud concert would be futile. Based on prior research, we knew that detecting interruptibility on a mobile phone would be impractical at best [SS00], but detecting the noise level in an environment using the phone's microphone to adjust the "level" of the cue is practicable. At the minimum, we hoped to learn how people would react to a mechanism for automatically choosing the mode of cue delivery.

When a cue was triggered to be delivered, ambient noise level was measured for 5 seconds and its average amplitude computed. In quiet environments, only the vibrotactile cues were played. In loud environments, both vibrotactile and sound cues were played, as vibrotactile cues can be felt in noisy environments when even loud sound cues might be inaudible. Quiet and loud thresholds were calibrated using both a quiet office environment and the loud student center of a University during a busy hour.



Last update: 3/5/2007 2:50:01 PM

- Nearby Buddies
  - Moira (3/5 12:31 PM)0.8
- Faraway Buddies
  - Olivia (3/5 2:45 PM)0.13
  - Rachel (3/5 1:27 PM)0.1
  - Sara (3/5 3:17 AM)0.14

Buddy: Moira

Time of alert: 3/5/2007 2:43:34 PM

☐ acted on
☐ nice to know
☐ not that useful
☐ annoyed me
☐ didn't notice

Figure 3.11: (a) The main PeopleTones user interface showing buddies that are near and far, as well as the last time the system was updated. (b) Post-alert prompt asking participants how useful the alert was.

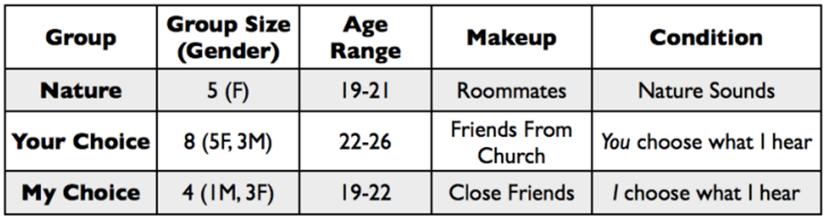
PeopleTones is implemented as a client-server application using standard SOAP web services. The client-side application is written in C#.NET on the Windows Mobile Smartphone platform. The interface is shown in Figure 3.11a. Each phone periodically pushes its GSM cell tower readings to the server, which computes buddy proximities and then notifies the phones of changes using the techniques described earlier. In situations where a client does not send the server an update of their location, the location uses a timestamp along with a copy of the last update received from that user. While all the users in our study used the same adaptive rate, this architecture allows clients to update their location as often as they like, allowing them to make their own power considerations.

3.6 User Study

We performed a naturalistic study by deploying PeopleTones to three groups of friends, forming three different test conditions, each for the course of two weeks. The purpose of this study was two-fold. First, the results from previous sections suggest that *proximity_ratio* when used in conjunction with *2-bit-filter*, should have high precision and modest recall. Yet, we wanted to test the hypothesis that, due to people lingering at places where they work, live, and play, that our participants would experience both high precision and high recall. Second, we sought to understand how peripheral cues worked in the wild, especially as regards obtrusiveness, comprehensibility, and the behaviors that resulted from their use. Thus, the three conditions were varied by the kind of peripheral cues that were employed.

3.7 Participants

We recruited three groups of friends forming groups of sizes 4, 5 and 8 people. These 17 participants consisted of students and young working professionals, 12 women and 5 men, aged 19–26. Participants were recruited based on interest in a buddy proximity application as well as having physically proximal friends. Participants were given an American Express Gift Card as a thank you for their time.



Group	**Group Size (Gender)**	**Age Range**	**Makeup**	**Condition**
Nature	5 (F)	19-21	Roommates	Nature Sounds
Your Choice	8 (5F, 3M)	22-26	Friends From Church	*You* choose what I hear
My Choice	4 (1M, 3F)	19-22	Close Friends	*I* choose what I hear

Table 3.4: Group makeup for the three groups.

3.7.1 Methodology

Participants used PeopleTones over the course of 2 weeks. We conducted 4 interviews over the course of the study. Prior to the study, a pre-study interview was conducted to gather basic demographic information, mobile phone usage habits and general "closeness" to the participant's friends whom were also participating in the study. Additionally, a pre-study was conducted to evaluate whether participants could match the semi-automatically generated vibrotactile patterns to sound clips. A mid-study evaluation was also conducted to make sure there were no problems with the system. Finally, a post-study interview was conducted to reflect on the participant's experience. A test of matching vibrotactile patterns to music cues was again performed to measure learning effects, if any, that may have taken place over the course of the study, and to evaluate consistency.

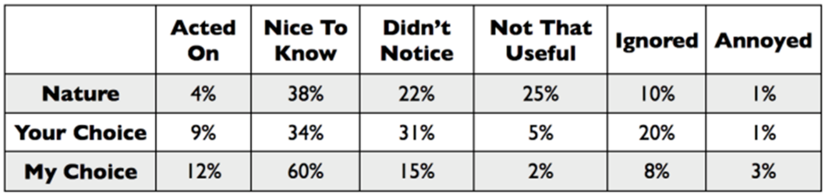
The three groups of friends formed three different conditions for cue-to-information mapping methods. Group Nature (N) consisted of 5 friends who were given a set of nature sounds to assign to their friends. However, they opted for automatic assignment of cues, since they felt there was no relationship between the cue and their friends. This eliminated the need for a full 2×2 study whereby a group of users who selected their own Nature cues would have been included. Group Your Choice (YC) consisted of 8 friends who selected a single sound for themselves, representing the cue that their friends would hear when they were nearby. Group My Choice (MC) consisted of 4 friends who each selected the cues that they would hear, when their friends were nearby. Table 3.4 summarizes these group conditions.

Before the study, participants identified music cues that they would want to use for the study. They were given the option to identify specific parts of the song that they wanted to use. Alternatively, participants could select from 2–3 different 3–5 second segments of the song selected by the authors, typically chosen for their mapability to vibrotactile patterns. After participants selected song segments they wanted to use, a corresponding vibrotactile pattern was generated, using the procedure described in section 3.4.

3.8 Usage and Self-reported Data

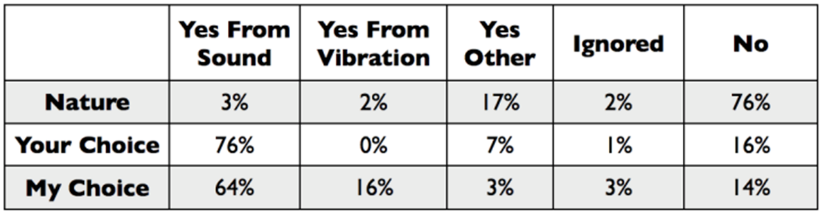
To perform data analysis, we performed client-side logging when a cue was triggered. Once a cue was triggered, the participant was presented with a form asking them if they acted on it, if it was nice to know, if it was not useful, or if it was annoying (Figure 3.11b). Alternatively, they had the option to ignore the form if they were busy. This post-cue questionnaire was left on the screen so they could later respond. If they did not choose to ignore the cue, they were presented with a form asking them if they could tell who it was to which they could respond “yes from sound” “yes from vibration “yes, other” or no. In this case, “yes, other” was intended for situations where the participant knew who it was based on other factors (e.g. knew their roommate was coming home around that time) This was recorded for completeness and not factored into the comprehension rates described below. Forms were left on the screen until they were responded to.

A total of 683 cues were sent over the course of 2 weeks, across all conditions with 122 cues in the Nature group, 466 cues in the Your Choice group, and 95 cues in the My Choice group. Each cue resulted in a post-alert form being displayed. Using self-reported forms displayed on the mobile phone, the user was queried both for their response to the cue as well as whether they could identify the buddy that the cue represented. The breakdown for these post-cue responses is shown in Table 3.5 and Table 3.6 respectively. Since all cues elicited a response form and all forms required a response (even if the response was “Ignore”), the percentages are also reflective of the 683 total.



	Acted On	**Nice To Know**	**Didn’t Notice**	**Not That Useful**	**Ignored**	**Annoyed**
Nature	4%	38%	22%	25%	10%	1%
Your Choice	9%	34%	31%	5%	20%	1%
My Choice	12%	60%	15%	2%	8%	3%

Table 3.5: Self-reported response to the cue.



	Yes From Sound	Yes From Vibration	Yes Other	Ignored	No
Nature	3%	2%	17%	2%	76%
Your Choice	76%	0%	7%	1%	16%
My Choice	64%	16%	3%	3%	14%

Table 3.6: Self-reported identification of the cue's information.

3.9 Discussion

In the following section we reflect on our two major research questions: the suitability of peripheral cues as an in-the-wild communication mechanism and the suitability of mobile phones for providing such cues. We draw on the observations and data above, as well as from our interviews with the study participants.

3.9.1 Peripheral Cues are a Viable Communication Mechanism in the Wild

Unobtrusive peripheral cues in the wild, while challenging, can be achieved by informed cue design and providing personal control over cueing mechanisms.

Designing and Choosing Cues for the Wild: Music and Personal Control

Although office-setting studies have found soothing nature ecologies to be effective for comprehension and unobtrusiveness, cues in the wild should be composed of music, and perhaps repeated.

With regards to comprehension, the self-reported usage data shows that groups Your Choice (YC) and My Choice (MC) both demonstrated an 83% comprehension rate, where comprehension is defined to be when the user could identify the buddy from the cue, collapsing results from audio and vibration (Table 3.6). In contrast, the nature group demonstrated a significantly lower rate of 22%. Interestingly, this lower rate did not result in lower usefulness ratings (42%) for the application when compared to the Your Choice group (43%). Perhaps the ability to look at the phone after receiving a cue mitigated the negative effects of cue comprehension. Many participants cited that they would prefer longer cues since they could be difficult to catch in the dynamic environments of their daily lives. For example, <MC-1> commented: "sometimes couldn't hear because the song was too short."

The obtrusiveness of music cues was not a concern. The reasons are somewhat surprising. <MC-3> comments: "When it went off in [the library] it didn't actually seem to annoy other people too much, they just thought it was just another phone." This observation points to the fact that mobile phones have become largely invisible and socially accepted, at least for young adults, even in a "quiet zone" like a library. (We reflect more on etiquette concerns in the next section.) Another reason cited for the unobtrusiveness of music cues was the positive feelings generated by the music. <MC-1> comments: "I would like longer songs so I could hear it and because I like the songs." The Your Choice group made similar comments, even though they did not pick their music cues. <YC-5> comments that she liked: "Just hearing the songs. I liked the fact that each person could choose whatever they want for their own identity. Since it was a small group of us, it's kind of fun and it felt like this is a group of us." Overall, 9 of the 12 music participants volunteered a liking for hearing music. Interestingly, it appears that cues with emotionally positive associations are generally unobtrusive.

Music cues are similar to the ringtones sometimes used for caller ID on mobile phones. However, ringtones in the wild are relatively unexplored. <YC-3> comments: "It was fun how everyone had a song specific to them. Adds a little bit of personality. I do not use ringtones, that's why it was neat for me. Too much trouble to do on my phone." The results of this study validate the usefulness of ringtones in being able to successfully convey information about people in a pleasurable way.

The usage of music cues also seems to reinforce learnability, with 83% of users in conditions My Choice and Your Choice being able to identify who the cue was for, based on self-reported post-notification questions. This learning effect is also reflected to some extent with vibrotactile patterns. The My Choice group was the only condition with an appreciable amount of cue identification from vibration, demonstrating 25% identification rate from vibration cues alone. While not overwhelming, this acts as a proof-of-concept for the delivery of ambient information via low fidelity haptic channels. Analysis of the before and after vibration studies suggest some users are consistent in the way they match vibrotactile patterns to sound, with 7 participants responding consistently, when comparing their before and after responses. 75% of My Choice was able to correctly map vibrations to sounds and then to people. Participants in the Your Choice condition were less successful in mapping vibrotactile patterns to music, possibly because of the larger number of cues or because they were not as familiar with the songs selected for cues. When presented with the task of matching vibrations to sounds in the post-study interview, participant <MC-3> exclaimed “Oh that’s Cathy!” when she felt the vibration associated with the music cue associated with Cathy. When comparing error rates for a matching vibration-to-sound task from before and after studies, minimal improvements were observed, suggesting minimal learning effects.

This does not necessarily suggest that music cues should always be used over nature cues. Rather, it reinforces the idea that when designing peripheral cues, it is important to maintain a semantic link between the cue and the content being delivered. Nature cues could be useful if they had some meaning when associated with the object of interest. In the case of buddy peripheral cues, participants were much more likely to have semantic associations between music and friends.

Personal Control over Cueing Mechanism for Unobtrusiveness

The discussion above suggests that personal selection of cues aids both comprehension and unobtrusiveness. In addition, for many users, explicit control of the notification modes was important. Although personal control has been cited as important in the design of a number of social mobile systems, these concerns typically have to do with privacy [ISC$^+$05]. In our study, personal control over the cueing mechanism was a critical element for controlling unobtrusiveness and interruptibility.

Even though an automatic ambient noise level feature was provided, many users opted not to use this mode, not even trying it before dismissing it. In fact, 12 of the 17 participants did not even try the Automatic mode, despite the fact that the phone was put in Automatic mode when given to the participants. When asked about the automatic mode in the post-study questionnaire, <MC-4> commented “I didn’t use it. I was afraid to use it since my professors this quarter are pretty anal. I kept it mostly on vibrate when I was in class, or in normal mode when I wasn't." For this participant, personal control of the notification mode was important because they feared PeopleTone cues being triggered in the audio mode in a classroom setting where it might be disruptive. Participant <YC-1> expressed a similar concern, saying "I was afraid that if I was at church, it wouldn't work. It would just backfire on me and I wanted to have been more sure about it." Like <MC-4>, <YC-1> was afraid of unwanted notifications while at church, another social context where an audio notification would be unacceptable. It should be noted that both <MC-4> and <YC-1> considered social contexts where they expected the notifications to be triggered, in this case as defined by their group's shared interests. In addition to a lack of trust in the application's accuracy for detecting ambient noise in high stakes situations, a number of users also expressed uncertainty as to what the system considered to be loud or quiet environments. <YC-4> said "I'm not too sure what happened or how loud the environment needed to be. I'd want to determine how reliable the function is before using it and to check how often it looks at the environment." <YC-4>'s comment suggests a potential solution to this problem is for some type of system feedback whereby through manually user-controlled notification management, the application gains the user's trust, demonstrating that it works well in a variety of environments. This could be done with some type of visual indicator in the application of the type of cue that would be delivered, which the user could check while in different environments. Of course, this solution requires the user's visual attention during the familiarization period.

Still, even with a trustworthy system in place, some users had different requirements than those retained by the system. <MC-2> commented "Sometimes when it's quiet, I don't need it to be quiet, like when I'm at home by myself. I think I felt it one time but I generally like to keep it on [loud]." Although the user trusts the Automatic mode to work as expected, she prefers the mode that plays both vibrotactile and audio cues, simply because in certain quiet situations, auditory cues will not be irritating to anyone.

Finally, for some participants, different notification modes offered different fidelities of information. <YC-8> commented that "Did use the vibrations, but didn't work out well. I felt it vibrate, but I could indicate [who it was] better with sound. The sound lets me instantly figure out who it is. With the vibrate, you have to wait 5–10 seconds to figure out who it is." Although this comment touches on the issue of learning mappings between the different notification modes, <YC-8> expresses a distinct preference for sound cues, crediting their higher fidelity. Similarly, <Nature-1> said "I wish it had given me louder alerts," suggesting that some mechanism for controlling the volume would have been useful for some.

