Supporting Everyday Activities through Always-Available Mobile Computing

Timothy Scott Saponas

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2010

Program Authorized to Offer Degree:
Department of Computer Science and Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Timothy Scott Saponas

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Co-Chairs of the Supervisory Committee:

____
James A. Landay

____
Desney S. Tan

Reading Committee:

____
James A. Landay

____
Desney S. Tan

____
James A. Fogarty

Date:______________________________________

University of Washington

Abstract

Supporting Everyday Activities through Always-Available Mobile Computing

T. Scott Saponas

Co-Chairs of the Supervisory Committee:

Professor James A. Landay
Computer Science and Engineering

Affiliate Professor Desney S. Tan
Computer Science and Engineering

Ubiquitous Computing provides us a vision of computing fading into the background and gracefully supporting our everyday activities. However, our interactions with mobile devices still demand so much of our physical, visual, and cognitive attention that mobile applications are tasks unto themselves, often interrupting the activities they seek to support. For example, the physical interfaces of portable music players are well designed for situations where people can see the controls while holding the device in their hand. However, when people use these devices to listen to music or manage their workout while exercising, they often are forced to interrupt their activity just to advance to the next song. As our primary use of computing continues to move off the desktop, we need new interfaces to mobile computing that expand the types of situations in which people can make use of computing. In this dissertation, I explore always-available interfaces to improve our current interactions and to enable new mobile computing opportunities. The experiments I have conducted and the systems I have built demonstrate my thesis that:

> *Finger level gestures detected and classified through forearm electromyography can enable an always-available interaction paradigm for mobile computing.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisors, James Landay and Desney Tan, for their amazing support. In particular, James has given me six years of guidance and support—always willing to talk and give me the best advice possible at any hour for any amount of time. James has always set the bar high and continuously pushed me to be better than I thought I could be. I would also especially like to thank Desney for taking a chance on me and not giving up the first time (or any of the times) it looked like I had broken something very expensive. Thanks guys! I could never have done it without your help.

I would also like to thank my wife, Dr. Meredith McLain Skeels (yes, you graduated before me), who not only makes every day of my life amazing, but also without whom I would spend all my time pursuing bad ideas – never discovering the better ideas that have made my research life successful.

My parents have also been a constant source of support as I have pursued my graduate work. At every turn, they have congratulated me on every accomplishment no matter how small. I also thank them for remaining interested in my research, reading my publications, and especially my dad for proofing this dissertation twice.

My dissertation research has been primarily supported by two sponsors who deserve recognition and thanks: Microsoft Research and UW Computer Science & Engineering.

I could not have accomplished the work in my dissertation without the help, advice, and encouragement of many people. In alphabetical order, I would especially like to thank Ravin Balakrishnan, Sumit Basu, Hrvoje Benko, Gaetano Borriello, James Fogarty, Andy Ko, Yoshi Kohno, Dan Morris, Babak Parviz, John Platt, Carl Ringler, Jay Stokes, Jim Turner, Dan Weld, and Jacob Wobbrock.

I would also like to thank the grad students at UW with whom I've enjoyed great friendship that has helped me endure six years of grad school. A special thank you to Jonathan Carlson, Richard Davis, Morgan Dixon, Kate Everitt, Jon Froehlich, Susumu Harada, Carl Hartung, Jonathan Lester, Kayur Patel, and Michael Toomim for being there to make me laugh, reflect on why we are still in school, and compete with me in HCI Lab dunk contests.

All of this research would not be possible without the participants who took time out of their lives to help me further understand and evaluate my technology. Thank you very much to all of you.

Finally, I would like to thank Gregory Abowd for introducing me to both research and HCI. Who knows where I would be today if he had not taken time drive a freshman to Radio Shack to buy some parts for a project. Thanks Gregory!

DEDICATION

To my parents, who inspire me to set the bar as high as I can.