User Information Need: Peripheral Cues Provide an Overview

When our participants were asked about how physically near friends needed to be to be considered "nearby," many people cited the mode of transportation as being relevant. For people traveling by foot, distances within about 0.5 miles were cited as being "nearby." For people traveling by car, a distance of 2–5 miles was considered nearby. Our results from Figure 3.5 show that our algorithm was able to achieve precise detection within these limits. Additionally, 15 of 17 participants reported that PeopleTones' implementation of near was good enough for buddy proximity, verifying the results from the dataset analysis.

The proximity algorithm used by PeopleTones, which detected proximity at around 0.2 miles, was accurate enough for user needs. When asked about the accuracy of the system, most participants commented that it detected "near" for a buddy most of the time when the buddy was known to be near, and "far" when far away. Activity detection [SVL$^+$06] could conceivably be used to adjust the cutoffs according to one's speed of movement, although the accuracy reported by users suggests that this may not be necessary.

Even though PeopleTones' proximity algorithm was deemed accurate enough, 4 people spontaneously volunteered that they also wanted to know the actual distances of their buddies, and 3 of those wanted the location. "I'd like if it could tell me exactly how close they actually are," said <MC-2>. <YC-5> offered that "The only thing I didn't like about using this phone was not knowing exactly where that person is." (PeopleTones's proximity ratio is not suitable for computing exact distances or locations, but ratios significantly above the "near" cutoff could be used to infer "close".) This information need is not surprising, since such information would inform, for example, a decision on whether to call the person to arrange a meeting. This information need not be provided by the cue itself. Schneiderman's Visual-Information Seeking Mantra "Overview first, zoom and filter, then details-on-demand" [Shn96] suggests that the peripheral cue should be treated as the overview, with additional information displayed in the user interface providing details on demand. Also, because the cue is an ephemeral overview, and may not be fully comprehended, the visual interface provides valuable redundancy. For this reason, the PeopleTones user interface used by the participants displayed the near/far state of the buddy and the time of the inference. <MC-4> volunteered "If I wanted to know if anyone was nearby, I liked how it showed the last time it had checked next to their name."

At the same time, displaying distance could be perceived as an invasion of privacy. Surprisingly, only 2 of 17 users reported privacy as a concern. This could be because the groups of friends were tightly knit, or because exact distance was not shown. The 2 participants concerned with privacy suggested providing user control of who could see their location.

3.9.2 Mobile Phones Are a Viable Platform for Context-Aware Peripheral Cue Applications

As the above results convey, mobile phones appear to be a viable platform for proactive context awareness when there are asymmetric tradeoffs to be leveraged. In the case of an application like PeopleTones, missing an event of interest, whether due to sensing or notification, is acceptable. There were few reports of false negatives and only one report of a socially problematic cue. There were also few reports of false negatives.

Yet, several other factors unrelated to conservative design contribute to the viability of peripherals cues on phones. Mobile phones and the sounds they make are socially accepted in many settings, aiding unobtrusiveness. As personal devices they enable personal control, which aids the comprehension of cues, creates the positive associations that permit managing the periphery, and ensures their unobtrusiveness when silence is paramount.

Additionally, with the use of our novel DSP techniques, the commodity vibration actuators found on mobile phones are an adequate channel for etiquette-sensitive situations. For music-based vibrations, people were very good at matching the vibration patterns to their songs. One group, the My Choice group, found the vibrations to be useful in the wild, serving as the delivery mechanism about 16% of the time.

Likewise, several factors unrelated to conservative design contribute to the viability of context sensing on mobile phones. Direct phone-to-phone comparison of cell tower readings not only achieves ubiquity but also avoids a possible source of error by not calibrating to a third frame of reference (absolute location). Emphasizing the elimination of false positives is apparently effective because the lingering of buddies eventually leads to successful recall. Timeliness is not a critical feature of buddy proximity, but some users did complain about the occasional slowness of the reporting, suggesting that dwelling in a place eventually leads to a positive report.

As corroborating evidence, the participants told many stories about how PeopleTones affected their behaviors or dispositions. Here are a few typical quotes, at most one per participant:

"One time at the library, I wanted to eat with someone and so I went outside to call someone. The phone vibrated. I just called the person to meet up."

"Whenever I drive to school I found out where <YC-7> works because I always get her alert when I'm driving on Miramar. Oh, so she works around here?"

"I thought it was so neat every time it would ring. It made me really happy. Oh! They're right here, or oh! They're right there."

"It was cool to see who was home by the time I got home. I could tell if <YC-1> was home when I passed by University. So if we were going to go eat or something I could ask her. Oh she's home, so let's call her and see if she wants to eat."

3.10 Conclusion

Employing mobile phones for proactive context awareness holds promise due to the ubiquity of mobiles and their infrastructure, yet phones' necessarily inexpensive construction presents challenges like imprecise sensors, clumsy actuators, and limited battery life. For the case of detecting and reporting on "nice to know" situations such as the proximity of a friend, the precision of sensing must be high enough to minimize annoying false notifications, and the notifications cannot be too obtrusive to the user or those in the vicinity. We explored these issues through the PeopleTones buddy proximity application.

We have contributed (1) an algorithm for detecting proximity, (2) techniques for reducing sensor noise and power consumption, and (3) a method for generating unobtrusive peripheral cues.

For detecting proximity we compared the cell towers seen by the mobile phones to estimate proximity. GSM's long range and random characteristics means that a phone will report false positives, so we filter the proximity data using a simple state machine. With these techniques we are able to achieve 99% precision for a ratio threshold of 0.4 and fair recall. The counter also manages power consumption by sampling at a slow rate when the state machine is in the typical far or near states. Power is further conserved by not reporting when the GSM signals are weak.

We took a peripheral cue approach to providing notifications, using both short audio and corresponding etiquette-friendly vibrotactile cues. To achieve a language of corresponding vibrotactile cues, we introduced an offline digital signal processing technique that captures the essence of audio cues, whose patterns are realized on the phone by generating range of amplitudes using a technique similar to pulse-width modulation.

Our controlled studies of proximity detection based on wardriving data revealed high precision, especially with the 2-bit counter, but only modest recall. In real world settings, where people dwell at locations for significant periods, recall appears to be much higher because the algorithm has more chances to detect proximity.

The user study revealed that peripheral cues are an effective, unobtrusive mechanism for notifying people of such inferences. Although haptics have often been suggested as a promising ambient delivery mechanism, sound was the preferred medium, possibly because of its higher fidelity. Our method for encoding sounds into vibration patterns on the limited vibration motors of mobile phones produces a representation of sound that is sensible to many, but not all people. An underlying theme of the study is the importance of personal control for peripheral cues. Peripheral cues in the wild are better comprehended and less obtrusive if derived from music and are chosen by the intended recipient. Moreover, people have an overriding need to directly control the modality of cue delivery to manage etiquette. Context-adaptive cueing requires support and mechanisms for gaining a person's trust. Peripheral cues can provide a sparse overview of the underlying situation, but the ability to get details on demand is important to users, especially since the cues are ephemeral and sometimes not understood.

We conclude that despite the challenges presented by appropriating commodity sensors and actuators, that mobile phones are a suitable platform for proactive context awareness, at least for the "nice to know" case. Likewise, peripheral cues are a viable notification modality on mobile phones, despite their simple on/off actuators.

Despite the success of mapping music to vibrotactile cues, not all stimuli maps well to vibration. Although music mapped well to vibration, other forms of stimuli such as human touch do not map well to vibrotactile cues. Additionally, not all types of information has strong association with music cues. In the next chapter, we explore the possibilities of using voice coil motors as an actuator. These have higher fidelity than the offset motors commonly used for vibration. We examine how these actuators can be used to recreate tapping and rubbing from interpersonal communication.

3.11 Acknowledgments

This chapter, in part, is a reprint of the material as it appears in PeopleTones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones. Li, K.A., Sohn, T.Y., Huang, S., Griswold, W.G. In Proceedings of the International Conference on Mobile Systems, Applications and Services (Mobisys 2008), pp. 160–173. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Tapping and Rubbing: An Exploration in Creating Computer Mediated Human Touch

As demonstrated in the previous chapter, music is a very powerful stimuli that many people already associate with certain semantics. We demonstrated how we could take advantage of these types of associations by assigning music to buddy identity in a buddy proximity application. By mapping the songs to vibrotactile cues, we were able to preserve the cue-to-information associations. When users felt the vibration patterns, they thought of the buddy associated with the corresponding music cue. Thus, we were able to demonstrate a way to convert semantics associated with music, to the tactile channel. However, not all types of information map well to music. For example, there is much information encoded in physical touch exchanged between people in interpersonal communication. This type of information would be difficult if not impossible to capture with music. Thus, we need a more expressive form of touch for encoding this type of information. Given the limitations of commodity vibrotactile actuators, we need to apply new types of actuators to solve this problem. This chapter examines how human touch might be replicated via computer-mediated mechanisms.[1]

---

[1] Note that this chapter is a reprint with minor changes of *Tapping and Rubbing: Exploring New Dimensions of Tactile Feedback with Voice Coil Motors* co-authored by Kevin Li, Patrick Baudisch, William Griswold and James Hollan.

4.1 Introduction

Vibrotactile feedback has been widely employed for eyes-free communication, which is particularly valuable in mobile scenarios. When auditory feedback is socially inappropriate [HLR01] or used for other cues, vibrotactile feedback can be the best or even the only channel that allows a device to communicate with the user [HB07].

However, current implementations of vibrotactile feedback are limited. Vibrotactile feedback can convey a variety of signals, but these are generally perceived as conveying urgency. While this is appropriate for alerting users, it might be less appropriate for notifying users about a non-urgent, enjoyable event, such as the receipt of a text message from a close friend. It seems particularly inappropriate if the tactile ring is the message, such as when trying to communicate “I am thinking of you” over a messaging system.

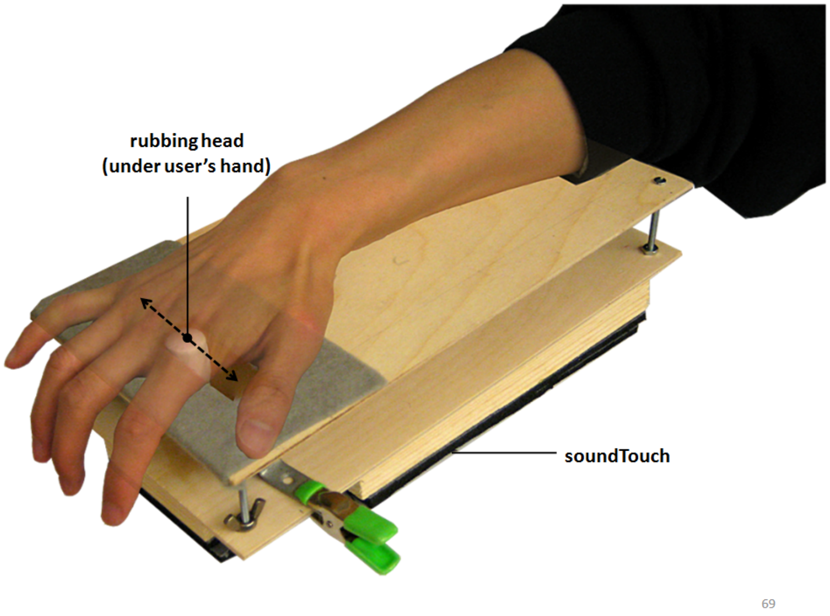


Figure 4.1: Rubbing interaction implemented using our soundTouch prototype.

We propose extending the haptic vocabulary of notification and messaging devices with tactile messages inspired by human-human communication. We make two contributions:

1.We introduce two new types of haptic feedback, *tapping* and *rubbing*. These are modeled after their human-human counterparts and designed to convey attention and comfort, rather than urgency.

2.We report the results of a user study demonstrating (a) that users indeed perceive the above modalities as tapping and rubbing as experienced in human interaction and (b) that users can distinguish a wide range of tapping and rubbing frequencies and amplitudes.

The proposed techniques cannot be implemented with traditional vibrotactile method, because these methods cannot produce sufficiently low frequencies. Below 20Hz, the offset (eccentric) DC motors used in these devices can no longer produce noticeable displacements. To implement rubbing and tapping we therefore developed a haptic device that we call *soundTouch*. SoundTouch uses a voice coil motor from a computer hard drive to sidestep the mechanical limitations of traditional vibrotactile devices. As a result, soundTouch supports a large space of tactile designs inaccessible with traditional vibrotactile methods.

In the following, we give an overview of the related work, introduce soundTouch, and describe how we implemented tapping and rubbing. We then present two exploratory studies on the quantitative and qualitative expressiveness of tapping and rubbing. We close with a discussion of our findings.

4.2 Related Work

We draw on three areas of related work: force feedback, haptics in HCI and applications of hard drive actuator technology.

4.2.1 Force Feedback

Force feedback offers a large range of tactile sensations with the goal of mimicking real world experience, and is often used in virtual reality environments. One approach commonly used in haptic gloves is to use an auxiliary system of actuators with pulleys and cables to provide force feedback [cyb]. Pneumatics have been proposed to reduce the size of the pulleys but still require a wearable device [BBPB02].

Salisbury's *Phantom* uses a similar approach to create visual haptics whereby a user can feel a space by holding a stylus connected to a rig of actuators [MS94]. Sensors detect the orientation of the user's finger and the rig generates the appropriate force feedback. These approaches can be effective for desktop scenarios, but require users to hold the stylus to get the feedback.

4.2.2 Haptics in HCI

Our work is guided by a large body of work in psychophysics. Studies on skin sensitivity have found fingers and hands to be more sensitive than thighs and arms [CC00]. Cutaneous sensitivity is generally accepted to be logarithmic in nature, both for the detection of pressure as well as the resolution of frequency [WWB$^+$86].

Hayward and MacLean present a good introduction to haptics [HM07]. The following projects highlight some of the technologies being used to create haptic interfaces.

The most widespread technology is the offset motor used to generate vibrotactile feedback in mobile phones and game controllers. Despite the aforementioned limitaions of the technology, researchers have been able to generate a variety of uses for vibrotactile feedback. Li et al. developed a technique similar to pulse-width modulation that generates on the order of 10 different amplitudes of vibration [LSHG08].

The *C2 Tactor* uses an alternative approach, generating vibration by moving a small contact head via a voice coil actuator [Tac]. Brown and Brewster have done a significant amount of work with the C2 Tactor showing how a variety of haptic icons can be generated by modulating waveform and location [BB04a, BBP06, BK06]. Chang uses a similar approach with *Multifunction Transducers* that allows a single actuator to be used for vibration and audio [CO05].

Haptics has also been proposed as a way of allowing users to communicate with one another. *HandJive* explored how users would communicate with a haptic input/output device using force-feedback [FCAE98] while Chang's *ComTouch* explored how users would communicate with one-another using vibration [COJ$^+$02]. Both employed an unstructured approach that resulted in an arbitrary abstract language

Poupyrouv's *AmbientTouch* uses layers of piezoelectric to generate vibrotactile feedback in PDAs [PMR02]. Luk implemented an array of piezoelectric tabs to generate lateral skin stretch, allowing different waveforms to be felt under the thumb [LPL$^+$06]. Lee's *Haptic Pen* used a solenoid to mimic the feeling of pressing down with a stylus [LDL$^+$04].

Rubbing and tapping have been proposed as input mechanisms for interacting with touch screens [OFH08] and with synthesized surfaces[MSWHQ08], but not as forms of feedback.

4.2.3 Applications of Hard Drive Actuator Technology

Hard drive actuators are attractive for their low cost, small size, and resilience. They have been used in biomedical telerobots to provide combined actuation and force sensing [MH91]. In subsequent work, hard drive technologies were used in multi-fingertip haptic displays for detecting surface variation in virtual and telepresence environments [VH99]. A similar multifinger display has been studied for its information transmission characteristics, employing three-dimensional taps and vibrations [TDRR99]. In a very different direction, hard drive motors were used to create a force feedback controller for steering and experiencing music [VGM02].

4.3 SoundTouch

Figure 4.2 shows our soundTouch prototype—it forms the basis for a series of tactile interfaces we have created. The prototype consists of a voice coil motor extracted from a disk drive. It is connected to the audio out jack of a notebook computer. The notebook computer delivers a sound signal that soundTouch converts to motion similar to the way a speaker converts an electrical signal into audible sound. Between the audio out and the voice coil motor is a custom amplifier circuit board that amplifies the 150mVPP of the audio-out jack to the 12V required by the motor (based on *Analog Devices AD815AYS*. Figure 4.3 and Figure 4.4 show the schematics we used for our amplifier circuit. To give a sense of the voltage waveform presented by our approach, Figure 4.5 is a plot of the voltage waveform produced by soundTouch to generate a tap of level 7.

The key element is the voice coil motor that we extracted from a regular hard

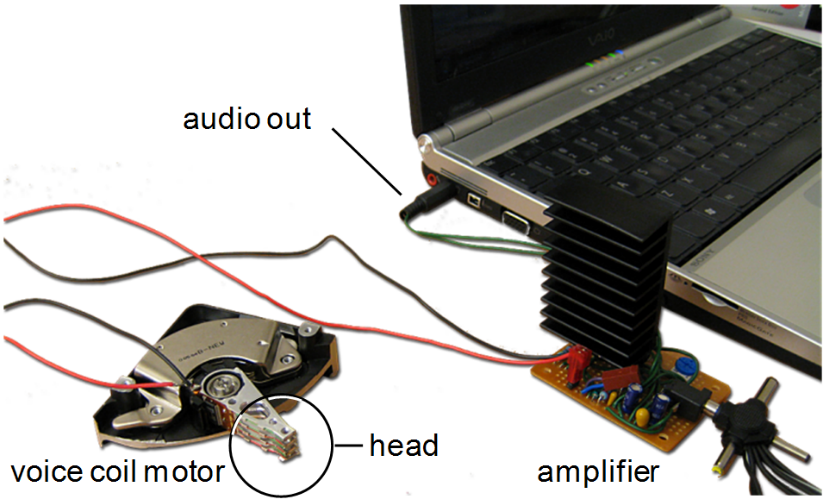


Figure 4.2: Our *soundTouch* prototype translates sound signals into tactile feedback.

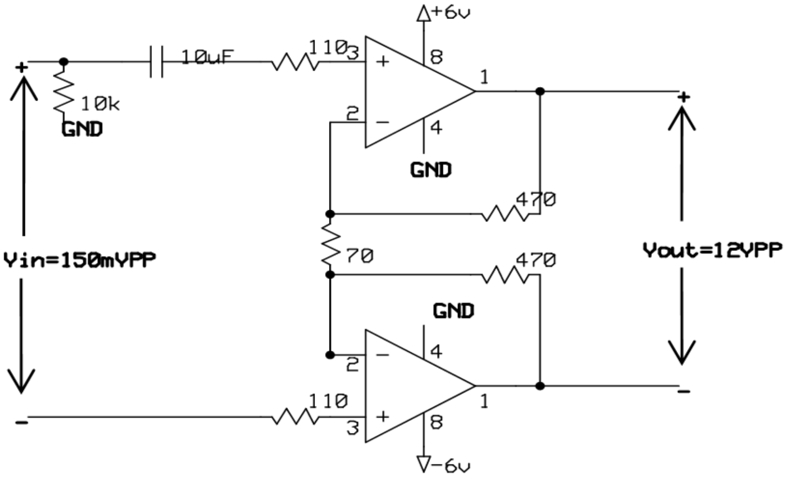


Figure 4.3: Amplifier circuit schematic.

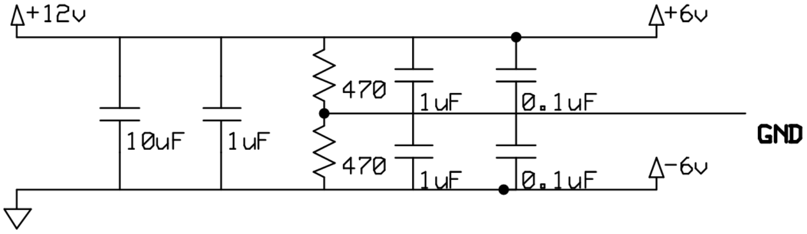


Figure 4.4: Voltage Divider to create +6V and −6V power rails for two amplifiers.

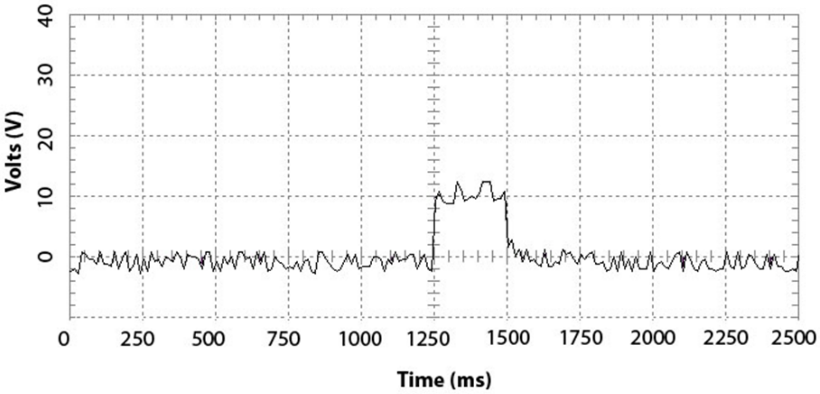


Figure 4.5: Voltage waveform produced by soundTouch to generate a tap with amplitude of level 7.

disk drive (a 3.5inch *Western Digital*). Figure 4.6a shows a close-up of the voice coil motor. Applying a voltage to the device actuates the coils, which rotates the arm. When creating tactile interfaces based on soundTouch, we attach covers with different tactile qualities to the head of the drive to create different tactile effects when it comes in contact with the skin (Figure 4.9).

4.3.1 Features

By feeding it a sound file, soundTouch can be manipulated freely, i.e., it can play back an arbitrary signal rather than, say, just a signal of a single frequency. In particular, it allows us to perform very coarse as well as very fine motions and any combination thereof. SoundTouch can produce actuations orders of magnitude below the audible range, i.e., $<<$ 20Hz. Additionally, soundTouch can move the head to a particular location at configurable speed.

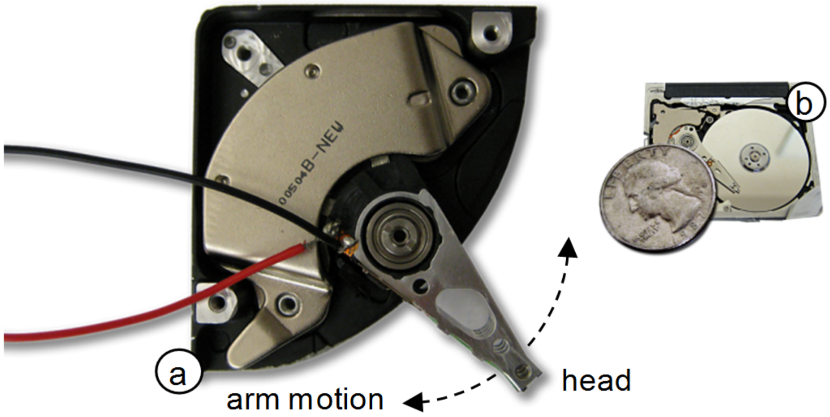


Figure 4.6: (a) Closeup of the voice coil motor in soundTouch (b) micro drive next to a US quarter.

At the same time soundTouch can perform fast and delicate motions when fed a high frequency signal. For the sake of illustration, we have developed a demo application that makes soundTouch play back audio files, including WAV and MP3 music files. SoundTouch can reproduce frequencies considerably outside the range relevant for tactile feedback (15 kHz and potentially higher). If the played signal contains frequencies in the audible range, then the device will vibrate audibly, basically functioning as a speaker.

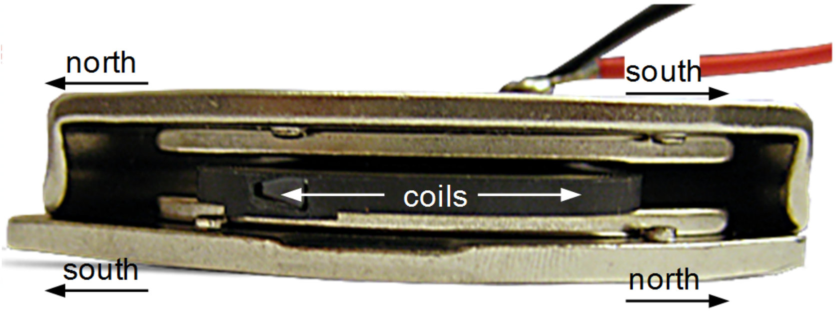


Figure 4.7: Side view of a voice coil motor.

Unlike the voice-coil-like motors used in the C2 Tactor [Tac], the hard drive actuator in soundTouch moves the coils instead of the magnet. Because the coils are lighter than the magnet, soundTouch can generate greater acceleration of the armature with less voltage. The hard drive motor in soundTouch uses a sandwich of two bipolar magnets (Figure 4.7), further increasing force.

The voice coil motor in our current prototype measures 5.5cm x 3.6cm. Future versions may achieve form factors suitable for mobile applications by using the mechanics from a smaller hard drive, such as an IBM micro drive (Figure 4.6b). Customized designs can generate even higher forces [LKDH05].

4.4 Tapping and Rubbing

We have built two tactile interfaces based on soundTouch. Both interfaces are designed to emulate common human-human touch gestures, namely *tapping* and *rubbing*.

4.4.1 Tapping

Figure 4.8 shows our *tapping* prototype. We created it by attaching a wooden “hammer” to the head of our soundTouch prototype. By driving the device with signals in the range around 1Hz, the device produces a tapping motion.

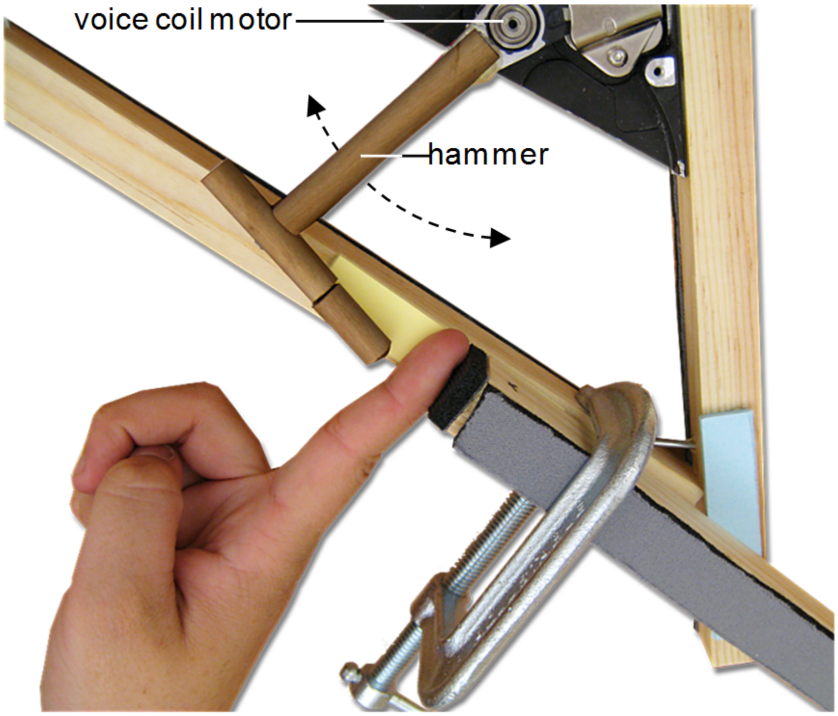


Figure 4.8: Tapping prototype: a hammer attached to the head of the soundTouch prototype taps on the user's finger.

The arm is 8.1cm long and has an angular displacement of 30°, resulting in a linear displacement of 3.4cm. A layer of foam at the bottom of the device reduces noise and structural vibrations.

We explored a number of materials for the hammer head including: a rubber eraser, a trackpointer tip, rubber cement, a toothbrush, glue, a paper clip, cotton, styrofoam, epoxy, wax, a sponge, a rubber band, and a foam earplug. Figure 4.9 shows some of them. The main factor impacting the experience was whether the material was deforming (cotton, foam) or non-deforming (rubber, epoxy), with little subjective difference within each of the two classes. Since the experience with deforming materials changed over time, eventually degrading to feeling like a non-deforming material, we ended up using a non-deformable head and chose the most durable one of them: an epoxy glue dot.

To quantify the range of forces generated by the tapping prototype, we mounted

Figure 4.9: Some of the materials we have used as hammer heads.

a force sensor (a *Measurement Specialties FC22*) perpendicular to the motion of the contact head and measured static force generated for voltages at 0.5V increments in the range 0V-12V. The force generated by the voice coil motor we used is characterized ($r^2 = 0.977$) by a linear regression: $F = 0.101V - 0.83$, where F is force in Newton and V is voltage in Volts.

4.4.2 Rubbing

Figure 4.1 shows our rubbing interface. As shown in Figure 4.10, rubbing is achieved by moving the head tangential to the user's hand, so this interface is literally "orthogonal" to the tapping interface.

During piloting, the head occasionally got caught on the edges of the user's hand. To address this, we used a window limiting the contact area to the participant's hand size (Figure 4.11a). Two clamps and repelling magnets on the sidewalls of the window limited lateral motion. These two fixes eliminated the problem.