Chapter 1
Introduction

Weiser's vision of Ubiquitous Computing (UbiComp) promises a world where computers fade into the background, seamlessly supporting our everyday activities (Weiser 1991). This vision has already been partially fulfilled; today's mobile devices allow us to do a wide variety of tasks, such as listen to music, get directions, and read email. Unfortunately, the human-computer interfaces of these devices require the full attention of our eyes and hands. Such interfaces limit our mobile computing interactions to situations where a physical device can be held in the hand conveniently. I believe the next step to realizing and extending Weiser's vision of UbiComp is developing new techniques to expand the input and output capabilities of our mobile devices, thereby enabling new opportunities for mobile computing to seamlessly support our everyday activities. Specifically, I explore always-available interfaces that enable us to use applications throughout our day when our hands are busy, when we cannot look at a screen, or when we do not have the time or desire to take a device out of our pocket.

Developing hands-free and implement-free input techniques is not a new idea; researchers have been working on "natural user interfaces" for years (Bolt 1980). Natural user interfaces are human-computer interfaces that make direct use of our natural abilities to move and speak instead

Figure 1-1: Marketing image of people playing the Kinect Sports game using the forthcoming Kinect Sensor for the XBOX 360. The Kinect Sensor is a no-controller approach to controlling video games employing computer vision using the combination of a depth camera and a traditional camera.

of relying on the manipulation of physical input devices like keyboards and mice. For example, advances in computer vision enable computers to recognize faces, track movement and gestures, and reconstruct 3D scenes. An example of this approach is the forthcoming Kinect Sensor for the Microsoft Xbox 360 gaming console. The paired devices use computer vision with a combination of a depth camera and a traditional camera to detect whole body and limb movement for control of video games (see Figure 1-1). One of the advantages of this approach to computer input is that games can be designed around body movements that mirror the actions of characters in a game; game designers are not restricted to mapping actions in a game to buttons on a handheld controller. Instead, games can be designed around body movements that mirror the actions of characters in a game. In this way, computer-vision systems like Kinect free users from the restrictions of handheld input devices.

Speech recognition is another technology that allows for hands-free interaction. A Bluetooth (wireless) headset connected to a mobile phone running speech recognition software can enable people to call contacts using only their voice (see Figure 1-2). Speech recognition also allows people to navigate remote telephone menus using a standard analog phone or dictate text on a microphone-equipped computer. A primary benefit of employing speech for computer input is that it does not require any physical motion (beyond moving one's mouth). In situations such as driving, where people's hands, eyes, and bodies are fully occupied with a non-computing task,

Figure 1-2: Marketing image of the Jawbone Bluetooth (wireless) headset for use with mobile phones and laptops.

people can still use speech for input even though physical input devices such as touchscreens are largely unusable and/or unsafe.

Although computer vision and speech recognition technologies have the ability to free users from the constraints of physical transducers, they have several inherent limitations that restrict the situations in which they are effective. First, they require fundamentally observable interactions that can be inconvenient or socially awkward in many situations. Second, they are relatively sensitive to environmental factors such as light and noise. Third, in the case of computer vision, sensors that visually interpret the environment are often susceptible to occlusion and are effective mostly in non-mobile scenarios where cameras can be placed in the environment.

In this dissertation, I propose a new approach that overcomes these limitations by utilizing the untapped bandwidth of the body through indirectly sensing input from our hands without encumbering our hands with physical transducers. The core of this technology is an armband worn on the upper forearm that can sense the muscle activations associated with hand and finger movement. To illustrate the role such an interface can play in future human-computer interfaces, consider this vision for how computing could weave into the fabric of our life:

> On a typical weekday, Karen wakes up early while her house is still quiet to go running. She usually listens to music while she runs. As she steps off the front porch and heads down the block, she pinches her index finger and thumb together to skip through the first couple of songs in her playlist until she finds a good song to start

her run. During her run, tactile feedback on her armband keeps her aware of how her pace compares to her fastest run this month. A little later, on her way back home, Karen sees her next door neighbor, an elderly lady who is out every morning walking her small friendly dog. Karen quickly pauses her music by pinching her ring finger and thumb together so she can hear what her neighbor is asking. They chat about the weather and holiday plans while Karen says hello to the dog. Afterwards, Karen stretches and goes inside.