An initial prototype used a shorter arm (8cm) as shown in Figure 4.11b. We eventually replaced it with the longer arm shown in Figure 4.10 (21cm) to obtain a longer rubbing motion (6.5cm). By mounting the head perpendicular to the plane of

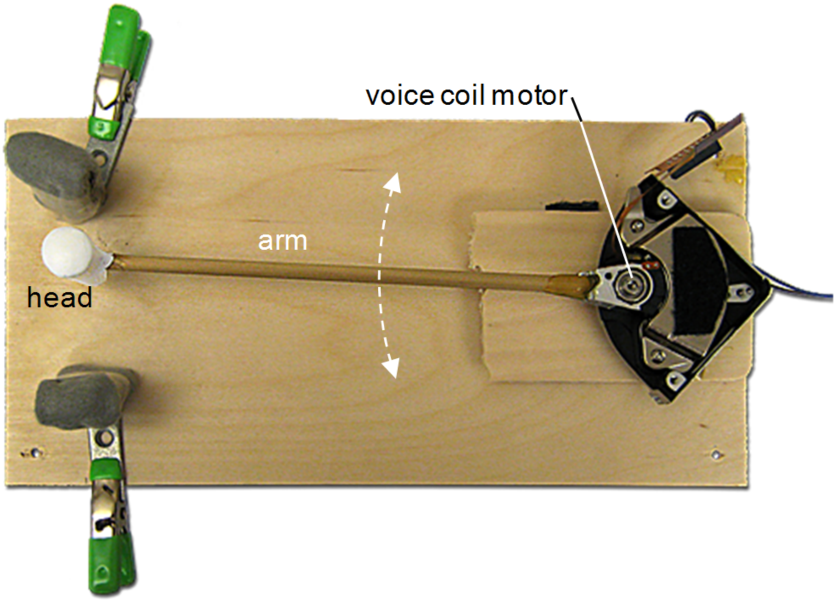


Figure 4.10: Rubbing prototype (cover removed).

motion, we obtained a very even rubbing motion. Repelling permanent magnets at the bottom of the contact head and underneath the device keep the head suspended and provide the desired pressure against the user's hand. The magnets effectively eliminate vertical torque forces on the soundTouch device.

For rubbing we explored a similar set of materials as with the tapping prototype. Unlike with the tapping prototype, the texture of the material used for the contact head significantly changed the rubbing experience. Many of the materials created a rough sensation that was uncomfortable or dragged on the skin too much. Smoother materials such as the glue dot felt too slick to elicit a rubbing experience. We ended up blending the two approaches by covering a smooth round surface with Teflon tape. This created a smoother surface than many of the materials we had tried earlier, and had an almost skin-like quality (Figure 4.12b).

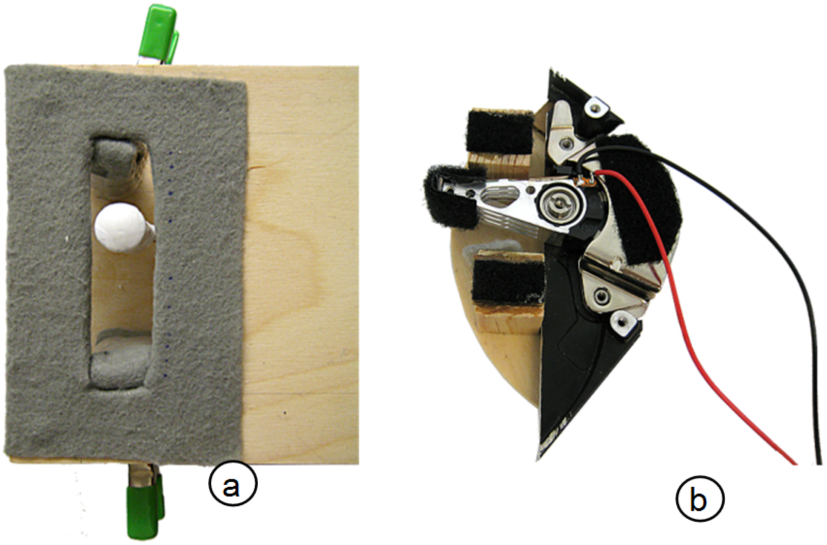


Figure 4.11: (a) Rubbing prototype with cover; only the head shows through (b) earlier prototype.

Figure 4.12: (a) Bare head and (b) covered in Teflon tape.

4.5 Scenarios

The primary motivation that inspired rubbing and tapping is to allow devices to extend the tactile vocabulary of devices. This is especially relevant when exchanging simple messages with close associates or family members. The use of a richer tactile vocabulary allows sending simple self-contained tactile messages, rather than requiring the combination of a generic vibration alert and a textual message.

For example, a haptic message could update others about common daily events (e.g. “I am leaving for home”), information that could be useful to communicate but not sufficiently important to merit a phone call.

Rubbing and tapping bear inherent associations with physical touch. This makes them particularly suited for messages that match the underlying connotations, such as reminders (tapping) and expressions of care and comfort (rubbing).

For the same reasons, the more expressive haptic vocabulary created by tapping and rubbing is well-suited for personal tactile ringtones for close friends and family members. Beyond this main scenario, we feel that a richer tactile vocabulary could be useful for the following situations.

Truly Silent Alerts

Vibrotactile alerts are intended to be unobtrusive, yet people in close proximity can often hear other people’s phone vibrating. A phone left on a desk can make a loud noise when vibrating. Rubbing, in contrast, is *silent* and can therefore be used for truly unobtrusive alerts. Notifications could be administered by a mobile device or, in a office setting, through the user’s chair.

Alerts Guaranteed to be Noticed

When users are on the move or in a noisy environment their audible ringtones might not be heard, vibrations not be felt. Escalating alerts via harder taps provides a means to deliver crucial alerts.

Game Controllers

Many popular game controllers use Immersion's vibrotactile technology in their rumble packs to augment the gaming experience [cyb]. Vibration is a good representation for some events, such as the user driving off the road or being shot at. For positive events, such as when picking up a health pack, a rubbing sensation might be better suited.

In-car Navigation

Car navigation systems use speech output to inform drivers where and when to turn, which can interfere with conversations with other passengers. Vibration alerts are easily missed, because cars tend to vibrate due to road irregularities. This limitation can be avoided by communicating turn directions or traffic events using tapping and rubbing, administered through actuators in the steering wheel or in the seat.

4.6 User Studies

We conducted two exploratory studies. Their purpose was to investigate how users perceive taps and rubs and how well users can distinguish different types of these signals. This would allow application designers to create messages out of sequences of taps and rubs.

4.7 Study 1: User Perceptions of Tapping

The first study investigated participants' perception of tapping. We varied *amplitude* and *frequency/number* of taps. We investigated whether users could *distinguish* and *identify* different amplitude and frequency levels.

4.7.1 Apparatus

The tapping device shown in Figure 4.8 was used to present taps to the participants' fingertips. For each stimulus condition, a stop guard was calibrated to each participant’s index finger. Sound waveforms were generated using a C++/C# program with DirectSound on a 2.0 GHz PC running *Windows Vista*.

4.7.2 Independent Variables

In the Amplitude condition, participants were presented taps of differing amplitudes. In the *Frequency* condition, participants were presented taps at differing frequencies (taps per second).

4.7.3 Task 1: Distinguish

When performing the *Distinguish* task, participants experienced a stimulus pair twice on each trial before making a forced-choice decision about which one felt stronger (*Amplitude* condition) or faster (*Frequency* condition). The interval between pair members was 1.0s and the interval between pairs was 2.0s.

The cues users can use to distinguish tap sequences depend on whether frequency or duration is kept constant. Varying tapping frequency leads to a different number of taps if duration is kept constant. As a pilot participant pointed out, this allowed participants to differentiate between the “slow ones” and the “really slow ones” by counting taps. Keeping the number of taps constant, in contrast, led to sequences of different lengths.

We explored both aspects. In the *Frequency* condition, half of the participants were presented stimuli of constant duration (*ConstantDuration*) while the other half of the subjects were presented with a constant number (*ConstantNumber*). Tap sequences were 3 taps long. The design was within-subjects for *Amplitude* and between subjects for Frequency.

All participants experienced the same sequences of stimuli. The order of stimuli presentation was pre-randomized. This allowed us to compare per trial performance across participants. For the *Distinguish* task, there were 3 blocks of 22 trials. Each block consisted of all pairs of stimuli differing by 1 or 2 levels over the 1–7 level range, as shown in Table 4.1. We considered pairs of stimuli differing by more than 2 levels, but pilot studies suggested these were fairly easy to differentiate and so we did not examine them in the formal study. Participants were given a 5 minute break between blocks.

We used 7 different amplitude levels evenly spaced from 0N to 1.0N and 7 different frequencies from 5Hz to 29Hz.

	1	2	3	4	5	6	7
1		x	x				
2	x		x	x			
3	x	x		x	x		
4		x	x		x	x	
5			x	x		x	x
6				x	x		x
7					x	x	

Table 4.1: Stimulus pairs differed by one or two levels. The resulting pairs are marked with an ×. Entries shaded in gray represent pairs that differed by two levels. Columns and rows denote level of first and second tap.

We measured error rate. For the distinguish task, a trial was considered an error if the participant identified the wrong stimulus as stronger.

4.7.4 Task 2: Identify

When performing the *Identify* task, participants were presented with the same stimulus twice, with an 0.5s interval in between, and asked to rate them on a 7-item Likert scale (1 = slow/soft ; 7 = fast/hard). The same 7 levels of amplitude and frequency used in Task 1 were used in Task 2.

One pre-randomized block of 49 trials was presented to participants. To avoid sequencing effects, the block consisted of a sequence of values such that both orderings of every pair of numbers from 1–7 appeared in the block.

For *Amplitude*, the contact head was placed 9° from the participant's fingertip, resulting in an arc length of 1.0cm. For *Frequency*, the contact head was positioned 4° from the participant's finger tip resulting in an arc length of 0.45cm. These distances were chosen based on pilot studies. The stimulus was generated using a 250ms square wave. Although the bandpass characteristics of the soundcard dampen the signal, a consistent tapping sensation can still be generated. Figure 4.5 shows a plot of the output signal. The presentation of the *Frequency* and *Amplitude* conditions was counterbalanced across participants.

The *Distinguish* task was always completed before the *Identify* task. This was done to give users an idea of the range of taps and rubs generated by the device, before asking them to rate taps on an absolute scale. Both tasks were completed for one stimulus condition before performing the other.

4.7.5 Questionnaire

For each condition, participants answered the questions "How would you describe the tactile sensations you just experienced to someone who has not experienced them?" and "Which aspects of the experience felt natural and which aspects did not?" Because we wanted to elicit how users naturally describe the tactile sensations they experienced during the study, the experimenters were careful not to mention the word "tapping" or other suggestive terms.

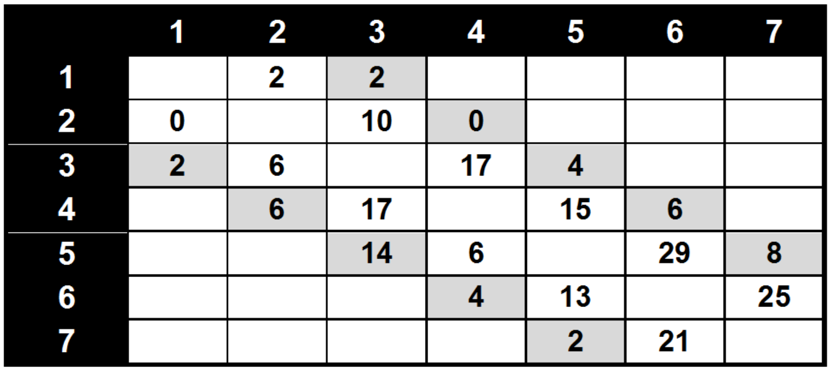
4.7.6 Participants

16 volunteers (8 female) ranging in age from 18–22 years (median 19) were recruited from within our institution. Participants received an American Express gift card as a gratuity for their time.

Two participants were left handed. Participants wore headphones playing pink noise from an MP3 player to eliminate ambient noise. Each participant took approximately 1 hour to complete the experiment.

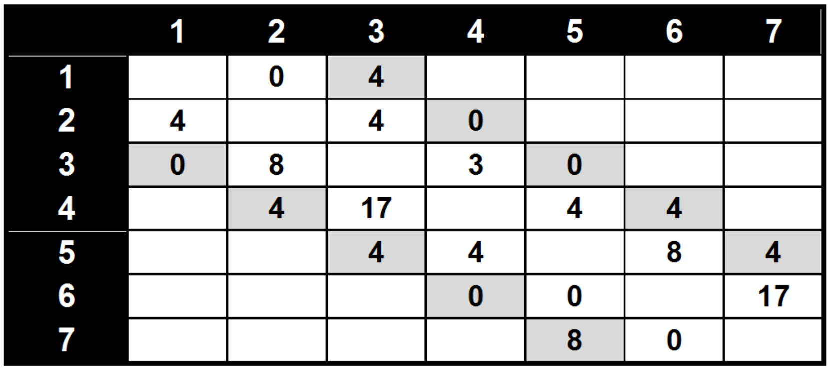
4.7.7 Results: Distinguish Task

For the *Distinguish* task, error percentages were aggregated over all participants for each ordered pair of stimuli for the *Amplitude, ConstantDuration* and *ConstantNumber* conditions (Tables 4.2, 4.3, and 4.4). A row/column pair represents the ordered pairs of levels for the stimuli presented.



	1	2	3	4	5	6	7
1		2	2				
2	0		10	0			
3	2	6		17	4		
4		6	17		15	6	
5			14	6		29	8
6				4	13		25
7					2	21	

Table 4.2: Tapping *AmplitudeDistinguish* error rate in % collapsed across all participants for *Differentiate* task when being presented a stimulus pair of intensity (<row>, <column>).



	1	2	3	4	5	6	7
1		0	4				
2	4		4	0			
3	0	8		3	0		
4		4	17		4	4	
5			4	4		8	4
6				0	0		17
7					8	0	

Table 4.3: Tapping *ConstantDuration Distinguish* error rate in % collapsed across all participants.

Post hoc multiple means comparisons showed no significant effects for block number on error rates, suggesting no learning effects. We aggregated errors for each participant for each of the stimulus conditions. Surprisingly, participants did not perform significantly different between *Frequency* and *Amplitude* conditions (t(15) = 0.66, *p*>0.05). Since we used the same trial sequences for both *Frequency* conditions, *ConstantDuration* and *ConstantNumber*, we were able to compare error rates between the groups. We aggregated errors for participants in each condition by trial number. Participants made significantly more errors in the *ConstantNumber* condition (t(65) = 3.338, *p*<0.001). As expected, participants performed significantly better on the *Distinguish* task for stimulus pairs that differed by 2 levels than those that differed by 1 level for all conditions (*Amplitude* t(16)=3.777, *p*<0.01, *ConstantDuration* t(8)=2.95, *p*<0.05 and *ConstantNumber* t(8)=2.084, *p*<0.05).

	1	2	3	4	5	6	7
1		0	4				
2	8		15	4			
3	0	0		13	8		
4		0	0		13	13	
5			4	13		46	29
6				8	13		63
7					8	17	

Table 4.4: Tapping *ConstantNumber Distinguish* error rate in % collapsed across all participants.

4.7.8 Results: Identify Task

Mean values of user reported levels for the *Identify* task are shown in Figure 4.13, Figure 4.14, and Figure 4.15. Post-hoc multiple means comparisons showed that users were able to identify the appropriate stimulus level for all levels and conditions with the exception of *ConstantNumber* for frequency levels 3 and 4.

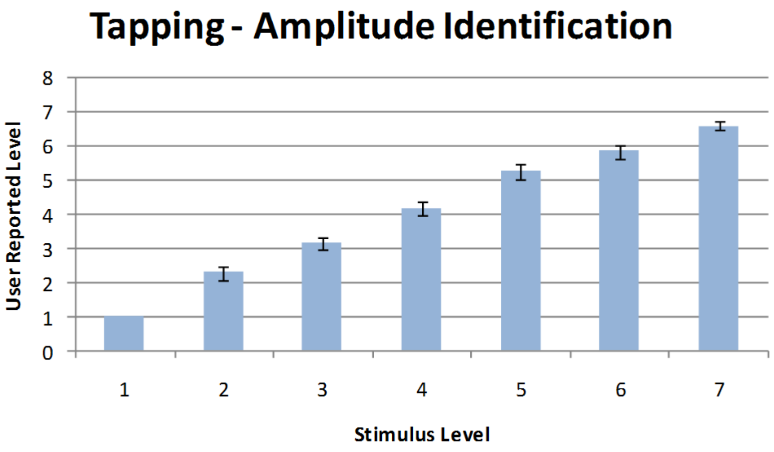


Figure 4.13: Mean values of user reported levels for the *Identify* task for the *Amplitude* stimulus condition. Error bars show 95% confidence intervals.

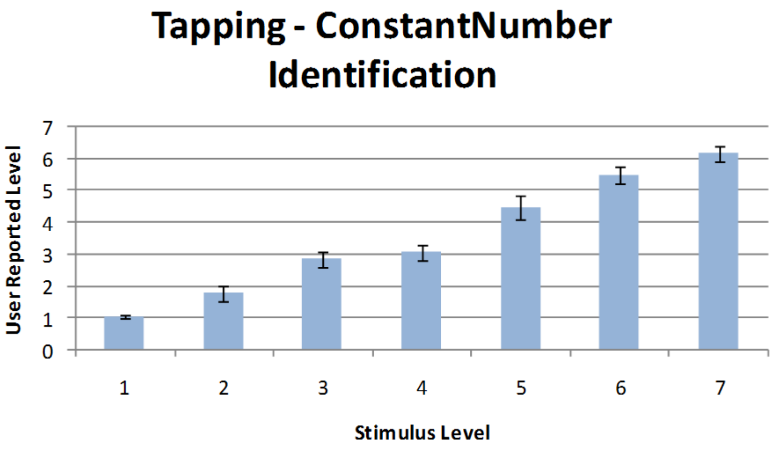


Figure 4.14: Mean values of user reported levels for the *Identify* task for the *Constant-Number* condition. Error bars show 95% confidence intervals.

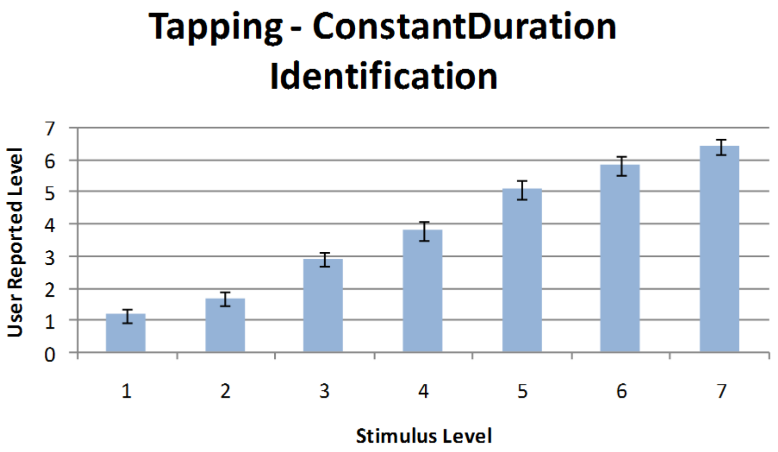


Figure 4.15: Mean values of user reported levels for the Identify task for the *Constant-Duration* condition. Error bars show 95% confidence intervals.

4.7.9 Results of the Questionnaire

When describing their perceptions, many participants used terminology drawn from human-human interaction. Thirteen of the 16 participants used the word *"tap"* in their descriptions. Additional descriptions included: "getting flicked on the finger", "tickling", "brushing something off", "drumming fingers" and "touch". Twelve participants volunteered that the experience had a human quality to it, often citing that it felt like "getting tapped on the shoulder, but on your finger". Fifteen participants indicated that the faster stimuli felt like vibrations from a mobile phone or game controller. Twelve participants mentioned that the harder taps did not feel "natural" and 5 said that the fast ones did not feel "natural".

When asking participants about potential usage scenarios where they would want to use the respective stimuli, six participants stated that single taps would be good for mobile phone alerts in quiet environments because of their silent nature. Seven participants thought they would be useful in situations in which they could not feel vibrations, as when outside or walking around.

4.8 Study 2: User Perceptions of Rubbing

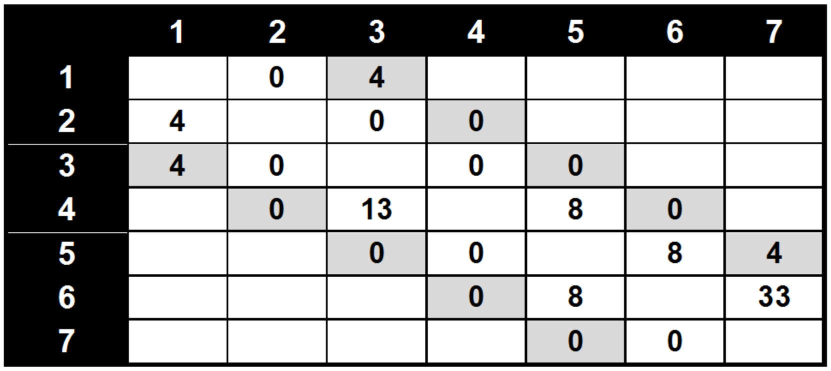
The purpose of the second study was to examine user perceptions of stimuli from the *rubbing* prototype.

Task and stimuli corresponded to those in Study 1, except for three differences. First, instead of a series of taps, participants were exposed to a series of rubs. The rubbing prototype shown in Figure 4.1 was used, adjusted to fit the respective participant's hand size. Second, there was no *Amplitude* condition since our pilots showed that the distance covered did not allow differentiation of differing amplitudes. Third, the *ConstantNumber* condition used 2 rubs instead of 3 taps.

The experiment was a within-subjects design for two Frequency conditions, *ConstantDuration* and *ConstantNumber*. Eight volunteers (6 male) from our institution between the ages of 18 and 26 participated in our study. All participants were right-handed.

4.8.1 Results: Distinguish Task

Post hoc multiple means comparisons showed no significant effects for block number on error rates, again suggesting no learning effects. To compare error rate for the two conditions, errors were aggregated across participants for each trial. Participants made significantly more errors on *ConstantNumber* than on *ConstantDuration* (t(66)=9.077, *p*<0.01).



	1	2	3	4	5	6	7
1		0	4				
2	4		0	0			
3	4	0		0	0		
4		0	13		8	0	
5			0	0		8	4
6				0	8		33
7					0	0	

Table 4.5: Rubbing *ConstantDuration Distinguish* error rate in % collapsed across all participants.

	1	2	3	4	5	6	7
1		7	4				
2	7		15	22			
3	7	12		22	15		
4		15	22		30	30	
5			15	30		48	33
6				22	15		56
7					15	41	

Table 4.6: Rubbing *ConstantNumber Distinguish* error rate in % collapsed across all participants.

As expected, participants performed significantly better on the *Distinguish* task for trials in which stimulus pairs differed by 2 levels than those that differed by 1 level in both the *ConstantDuration* condition (t(8)=2.528, $p<0.05$) and in the *ConstantNumber* condition (t(8)=2.828, $p<0.05$).

4.8.2 Results: Identify Task

Figure 4.16 and 4.17 show the results of the *Identify* task.

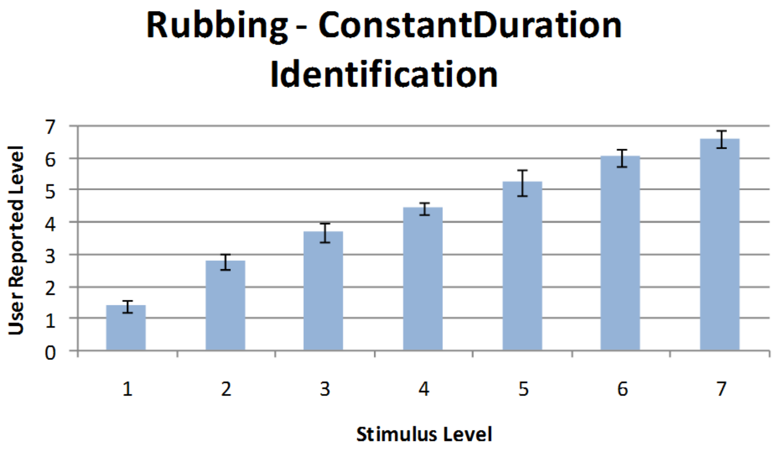


Figure 4.16: Mean values of user reported levels for the *Identify* task for the *ConstantDuration* condition. Error bars show 95% confidence intervals.

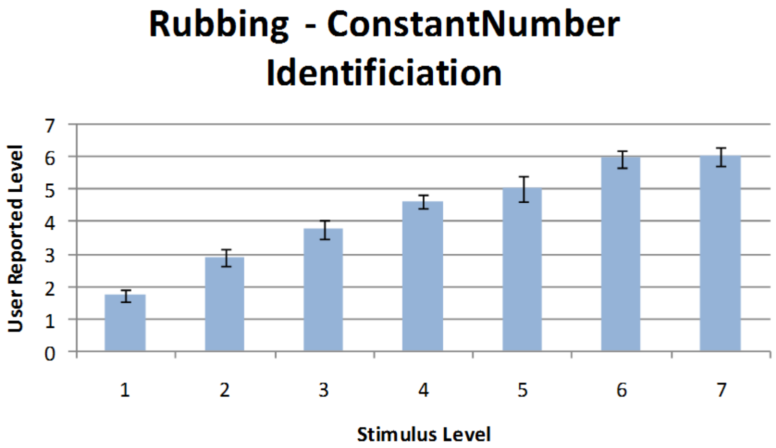


Figure 4.17: Mean values of user reported levels for the *Identify* task for the *ConstantRubs* condition. Error bars show 95% confidence intervals.

4.8.3 Results of the Questionnaire

Three of the eight participants volunteered that the experience felt like *"rubbing"*. Those that did not described it as *"grazing"* or a *"light sweeping"*. Four participants volunteered that it had a human-like quality to it as if someone else was touching them. One participant said *"It felt strangely comfortable, almost like the touch of someone else. It was more like a finger touching my skin than an object."*

Half of the participants felt that the faster rubs felt more natural while the other half thought the slower ones were more natural. Those who cited faster ones being more natural mentioned that it felt more like "sliding your hand across a table" or "dropping a marble through your hands". These participants said that for the slow ones, you could feel the actuator moving against the palm and could tell it was an artificial thing. Participants who said slower was more natural used comments like "I don't come across anything that moves that quickly" to describe their experiences. They also described the sensation as being more like "rubbing your hands together" or "running a cotton swap through my hand" or "playing with a rubber eraser".

When asked about usage on a mobile phone, comparisons between rubbing and vibration inevitably came up. Four participants volunteered that they would prefer this over vibration for truly silent scenarios where the sound from vibration would be annoying. Four participants suggested it would be better for in-the-hand tasks because it was less jarring than vibration.

Five participants described the *ConstantNumber* stimuli as feeling like they were rubbing or grazing an object against their hand while the *ConstantDuration* stimuli felt more like touching something that was moving (water, marble, bus handle, etc). One participant said the sensation caused by *ConstantNumber* stimuli "felt more like I'm shaking, whereas [*ConstantDuration*] seemed more like I'm holding onto something that's shaking".

4.9 Discussion and Design Implications

Our experiment provides some initial insight about how people experience stimuli generated by our soundTouch prototype. A few design implications also emerge. SoundTouch's tapping and rubbing mimic real world The participants consistently described their experiences with terms like tapping and rubbing and seemed to readily relate the experiences to common human-human interactions.

The softer taps were consistently reported as feeling natural. The naturalness of taps was tested for the hardest and fastest ones. The fastest taps were frequently described as vibrations. This implies that tapping and vibration are perhaps on a frequency continuum, yet perceptually distinct.

The participants split on describing their rubbing experiences as rubbing or lighter grazing. This may be in large part due to the implementation, which could not push hard enough into the participant's palm. This suggests using some kind of force feedback approach with a pressure sensor and an actuator into the contact plane to maintain consistent pressure across an uneven surface. This is in essence combining tapping and rubbing.

4.9.1 Design of Notifications

The qualitative responses we collected indicate how tapping and rubbing cues could be used for mobile phone alerts and feedback. Implicitly, participants compare the tapping and rubbing sensations to vibrotactile feedback commonly found on mobile phones. They also suggested a wide range of scenarios for tapping. On the one hand, strong, single taps were proposed for mobile phone alerts in outdoor environments where audio cannot be heard and vibration can often not be felt. On the other hand, taps were also proposed for use in quiet environments where the audible nature of vibrations makes them inappropriate.

Although most participants thought rubbing would be too subtle for alerts, many proposed that they would be great for feedback for when the device is in-hand, like when sending a text message. Many participants described this as preferable to the current "buzzing" that they get as confirmation, which is "uncomfortable" when the device is in-hand.

4.9.2 Distinguishing and Identifying Taps and Rubs

The accuracy on the *Distinguish* task is high, with a few exceptions. Likewise, results on the *Identify* task show that participants can identify 6–7 levels in the range 0–1N force and 5–29Hz.

This implies that the just-noticeable-differences for these levels are smaller than the intervals we used. While more work is needed to examine this, it is clear that our approach allows a fairly expressive haptic vocabulary. We expect that the higher error rates seen for the *Distinguish* task at higher amplitude and frequency stimuli resulted from decreased sensitive for these stimuli.

For all the stimulus conditions we tested for the *Distinguish* task, participants made significantly more errors in the *ConstantNumber* task than in the *ConstantDuration* task. In other words, the force or frequency of the stimulus was less of a distinguishing factor than the number of taps or rubs. For some applications, the number of taps and rubs may have a pre-learned meaning. Although designers can leverage this to improve the learnability of haptic icons, it also limits the number of viable distinct icons.

4.9.3 Locations for Tapping or Rubbing Apparatuses

Several situational and physical contexts elicit special design requirements. According to our participants, walking and driving reduce one's sensitivity to tactile feedback. Likewise, the pocket location on the thigh exhibits lower cutaneous sensitivity than the fingertip, in part due to the clothing and in part due to the reduced concentration of nerve endings. Using our current scale, harder taps should be used in applications to be used in these contexts, or be adapted to requirements of specific contexts.

Rubbing (at least our current version of it) is too subtle for in-the-pocket cues. Modifying it to allow pressing into the contact surface (combination of rubbing and tapping) might mitigate that issue. For in-the-hand, lighter tapping is best. Some participants mentioned discomfort with the harder taps and so for scenarios where the device is expected to be in the user's hand softer taps might be more appropriate. This suggests rubbing will be most effective when applied to in-the-hand scenarios.

4.10 Conclusions and Future Work

We presented tapping and rubbing, two tactile feedback techniques based on physical human-human interaction. These techniques are the result of our exploration into low frequency feedback using our soundTouch device, which uses voice coil motors to generate tactile feedback.

We made two contributions. First, we presented two new naturalistic tactile feedback techniques, tapping and rubbing, using the soundTouch technology. Second, our exploratory user studies of these two techniques demonstrated both that users perceive them as the taps and rubs encountered in daily experience, and that they provide a large range of distinguishable cues.

Future work will explore mobile implementations of our tapping and rubbing interfaces, applications to exploit these cues, and design of haptic icons for the mobile application space. One particular interest in this space concerns the pre-learned semantics of tapping and rubbing, and how they could productively guide haptic icon design. Another promising idea is to use multiple tapping actuators to generate perceptually different icons.