While Karen was out, Bill woke up their daughter, Ashley (age 4), and their baby, Alex, woke up. While they get the kids dressed and eat breakfast, Karen is kept aware of how the stock market has opened and the local traffic conditions on her route to work by subtle tactile actuators on her stylish armband. Traffic doesn’t seem too bad today so she takes her time getting Ashley’s backpack packed for preschool and listening to her daughter’s stories about the class hamster. On mornings when the traffic is really bad, Bill takes the kids so Karen can get an earlier start and beat rush hour. On the way out to the car, Karen holds Ashley’s hand, her husband carries the baby and they both have their hands full of everyone’s bags, jackets, and coffee mugs. Karen squeezes her coffee travel mug a little harder and the back of the car pops open so she can load the bags and then help Ashley into her car seat.

After dropping the kids off at school, Karen heads to work at an asset management firm where she is an account manager. On the way to work she browses through the news using her dental retainer. Her retainer is loaded with the latest news stories about national politics, children’s health issues, markets around the world, and reviews of new books. Karen swipes her tongue across the back of her top front teeth to flip through the headlines. She listens to each headline through a bone conduction speaker in her retainer. When she hears there is new information about how kids develop allergies, she chooses that story and listens to the news report. She continues browsing the news as she walks into her building and takes the stairs up to her office, smiling and saying hello to co-workers along the way.

Figure 1-3: Potential Applications of Always-Available Interfaces.

After lunch, Karen heads off to a meeting with a client in an unfamiliar part of town. She turns on her portable navigation system and examines the map. She uses her armband interface to review her route by squeezing the steering wheel with her index and middle fingers to move forward and backward through the directions. Traffic in this part of town is quite heavy; so, while she's driving, she gets audio updates about her next turn to avoid taking her eyes off the road. She remembers that the driving directions said that after she gets off the freeway in two miles, there are a couple of quick turns. She again squeezes the steering wheel harder with her index finger to hear again what streets she'll be looking for when she exits the freeway.

During Karen's meeting with her client, her armband discretely keeps her aware of how much email she is receiving and the importance level of the emails. Midway through the meeting, she feels that her boss has sent her an urgent email. She subtlety sends a preformed email saying "I'll call you next chance I get" with a squeeze of her hands and a couple of finger pinches. Karen finds the next

a)  b) 

Figure 1-4: a) a jogger attempting to manipulate the controls of a portable music player strapped to his bicep while continuing to jog, b) a jogger using finger-pinching gestures to control his portable music player while jogging (exagerated for illustration purposes).

> opportune moment to briefly excuse herself from the meeting, takes her phone out of her bag, reads her boss's email, and calls her office.

Throughout this narrative, we see several scenarios where always-available interfaces enable Karen to better interact with her mobiles devices (see Figure 1-3). These interfaces increase the usability of input when her hands are busy, provide lightweight interactions that blend into her everyday life, and offer unobtrusive output to keep her aware of external information sources. A key building blocks to this future view of interacting with computing is a mechanism for providing *computer input that does not interfere with our everyday activities*.

To further illustrate the importance of this building block, consider how well a simple interface to a portable music player works when someone is standing still holding the player in one hand and manipulating the player's controls with the other hand. Now, compare that with how poorly the same interface functions when that same portable music placer is strapped to the person's bicep while they are jogging (see Figure 1-4a). Imagine instead someone being able to control their portable music player indirectly by just pinching their fingers together while jogging, even when their arms are in motion and even if they are wearing gloves (see Figure 1-4b). A human-computer interface based on the concept of people being able to pinch their fingers together and squeeze their hands into a fist can free them from the constraints of traditional touch-based interfaces without the drawbacks of "natural user interfaces" such as computer vision and speech recognition.

In this dissertation, I develop techniques, build tools, and demonstrate experimental results showing that such an interface can be created using electromyography (Merletti and Parker 2004) to sense finger level gestures from the upper forearm. In particular, the experiments I have conducted and the systems I have built demonstrate my thesis that:

> *Finger level gestures detected and classified through forearm electromyography can enable an always-available interaction paradigm for mobile computing.*

1.1 Hypotheses

In this dissertation, I demonstrate my thesis experimentally by testing the following three hypotheses:

1. Finger-level gestures can be detected via electromyography (EMG) on the forearm in real time with high-enough accuracy and low-enough latency for interactive applications.
2. It is possible to classify finger gestures in unconstrained, hands-busy situations.
3. Finger gestures can be detected through EMG with no new training or calibration after re-donning a muscle-sensing armband.