Mimicking physical touch opens up a new dimension for the types of information that can be conveyed through computer mediated touch. However, like the vibrotactile mappings of music discussed in the previous chapter, mimicking human touch also has its shortcomings. Not all types of information are well expressed using music and/or human touch. In the next chapter, we focus more on semantic content, looking at how to encode text and speech in tactile messages.

4.11 Acknowledgments

This chapter, in part, is a reprint of the material as it appears in Li, K.A., Baudisch, P., Griswold, W.G., and Hollan, J.D. Tapping and Rubbing: Exploring New Dimensions of Tactile Feedback with Voice Coil Motors. In Proceedings of the ACM Symposium on User Interface Software and Technology (UIST), pp. 181–190. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Bridging the Gap Between Semantics and Touch: Towards a Vibrotactile Encoding of Speech

5.1 Introduction

Text-based forms of Computer-Mediated Communication such as instant messaging, SMS and email have become increasingly popular. They are particularly useful in scenarios where users cannot communicate face-to-face. At the same time, people use text-based messaging for different types of tasks other then voice based solutions. For example, people will often send each other SMS messages just as a friendly hello [GP02]. At the other end of the spectrum, some people use SMS to avoid small talk, providing an even more efficient form of communication [Ito05].

Despite their many advantages over voice calls, these text-based forms of communication suffer from a number of drawbacks. One of the main drawbacks of text-based communication is that it lacks the ability to convey the sender's intent. In face-to-face communication, intent and emotive aspects are often conveyed via prosody [CDC96]. Orthographic conventions such as punctuation approximate this to some extent but for the most part, text-based communication still lacks the ability to express this information. As a result, misunderstandings commonly occur in text-based communication channels.

A vibrotactile communication channel has the potential to advance the state of computer mediated communication in two ways. First, it can augment existing forms of text-based communication, such as SMS, by providing accompanying prosodic cues. This could be used to overcome text-based communication's inability to convey emotive and intent aspects. Second, vibrotactile sequences could be sent as stand-alone messages, acting as a messaging backchannel. This would allow users to stay in touch with their friends or loved ones throughout the day.

One approach would be to create differentiable, distinguishable vibration sequences and map each of these sequences to a different message. This could be used to form a potentially large corpus of messages, but would be abstract, and thus difficult to learn. Geldard emplyed a similar approach has been used to create a vibrotactile encoding of written language, akin to vibration Morse code [Gel60]. Using his *Vibratese Language*, users were able to receive words at 90% accuracy at rates of up to 38 words per minute after 65 hours of training.

We were interested in developing a vibrotactile encoding of language that could be learned in significantly less time, by taking advantage of human experience. In doing so, we hoped to generate vibrotactile sequences with pre-learned meaning. Our goal was to examine how people perceive text and speech and how these forms of communication could be mapped to the tactile space.

To examine which elements of text were most important for creating a vibrotactile mapping of English, we ran a user study. Based on the results along with a literature survey of linguistics, we determined that linguistic prosody (rhythm, intonation, and stress) and number of syllables were important properties for a vibrotactile encoding of English speech.

As we were particularly interested in grounding our findings in a practical application, we have incorporated these messages into a messaging backchannel application called *VibeMessaging*. This is an application that runs on Windows Mobile phones that allows users to send each other purely vibrotactile messages. We present the results of a formative study, looking at how couples might use such a messaging backchannel to communicate simple messages from a closed language. We conclude with future directions on how to build on the findings we present here.

5.2 Related Works

We draw on the literature from vibrotactile communication and linguistics.

5.2.1 Vibrotactile Communication

Chang's *ComTouch* explored what type of communication could be conveyed using unstructured vibrations generated by end-users with pressure sensitive pads [COJ$^+$02]. To create a larger vocabulary, a number of other researchers have looked at using multiple actuators either with vibrotactile actuators [BBP06] or with voice coil motors [TDRR99]. Geldard created a messaging application using multiple actuators, examining how a vibrotactile Morse code equivalent could be generated [Gel60]. He mapped different letters of the English language to vibration pulses of varying duration, and location using five different actuators mounted on a user's chest. Users of his *Vibratese Language* were able to receive messages at 38 words per minute after 65 hours of training. This demonstrates the upper bound of how many bits of information could be received by users via tactile methods, but requires a significant amount of training.

A number of researchers have also attempted to create vibrotactile encodings of speech, directly from acoustic data. This work has generally been motivated by two problems. First, studies of hearing impaired individuals has found that the intelligibility of their speech is not a problem, but they often have difficulty conveying emotion due to prosodic errors [CDC96]. A vibrotactile mapping of prosody could be used by members of this community to convey emotion. Second, a tactile encoding of speech could be used for enabling hearing impaired users to receive information they otherwise would not be able to sense [Spe80].

Different approaches to creating a vibrotactile encoding of speech have considered both single as well as multiple actuator approaches. Single actuator approaches have typically focused on using envelope detectors to perform syllable segmentation [Mer75]. Frequency based approaches commonly use multiple actuators, with different frequency bands mapped to different actuators. Carney and Beachler found that with this approach, multiple vibrators did not perform better than single ones [Car88]. Similarly, Yuan et al. also found that multiple voice coil motors did not outperform a single one, when used to map accoustic information to a tactile array [YRD05]. Both of these researchers noted that mapping acoustic information to multiple channels convolutes a user's perception of the temporal aspects of speech. Like other research in this area, their findings suggest that the usage of multiple actuators might actually be counterproductive for mapping different elements of speech to the vibrotactile channel. Based on these findings, we opted to use a single actuator instead of multiple ones.

Additionally, because people do not think of speech in terms of low level acoustic properties, we focused instead on higher level linguistic features. Although this deviates from past approaches, we hypothesized that by doing so, we could generate messages that had pre-learned meaning, leveraging users' familiarity with language.

5.2.2 Linguistics

In considering how people perceive speech, the area of linguistics is particularly relevant. Linguistic prosody can encompass a number of characteristics of speech but most linguists agree that the main components of linguistic prosody are rhythm, stress and intonation [LL82]. Prosody reflects information such as emotion that is not encoded in writing [CDC96]. Because these prosodic elements are difficult to convey in writing, text-based computer mediated communication is unable to convey this aspect of communication [SHT96].

Speech recognition researchers have attempted to extract linguistic features from speech data. One approach to speech recognition is to extract linguistic features from speech data and then perform the recognition on each of the different linguistic components. This could significantly improve the accuracy of speech recognition systems. Common approaches look at performing syllable segmentation, identifying stress and monitoring changes in intonation. Approaches to syllable segmentation typically use envelope detection [VWT06]. Energy level detection has been used for stress detection [TN05]. Pitch tracking has been used to identify changes in intonation [SBP^{+}92]. It is likely that in the future, algorithms for detecting these different linguistic properties will be feasible. Unfortunately, as of today, none of these techniques have been perfected. Since these automated approaches for extracting linguistics are still error-prone, we were unable to use an automated approach for our user study. Instead we used a Wizard of Oz approach, generating vibrotactile patterns by hand.

5.3 What Elements of Linguistics are Relevant?

The purpose of this study was to get a better understating of the important properties of text messages for a vibrotactile encoding. Although we were interested in how people perceive speech to be mapped to vibrotactile sequences, we were also interested in how approximations of speech could be mapped to vibrotactile sequences when using orthographic conventions such as punctuation.

5.3.1 Apparatus

The study was conducted using an *HTC Tornado Smartphone*. This device has 64MB RAM, 200MHz CPU and runs Windows Mobile 5.0. The phone hardware was not modified in any way.

5.3.2 Task and Stimuli

To generate vibrations of different intensity, we used the software approach described in Chapter 3. This approach has been shown to be able to generate 10 user-differentiable levels of vibration intensity at 20ms duration. These pulses formed the building blocks for the vibration sequences we used.

Twenty vibration sequences were manually generated a priori. These sequences were approximately 0.5s long and were constructed from 20ms blocks of different vibration intensities.

In each trial, participants were presented with a vibration sequence, which was repeated after a 1s pause. Then, in a forced choice response, participants were asked to select which one of five text phrases best matched the vibration sequence. The text-phrases used were “hello?”, “goodbye.”, “I miss you.”, are you busy?” and “where are you?”. These phrases were selected because they occur frequently in SMS messages

[GP02]. The vibration sequence was delivered via a mobile phone held in the user's hand.

5.3.3 Participants

Sixteen volunteers (5 female) from within our institution were recruited. Ages ranged from 22 to 31. Each received a gift card as a gratuity for their time. All participants owned a mobile phone. Each participant took approximately 20 minutes to complete the experiment.

5.3.4 Procedure and Design

There were three instances of each of the ten different vibration sequences, resulting in 30 trials per participant. The order of presentation was pre-randomized. The order of presentation of stimuli was the same for all participants.

5.3.5 Discussion

Participants largely agreed on the same text phrases matching to the same vibrations. At least 10 of 16 participants chose the vibration patterns plotted in Figure 5.1. While we cannot make statements about how elements of linguistic prosody should be mapped to the vibrotactile channel, these results do help us form some hypotheses going forward.

Based on the results, there were a number of takeaways about what characteristics of English speech are important for a vibrotactile mapping. Additionally, as this study looked at mapping text to vibrations, there are a number of different factors that arise when mapping speech to vibration that will need to be considered. We consider some of the most important factors here and discuss a few of our hypotheses going forward.

Duration

Our findings suggest duration does not need to be preserved. Specifically, vibration patterns can be longer than the time it would take to speak a particular phrase. This

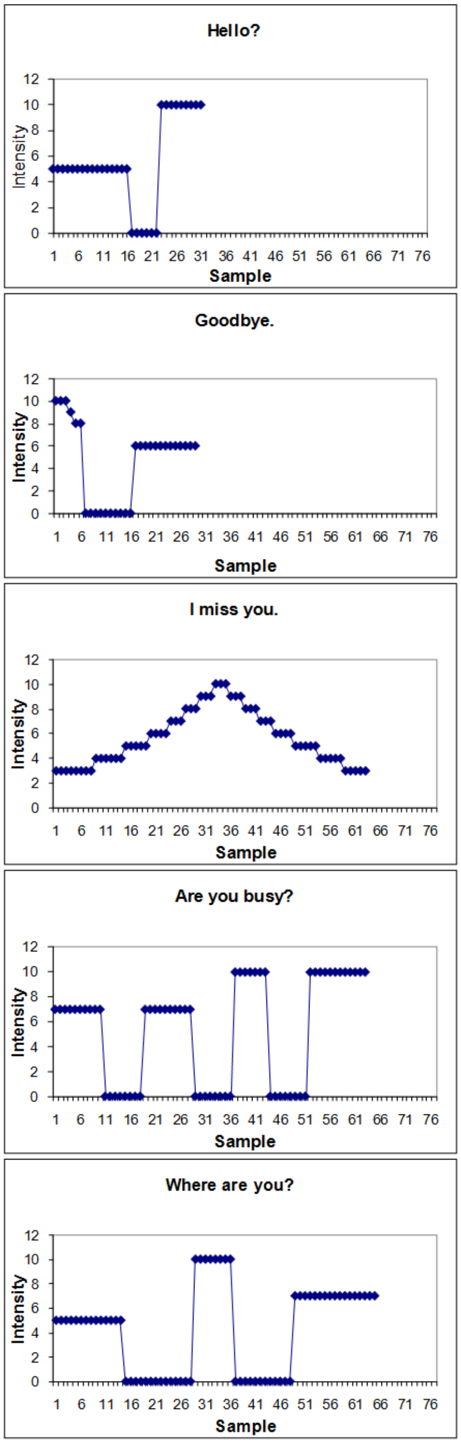


Figure 5.1: Plots of the vibrotactile patterns that most participants associated with the different text phrases.

suggests that duration may be an additional variable in which extra information can be encoded.

Syllables

The number of syllables seems to be one of the most important characteristics for mapping phrases to vibrotactile patterns. The number of syllables is important for comprehension and should be mapped to demarcated vibration pulses. It also seems that inserting pauses between syllables will yield higher recognition rates.

Intonation

Intonation seems to be an important factor for mapping phrases to vibrotactile patterns. Users seem to perceive that higher pitch should be mapped to stronger vibrations. Lower pitch should be mapped to weaker vibrations.

Stress

Stress within phrases seemed to map well to stronger vibrations. This potentially collides with rising intonation and will need to be resolved.

5.4 Vibe Messaging: A Tactile Messaging Backchannel Application

We were particularly interested in a practical application of the findings from this work. To explore how users might use such a communication backchannel in their day-to-day lives, we built an application called Vibe Messaging. Vibe Messaging runs on Windows Mobile Smartphones. The application allows users to send each other one of the five vibrotactile messages that we reported on in the previous section. Currently, users can only send and receive messages that have been manually constructed by us but we have plans to expand that to end-user authored vibration sequences.

5.4.1 Initial Feedback

We were particularly interested in how couples might use this type of messaging application as a lightweight communication backchannel throughout their day. In a formative user study, two couples used the application over the course of a few days. Participants reported being able to identify the message based on the vibrotactile pattern. Additionally, there seems to be differences in terms of the types of messages couples want to use, depending on whether they are long distance or not. Finally, all participants mentioned that they would like to create their own messages that better reflected the types of messages they might convey.

5.5 Future Work

This work forms the preliminary work for creating a vibrotactile encoding of speech. Based on the findings discussed here, a natural next step would be to test each one of the linguistic properties that we discussed here and how they map for a variety of different phrases spoken by a variety of different speakers.

An augmented SMS application could help visualize a text message with an accompanying vibrotactile pattern. We are also currently looking at how to enable end-user generated vibrotactile patterns. We are also interested in giving this application to users in a longitudinal study to examine their usage over time. Also, while conveying intent and emotion is easier facilitated with our vibrotactile communication language, it remains to be seen exactly what types of emotion can be conveyed with different types of messages. This

Finally, this work has largely focused on the English language which is not a tonal language. It would be interesting to compare and contrast the applicability of this approach to a tonal language such as Chinese.

5.6 Conclusion

In this chapter we have presented the initial results of an effort to create a tactile encoding of speech. We have presented the results of a user study examining how users perceive different types of text messages to be mapped to different vibrotactile patterns. Using these results, we have identified a number of promising vibrotactile sequences for a closed language of commonly used text messages. Furthermore, we have incorporated our findings into a lightweight messaging application. Initial results from a formative user study have found that users are able to identify the different messages associated with the closed language. Additionally, the ability to author their own messages seems very important to users.

Chapter 6

New Directions for Research on Tactile Communication

A core principle of this dissertation is that we can exploit a users' familiarity with existing stimuli to generate tactile messages with pre-learned meanings. However, like much of the rest of the work in the haptics space, our work has focused on how users receive tactile messages. In future work, I plan to extend this approach to a new body of work focusing on how users might author tactile messages to be sent to other users. By providing users with tools to create and send tactile messages with pre-learned meaning, a new class of messaging emerges, which I like to call *Creative Tactile Interaction and Communication*.

I have taken the first step in this direction with a project on end-user generated tapping and rubbing. This extends my previous work on tapping and rubbing, allowing users to send taps and rubs to someone else by tapping and rubbing their own input device. My current prototype will explore how users interpret and send different messages of intent. This is the first instantiation of a new class of input devices that will support end-user generation of tactile messages. Here I mention a number of projects that could exist in this new space.

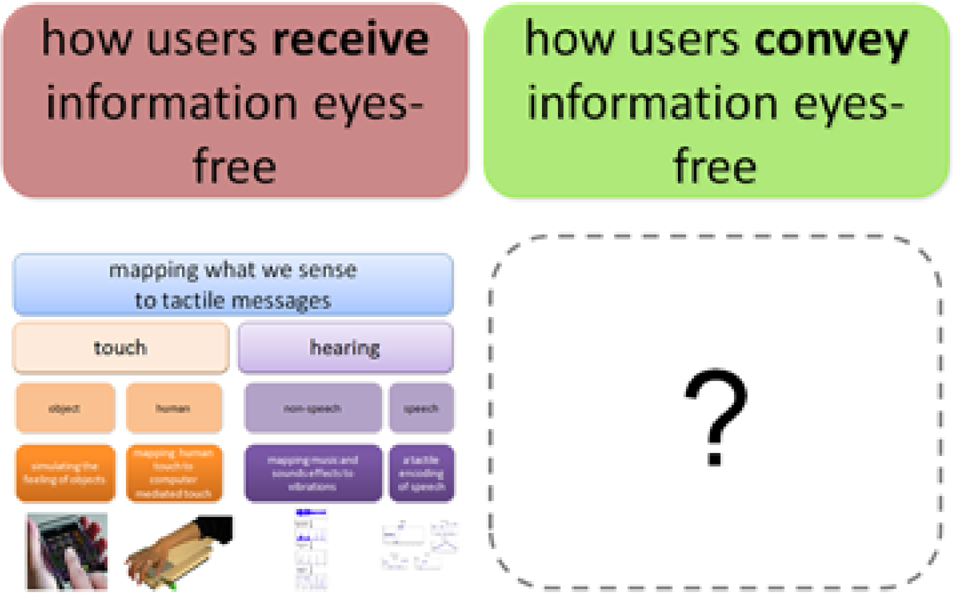


Figure 6.1: Past work has focused on how people *receive* messages, not how they can be *sent*.

6.1 Creation of New Tactile Languages

My previous work on vibrotactile encodings of speech provides users with a common starting point to communicate with one another via tactile methods. When this is combined with the ability to generate arbitrary tactile messages, a new class of tactile language emerges. How will users create new tactile languages? Will the resulting languages differ between groups? How will languages generated across groups be similar?

6.2 Emotive Aspects of Computer Mediated Tactile Communication

A large body of psychology literature has found that human-human touch in the physical world is excellent for conveying intent and emotion. With mechanisms in place to support the sending and receiving of tactile messages, a natural question to ask is, can computer mediated tactile communication provide the emotive aspects of com-

Figure 6.2: Prototype for enabling end-user generated tapping and rubbing.

munication that are currently absent in text-based communication mediums? Are there pre-learned emotive aspects for this type of touch? My earlier work with tapping and rubbing suggests pre-learned emotive aspects exist but only when provided in context. Given the appropriate context, this can be more closely examined.

6.3 The Role of Tactile Communication in a Multimodal Communication Device

New communication mediums such as text-messaging, instant messaging and email have resulted in different usage habits. Tactile-based communication will likely be used for different purposes as well, but it remains to be seen what its role will be and how it will be used in combination with other forms of communication.

6.4 Engineering New Input and Output Devices

New forms of communication and interaction require a new class of devices. This can range from wearable devices such as watches to actuators embedded in a car seat. A number of engineering challenges exist with producing hardware appropriate for these scenarios. New actuator technologies need to be explored to provide appropriate tactile feedback. Miniaturization and power consumption are issues when trying to deploy these actuators to mobile devices.

Chapter 7

Final Thoughts

As users become increasingly mobile, computing must take on a different form to address the needs of users. Mobile phones have advanced considerably in recent times, providing new functionality for users on the go and have become the *de facto* standard for mobile computing. Unfortunately, since their user interfaces have been guided by the design lessons learned from the desktop space, these interfaces have been highly visual. While visual interfaces work well for conveying certain types of information, the mobile nature of users requires new forms of interaction for new usage scenarios. Specifically, there is a need for enabling eyes-free interaction for when users cannot look at their devices.

7.1 Methodology

Throughout this dissertation we have applied a number of different methods to evaluate our interfaces, ranging from surveys and qualitative measures to quantitative measurements in laboratory experiments. A thorough discussion of the different tradeoffs of these different approaches is outside the scope of this dissertation. However, because we have employed a variety of different techniques, we reflect on our experience with them and what we have learned throughout the process. In this section we comment on our usage of surveys, quantitative measurements, and qualitative techniques throughout this dissertation. The overall theme behind the application of our methods has been the goal of raising the bar for technology by enabling new usage scenarios that were previously not possible.

7.1.1 Surveys

In Chapter 2 we approached the problem of eyes-free information access on mobile phones from a needs assessment point of view. We were particularly interested in what types of information needs people have while talking on the phone. To find out, we interviewed a number of current smartphone users about the types of information that they would like access to while talking on the phone.

There are a number of tradeoffs with this approach. On the one hand, we could collect quantitative data on their usage habits, monitoring their most commonly used features. However, the downside of this approach would be that they might not report features that they would really like to use but simply do not because they are too hard to access.

On the other hand, by asking users what they would like to have access to if they could, we are able to get an idea of the types of information that users are interested in. The downside of this approach is that users often make mistakes when self-reporting data; the feature they think they use the most often may not actually be the one they use most often. Given the tradeoffs between these two approaches, we opted for the latter. The main reason was that we already knew there were certain tasks that users would not use because they were difficult to access while talking on the phone. Rather than focusing on increasing the speed at which users could access information on the phone, we were more interested in enabling them to use their phone in situations they previously could not.

7.1.2 Quantitative Measurements

In a number of our projects we employed quantitative measurements. In these projects we did not perform a direct comparison of our projects with other existing solutions, with the exception of blindSight which was discussed in Chapter 2. The main reason for this was once again our desire to enable new usage scenarios rather than improving existing interaction techniques. With respect to blindSight, our quantitative measures were used to evaluate the cost of using a new system rather than measuring whether it outperformed an existing system, as is often the case with quantitative methods. With respect to tapping and rubbing described in Chapter 4, we were again interested in the limitations of our new technology. Although we had qualitative findings suggesting that users perceived our prototypes to generate taps and rubs, we wanted to know the limitations of this new technology. Specifically, we wanted to ensure that a sufficient range of bandwidth could be generated with this new form of feedback.

7.1.3 Qualitative Measurements

Our qualitative studies often proved to be the most insightful. Although it is harder to make definitive claims with respect to these findings, we also found that for this type of research that looks to enable new usage scenarios. This was particularly the case in the final blindSight study described in Chapter 2. In this study we examined how users would use the application while talking on the phone. While we could have performed a quantitative study, focusing on task time and error rates, we would have had to control so many variables that we would have gained little insight. Instead, we opted for a qualitative study to get a better sense of the strengths, weaknesses and overall usability of the application. These types of results would have been impossible to gather from a quantitative study. Similarly, in our studies of tapping and rubbing we collected qualitative results as well as quantitative ones. We did collect quantitative numbers to demonstrate the wide variety of differing amplitude and frequency of taps and rubs that we could generate, but we were particularly interested in the qualitative findings from this work. This allowed us to gauge how users perceived this new form of tactile feedback, something quantitative data and surveys could not have achieved.

7.2 Contributions

The main contribution of this work has been to propose a new direction for haptics research. Rather than focusing on increasing the bandwidth of information transfer through touch, this dissertation has proposed mapping stimuli that people are already familiar with to tactile messages. By doing so, tactile messages have a pre-learned meaning associated with them. We have presented the foundations for a new area of haptics research that enables eyes-free interaction by creating tactile messages based on human experience. The contribution of this thesis has come in four parts:

1. We presented the strengths and weaknesses of auditory feedback with a motivating example for how eyes-free interaction can be enabled using auditory feedback. To explore the properties of auditory feedback, we built an application called blindSight that allows users to access mobile phone content via auditory feedback. This allows them to interact with their mobile phones in scenarios where they cannot look at their devices, such as when talking on the phone.

2. We demonstrated how semantic associations from music can be transferred to the tactile channel. This was done by encoding music in vibrotactile sequences. The presented algorithm was incorporated into a buddy proximity application, mapping music cues to vibrotactile sequences played on commodity mobile phones. An ecologically valid study was presented to evaluate the utility of such an application.

3. We demonstrated that computer-mediated human touch can be used to create the sensation of interpersonal physical touch. As a proof of concept, we presented two prototypes that mimic tapping and rubbing. This was motivated in part by the lack of control of the vibrotactile actuator. Although vibration was adequate for mapping music and sounds to the tactile channel, we needed a higher fidelity tactile device to recreate human touch.

4. We presented the initial work for creating a vibrotactile mapping of English text. We presented the results of user studies examining how users perceive linguistic prosody to be mapped to different types of vibrotactile sequences. This has implications both for an eyes-free messaging backchannel as well as for augmenting text-messages with additional emotive information.

7.3 Outlook on Tactile Communication Research

We have provided the initial workings for a new area of haptic research that looks at exploiting human experience to create tactile messages with pre-learned meaning. Like most of the haptic literature, we have also focused on how users *receive tactile messages*. To date, very little research has been done on how users can *generate* tactile messages. The result is that most applications have been one-way communication, focusing only on giving information to users, providing no way for users to generate their own messages. We believe there is tremendous opportunity in this space for tactile message generation. As the next step in this area of research, we believe users can be empowered to create tactile messages in a similar manner as what we have described here. By leveraging human experience, we can provide input mechanisms that users are familiar with, providing a powerful tool for end-user authoring of tactile messages. We briefly discussed a number of interesting projects that could appear in this new space.

As mobile devices become more pervasive in an increasingly mobile world, users will require new ways to receive information from their devices. Tactile feedback is a promising alternative to the traditional visual and auditory interfaces of times past. Although the work here has presented a new perspective for haptics research, there are still many research opportunities for this promising interaction modality in the mobile space.

Bibliography

[Agr87] ML Agronin. The design of a nine-string six-degree-of-freedom force-feedback joystick for telemanipulation. In *Proceedings of the Workshop on Space Telerobotics,*, volume 2, 1987.

[Bau98] Patrick Baudisch. Don't click, paint! using toggle maps to manipulate sets of toggle switches. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 65–66, New York, NY, USA, 1998. ACM.

[BB04a] Stephen Brewster and Lorna M. Brown. Tactons: structured tactile messages for non-visual information display. In *AUIC '04: Proceedings of the fifth conference on Australasian user interface*, pages 15–23, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[BB04b] Stephen A. Brewster and Lorna M. Brown. Non-visual information display using tactons. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 787–788, New York, NY, USA, 2004. ACM.

[BBP06] Lorna M. Brown, Stephen A. Brewster, and Helen C. Purchase. Multidimensional tactons for non-visual information presentation in mobile devices. In *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 231–238, New York, NY, USA, 2006. ACM.

[BBPB02] M. Bouzit, G. Burdea, G. Popescu, and R. Boian. The Rutgers Master II-new design force-feedback glove. *IEEE/ASME transactions on mechatronics*, 7(2):256–263, 2002.

[BK06] Lorna M. Brown and Topi Kaaresoja. Feel who's talking: using tactons for mobile phone alerts. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 604–609, New York, NY, USA, 2006. ACM.

[BLEW04] A. Berenzweig, B. Logan, D.P.W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[BMV+98] Richard Borovoy, Fred Martin, Sunil Vemuri, Mitchel Resnick, Brian Silverman, and Chris Hancock. Meme tags and community mirrors: moving from conferences to collaboration. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 159–168, New York, NY, USA, 1998. ACM.

[Bre98] Stephen A. Brewster. Using non-speech sounds to provide navigation cues. *ACM Transactions on Computer-Human Interaction*, 5:224–259, 1998.

[BWE93] Stephen A. Brewster, Peter C. Wright, and Alistair D. N. Edwards. An evaluation of earcons for use in auditory human-computer interfaces. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 222–227, New York, NY, USA, 1993. ACM.

[Car88] A.E. Carney. Vibrotactile Perception of Segmental Features of Speech A Comparison of Single-Channel and Multichannel Instruments. *Journal of Speech, Language and Hearing Research*, 31(3):438–448, 1988.

[CC00] R.W. Cholewiak and A.A. Collins. The generation of vibrotactile patterns on a linear array: Influences of body site, time, and presentation mode. *Perception and Psychophysics*, 62(6):1220–1235, 2000.

[CDC96] R. Cowie and E. Douglas-Cowie. Automatic statistical analysis of the signal and prosodic signs ofemotion in speech. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 3, 1996.

[CO05] Angela Chang and Conor O'Sullivan. Audio-haptic feedback in mobile phones. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1264–1267, New York, NY, USA, 2005. ACM.

[COJ+02] Angela Chang, Sile O'Modhrain, Rob Jacob, Eric Gunther, and Hiroshi Ishii. Comtouch: design of a vibrotactile communication device. In *DIS '02: Proceedings of the 4th conference on Designing interactive systems*, pages 312–320, New York, NY, USA, 2002. ACM.

[CSC+06] Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith, and Alex Varshavsky. Practical metropolitan-scale positioning for GSM phones. In *UbiComp 2006: Ubiquitous Computing*, volume 4206 of

Lecture Notes in Computer Science, pages 225–242. Springer Berlin / Heidelberg, 2006.

[cyb] http://www.immersion.com/3d/products/cybergrasp.php.

[DMSW90] TW Duerig, KN Melton, D. Stockel, and CM Wayman. Engineering aspects of shape memory alloys. *Butterworth-Heinemann, Reed Books Services Ltd, P. O. Box 5, Rushden, Northants, NN 10 9 YX, UK, 1990. 499*, 1990.

[Dod] http://www.dodgeball.com.

[DY01] Paul H. Dietz and William S. Yerazunis. Real-time audio buffering for telephone applications. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 193–194, New York, NY, USA, 2001. ACM.

[ELW+98] S. Ertan, C. Lee, A. Willets, H. Tan, and A. Pentland. A wearable haptic navigation guidance system. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 164–165, 1998.

[FCAE98] BJ Fogg, Lawrence D. Cutler, Perry Arnold, and Chris Eisbach. Handjive: a device for interpersonal haptic entertainment. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 57–64, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[FT99] Masaaki Fukumoto and Yoshinobu Tonomura. Whisper: a wristwatch style wearable handset. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 112–119, New York, NY, USA, 1999. ACM.

[Gav89] W.W. Gaver. The SonicFinder: An interface that uses auditory icons. *Human-Computer Interaction*, 4(1):67–94, 1989.

[GC56] F.A. Geldard and VIRGINIA UNIV CHARLOTTESVILLE. *Adventures in Tactile Literacy.* Defense Technical Information Center, 1956.

[Gel60] F.A. Geldard. Some neglected possibilities of communication. *Science*, 131(3413):1583–1588, 1960.

[Gel75] F.A. Geldard. *Sensory saltation: Metastability in the perceptual world*. Lawrence Erlbaum Associates, 1975.

[GP02] R.E. Grinter and L. Palen. Instant messaging in teen life. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 21–30. ACM New York, NY, USA, 2002.

[GR93] D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 80–87. ACM New York, NY, USA, 1993.

[GSB$^+$04] WG Griswold, P. Shanahan, SW Brown, R. Boyer, M. Ratto, RB Shapiro, and TM Truong. ActiveCampus: experiments in community-oriented ubiquitous computing. *Computer*, 37(10):73–81, 2004.

[GWC$^+$07] Iván E. González, Jacob O. Wobbrock, Duen Horng Chau, Andrew Faulring, and Brad A. Myers. Eyes on the road, hands on the wheel: thumb-based interaction techniques for input on steering wheels. In *GI '07: Proceedings of Graphics Interface 2007*, pages 95–102, New York, NY, USA, 2007. ACM.

[HB07] Eve Hoggan and Stephen Brewster. New parameters for tacton design. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2417–2422, New York, NY, USA, 2007. ACM.

[HFW99] L.E. Holmquist, J. Falk, and J. Wigström. Supporting group collaboration with interpersonal awareness devices. *Personal and Ubiquitous Computing*, 3(1):13–21, 1999.

[HH09] C. Harrison and S.E. Hudson. Providing dynamically changeable physical buttons on a visual display. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 299–308. ACM New York, NY, USA, 2009.

[HI05] Joyce Ho and Stephen S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 909–918, New York, NY, USA, 2005. ACM.

[HLR01] Rebecca Hansson, Peter Ljungstrand, and Johan RedstrC6m. Subtle and public notification cues for mobile devices. In *Ubicomp 2001: Ubiquitous Computing*, volume 2201 of *Lecture Notes in Computer Science*, pages 240–246. Springer Berlin / Heidelberg, 2001.

[HM07] V. Hayward and K.E. MacLean. Do it yourself haptics: part I. *IEEE Robotics and Automation Magazine*, 14(4):88–104, 2007.

[HMT03] S. Hiraoka, I. Miyamoto, and K. Tomimatsu. Behind Touch: A Text Input Method for Mobile Phone by The Back and Tactile Sense Interface. *Joho Shori Gakkai Shinpojiumu Ronbunshu*, 2003(7):131–138, 2003.

[Hor94] T. Hornstein. Telephone voice interfaces on the cheap. In *Computer Science Research at UBILAB, Strategy and Projects; Proc. UBILAB Conf.-94. Konstanz: Universitätsverlag Konstanz*, 1994.

[HP97] J.L. Hennessy and D.A. Patterson. *Computer organization and design: the hardware/software interface*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1997.

[HPG$^+$00] J.H.L. Hansen, J. Plucienkowski, S. Gallant, B. Pellom, and W. Ward. " CU-Move": Robust Speech Processing for In-Vehicle Speech Systems. In *Sixth International Conference on Spoken Language Processing*. ISCA, 2000.

[HS96] Scott E. Hudson and Ian Smith. Electronic mail previews using non-speech audio. In *CHI '96: Conference companion on Human factors in computing systems*, pages 237–238, New York, NY, USA, 1996. ACM.

[HVV98] S.S. Haykin and B. Van Veen. *Signals and systems*. John Wiley & Sons, Inc. New York, NY, USA, 1998.

[Imm] http://www.immersion.com.

[ISC$^+$05] Giovanni Iachello, Ian Smith, Sunny Consolvo, Gregory D. Abowd, Jeff Hughes, James Howard, Fred Potter, James Scott, Timothy Sohn, Jeffrey Hightower, and Anthony LaMarca. Control, deception, and communication: Evaluating the deployment of a location-enhanced messaging service. In *UbiComp 2005: Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pages 213–231. Springer Berlin / Heidelberg, 2005.

[Ish08] Hiroshi Ishii. Tangible bits: beyond pixels. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages xv–xxv, New York, NY, USA, 2008. ACM.

[Ito05] M. Ito. Introduction: personal, portable, pedestrian. *Personal, portable, pedestrian: Mobile phones in Japanese life*, pages 1–16, 2005.

[IWR02] Ellen Isaacs, Alan Walendowski, and Dipti Ranganthan. Hubbub: a sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 179–186, New York, NY, USA, 2002. ACM.

[KH04] John Krumm and Ken Hinckley. The NearMe wireless proximity server. In *UbiComp 2004: Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 283–300. Springer Berlin / Heidelberg, 2004.

[LBH08] Kevin A. Li, Patrick Baudisch, and Ken Hinckley. Blindsight: eyes-free access to mobile phones. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1389–1398, New York, NY, USA, 2008. ACM.

[LCC$^+$05] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill Schilit. Place lab: Device positioning using radio beacons in the wild. In *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 116–133. Springer Berlin / Heidelberg, 2005.

[LDL$^+$04] Johnny C. Lee, Paul H. Dietz, Darren Leigh, William S. Yerazunis, and Scott E. Hudson. Haptic pen: a tactile feedback stylus for touch screens. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 291–294, New York, NY, USA, 2004. ACM.

[LKDH05] R. Leuschke, EKT Kurihara, J. Dosher, and B. Hannaford. High fidelity multi finger haptic display. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 606–608, 2005.

[LL82] P. Ladefoged and P. Ladefoged. A course in phonetics. 1982.

[Loo] http://www.loopt.com.

[LPL$^+$06] Joseph Luk, Jerome Pasquero, Shannon Little, Karon MacLean, Vincent Levesque, and Vincent Hayward. A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 171–180, New York, NY, USA, 2006. ACM.

[LSHG08] Kevin A. Li, Timothy Y. Sohn, Steven Huang, and William G. Griswold. Peopletones: a system for the detection and notification of buddy proximity on mobile phones. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 160–173, New York, NY, USA, 2008. ACM.

[LSP$^+$04] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. Twiddler typing: one-handed chording text entry for mobile phones. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 671–678, New York, NY, USA, 2004. ACM.

[LSS$^+$04] Kent Lyons, Christopher Skeels, Thad Starner, Cornelis M. Snoeck, Benjamin A. Wong, and Daniel Ashbrook. Augmenting conversations using dual-purpose speech. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 237–246, New York, NY, USA, 2004. ACM.

[LTCK03] A. Liu, F. Tendick, K. Cleary, and C. Kaufmann. A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators & Virtual Environments*, 12(6):599–614, 2003.

[Mac02] S. MacKenzie. Mobile text entry using three keys. In *Proceedings of the second Nordic conference on Human-computer interaction*, pages 27–34. ACM New York, NY, USA, 2002.

[MBWF97] Elizabeth D. Mynatt, Maribeth Back, Roy Want, and Ron Frederick. Audio aura: light-weight audio augmented reality. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 211–212, New York, NY, USA, 1997. ACM.

[MDH$^+$03] Jennifer Mankoff, Anind K. Dey, Gary Hsieh, Julie Kientz, Scott Lederer, and Morgan Ames. Heuristic evaluation of ambient displays. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 169–176, New York, NY, USA, 2003. ACM.

[MDM$^+$04] Tara Matthews, Anind K. Dey, Jennifer Mankoff, Scott Carter, and Tye Rattenbury. A toolkit for managing user attention in peripheral displays. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 247–256, New York, NY, USA, 2004. ACM.

[ME97] MA Marics and G. Engelbeck. Designing voice menu applications for telephones. *Handbook of human-computer interaction*, pages 1085–1102, 1997.

[Mer75] P. Mermelstein. Automatic segmentation of speech into syllabic units. *The Journal of the Acoustical Society of America*, 58:880, 1975.

[MH91] PH Marbot and B. Hannaford. Mini direct drive arm for biomedical applications. In *Proceedings of ICAR*, volume 91, pages 859–864, 1991.

[MH08] KE MacLean and V. Hayward. Do It Yourself Haptics: Part II. *IEEE Robotics & Automation Magazine*, 15(1):104–119, 2008.

[MI08] Benoît Martin and Poika Isokoski. Edgewrite with integrated corner sequence help. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 583–592, New York, NY, USA, 2008. ACM.

[MS94] T.H. Massie and J.K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, volume 55, pages 295–300, 1994.

[MSS04] Natalia Marmasse, Chris Schmandt, and David Spectre. WatchMe: Communication and awareness between members of a closely-knit group. In *UbiComp 2004: Ubiquitous Computing*, volume 3205 of *Lecture Notes in Computer Science*, pages 214–231. Springer Berlin / Heidelberg, 2004.

[MSWHQ08] Roderick Murray-Smith, John Williamson, Stephen Hughes, and Torben Quaade. Stane: synthesized surfaces for tactile input. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1299–1302, New York, NY, USA, 2008. ACM.

[OFH08] Alex Olwal, Steven Feiner, and Susanna Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 295–304, New York, NY, USA, 2008. ACM.

[OTT$^+$95] M. Ouhyoung, W.N. Tsai, M.C. Tsai, J.R. Wu, C.H. Huang, and T.J. Yang. A low-cost force feedback joystick and its use in PC video games. *IEEE Transactions on Consumer Electronics*, 41(3):787–794, 1995.

[PAM07] Saverio Perugini, Taylor J. Anderson, and William F. Moroney. A study of out-of-turn interaction in menu-based, ivr, voicemail systems. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 961–970, New York, NY, USA, 2007. ACM.

[peo] http://www.plazes.com.

[PMR02] Ivan Poupyrev, Shigeaki Maruyama, and Jun Rekimoto. Ambient touch: designing tactile interfaces for handheld devices. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 51–60, New York, NY, USA, 2002. ACM.

[RG83] J.L. Rosenberger and M. Gasko. Comparing location estimators: Trimmed means, medians, and trimean. *Understanding robust and exploratory data analysis*, pages 297–338, 1983.

[RISO] J. Rekimoto, T. Ishizawa, C. Schwesig, and H. Oba. PreSense: interaction techniques for finger sensing input devices.

[RV92] Paul Resnick and Robert A. Virzi. Skip and scan: cleaning up telephone interface. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 419–426, New York, NY, USA, 1992. ACM.

[Sat61] M. Sato. Response of Pacinian corpuscles to sinusoidal vibration. *The Journal of physiology*, 159(3):391, 1961.

[SBP$^+$92] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg. ToBI: A standard for labeling English prosody. In *Second International Conference on Spoken Language Processing*. ISCA, 1992.

[SCL$^+$05] Ian Smith, Sunny Consolvo, Anthony Lamarca, Jeffrey Hightower, James Scott, Timothy Sohn, Jeff Hughes, Giovanni Iachello, and Gregory D. Abowd. Social disclosure of place: From location technology to communication practices. In *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 134–151. Springer Berlin / Heidelberg, 2005.

[SH06] Masanori Sugimoto and Keiichi Hiroki. Hybridtouch: an intuitive manipulation technique for pdas using their front and rear surfaces. In *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 137–140, New York, NY, USA, 2006. ACM.

[Shn96] B. Shneiderman. The eyes have it: a task by data type taxonomy for informationvisualizations. In *IEEE Symposium on Visual Languages, 1996. Proceedings.*, pages 336–343, 1996.

[SHT96] S. Shattuck-Hufnagel and A.E. Turk. A prosody tutorial for investigators of auditory sentence processing. *Journal of psycholinguistic research*, 25(2):193–247, 1996.

[SLL$^+$05] Timothy Sohn, Kevin A. Li, Gunny Lee, Ian Smith, James Scott, and William G. Griswold. Place-its: A study of location-based reminders on mobile phones. In *UbiComp 2005: Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pages 232–250. Springer Berlin / Heidelberg, 2005.

[SMS$^+$01] S.S. Snibbe, K.E. MacLean, R. Shaw, J. Roderick, W.L. Verplank, and M. Scheeff. Haptic techniques for media control. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 199–208. ACM New York, NY, USA, 2001.

[Spe80] KE Spens. Tactile speech communication aids for the deaf: A comparison. *Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, Sweden, Quarterly Progress and Status Report*, 4:185–231, 1980.

[SS00] Nitin Sawhney and Chris Schmandt. Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments. *ACM Trans. Comput.-Hum. Interact.*, 7(3):353–383, 2000.

[SVL$^+$06] Timothy Sohn, Alex Varshavsky, Anthony LaMarca, Mike Y. Chen, Tanzeem Choudhury, Ian Smith, Sunny Consolvo, Jeffrey Hightower, William G. Griswold, and Eyal de Lara. Mobility detection using everyday GSM traces. In *UbiComp 2006: Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*, pages 212–224. Springer Berlin / Heidelberg, 2006.

[Tac] http://www.eaiinfo.com/Tactor%20Products.htm.

[TDRR99] H.Z. Tan, N.I. Durlach, W.M. Rabinowitz, and C.M. Reed. Information transmission with a multi-finger tactual display. *Perception and Psychophysics*, 61(6):993–1008, 1999.

[Tel] http://www.tellme.com.

[Tes81] L. Tesler. The smalltalk environment. *Byte Magazine*, 6(8):90–147, 1981.

[TN05] J. Tepperman and S. Narayanan. Automatic syllable stress detection using prosodic features for pronunciation evaluation of language learners. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05)*, volume 1, 2005.

[TP97] H.Z. Tan and A. Pentland. Tactual displays for wearable computing. *Personal and Ubiquitous Computing*, 1(4):225–230, 1997.

[TW06] Simon Tucker and Steve Whittaker. Time is of the essence: an evaluation of temporal compression algorithms. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 329–338, New York, NY, USA, 2006. ACM.

[TYB$^+$01] John C. Tang, Nicole Yankelovich, James Begole, Max Van Kleek, Francis Li, and Janak Bhalodia. Connexus to awarenex: extending awareness to mobile users. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 221–228, New York, NY, USA, 2001. ACM.

[VGM02] B. Verplank, M. Gurevich, and M. Mathews. The Plank: designing a simple haptic controller. In *Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–4. National University of Singapore Singapore, Singapore, 2002.

[VH99] S.C. Venema and B. Hannaford. Experiments in fingertip perception of surface discontinuities. *IJRR*, 6:1, 1999.

[VWT06] R. Villing, T. Ward, and J. Timoney. Performance Limits for Envelope based Automatic Syllable Segmentation. In *Irish Signals and Systems Conference, 2006. IET*, pages 521–526, 2006.

[WB96] M. Weiser and J.S. Brown. Designing calm technology. *PowerGrid Journal*, 1(1):75–85, 1996.

[WCM07] Jacob O. Wobbrock, Duen Horng Chau, and Brad A. Myers. An alternative to push, press, and tap-tap-tap: gesturing on an isometric joystick for mobile phone text entry. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 667–676, New York, NY, USA, 2007. ACM.

[Wei99] M. Weiser. The Computer for the 21 st Century. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3):3–11, 1999.

[WFB$^+$07] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: a see-through mobile device. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 269–278, New York, NY, USA, 2007. ACM.

[WHFG92] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992.

[Win] http://microsoft.com/windowsmobile/voicecommand/default.mspx.

[WWB$^+$86] RB Welch, DH Warren, KR Boff, L. Kaufman, and JP Thomas. Handbook of perception and human performance. 1986.

[YRD05] H. Yuan, C.M. Reed, and N.I. Durlach. Tactual display of consonant voicing as a supplement to lipreading. *The Journal of the Acoustical Society of America*, 118:1003, 2005.

[YZ06] Min Yin and Shumin Zhai. The benefits of augmenting telephone voice menu navigation with visual browsing and search. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 319–328, New York, NY, USA, 2006. ACM.

[ZDC$^+$07] Shengdong Zhao, Pierre Dragicevic, Mark Chignell, Ravin Balakrishnan, and Patrick Baudisch. Earpod: eyes-free menu selection using touch input and reactive audio feedback. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1395–1404, New York, NY, USA, 2007. ACM.