These hypotheses can be further broken down into the following research goals:

Goal 1: Addressing physiological constraints

- Finger-level gestures can be classified via EMG on the upper forearm
- Classification is robust to forearm rotation (supination and pronation)

Goal 2: Satisfying core gesture classification requirements

- Classification is done in real-time
- Whole hand squeezes can be detected reliably for turning on gesture detection mode
- Classification requires no new training after re-donning an EMG armband

Goal 3: Expanding the practicality of muscle-computer interfaces

- Gesture sets are not constrained to resting hand or arm on a surface
- Can detect gestures even when hands are busy holding objects
- EMG platform embedded in a wireless slip-on armband

1.2 Contributions

By experimentally testing these hypotheses and attaining the research goals I presented in the previous section, this dissertation makes several contributions that bring us closer to computing that fades into the background. Creating interfaces that allow people to interact with computers while their hands and eyes are occupied elsewhere enables people to utilize computing without having to interrupt their primary task or the people around them.

1.2.1 Concepts and Techniques

To achieve the larger vision of unobtrusive human-computer interfaces, I developed new techniques for detecting finger and hand gestures. First, I developed an approach that allows for finger-level gesture classification. With a ring of 6-8 electrodes placed on the upper forearm I can extract features from the incoming EMG channels to detect which fingers a person is moving. Creating a robust classification system also required refining a method for training a statistical machine learning model for use in finger classification.

Finger level gesture classification approach

- Upper forearm ring electrode placement
- A set of features extracted from 6 to 8 EMG channels
- Method for training a support vector machine using these features

Second, I developed a technique to detect activation of the device, so users can signal that they are about to provide relevant input. The device should be usable by people who are also using their hands to lift, hold, and manipulate objects, so creating and detecting an activation gesture enables the technology to know when to "listen" for user input. Activation detection required developing and iterating on a technique for differentiating an intentional whole-hand squeeze from the muscle activations typical during everyday activities.

Activation detection techniques

- Threshold approach with calibration
- Gradient detector with frequency band ratio-based constraints

Third, I created a bimanual interaction paradigm that enables users to do finger-gesture based control with one hand that is not accidentally triggered by combining these gestures with a selection gesture on their contralateral hand (using the activation detection mechanism above). This expands the situations in which people can use finger-level gestures while also giving people fine grained control over when the gestures are detected by the system.

Bimanual interaction paradigm

- Select plus activate concept
- Smoothing via majority vote with streak-based short circuiting

Fourth, my new technique for identifying finger gestures requires that we be able to distinguish not only which finger is moving, but also what that finger is doing. Fingers and hands can move in any number of ways, but I identified a specific set of gestures that can be robustly classified. These gestures can then be mapped to actions in a human-computer interface. For example, advancing to the next song on a portable music player could be the pinch of two fingers or tapping a finger several times.

Gesture sets

- Pressing, using pressure, lifting, and tapping on a horizontal surface
- Pinching with empty hands in free space
- Pulling and squeezing while holding objects

Fifth, I developed techniques for mapping my gesture recognition capabilities to interactions on an interactive surface. Knowing which fingers are being moved and what those fingers are doing allowed me to develop an interface where users draw in different colors, draw with different color intensity, undo operations on a drawing canvas, cut or copy objects from a drawing area, and throw objects onto a drawing area by simply using different fingers and different amounts of pressure.

Muscle-computer interfaces for interactive surfaces

- Drawing with different fingers
- Drawing with varying pressure
- Pinching with different fingers
- Waving
- Throwing

1.2.2 Artifacts

Accompanying the gesture recognition techniques described above, I also contribute software that enables EMG data collection, training of the system, model creation, and ultimately classification that can be used in specific human-computer interfaces. As my system's ability to classify gestures improved, I also collaborated on new hardware for a wireless sensor platform. There are many possible applications for muscle-computer interfaces; in this dissertation, I have developed two specific applications: Wireless Air Guitar and interactions with the Microsoft Surface.

The first artifact I contribute is the Physiological Sensing Platform. This is a collection of software including primarily a library called the Physiological Sensing Library (PhyLib) and a graphical interface to that library. The PhyLib contains implementations for capturing streaming EMG data, computing features of that EMG data, the ability to train and classify in real-time using several statistical machine learning techniques (primarily two external implementations of support vector machines), and the ability to run offline tests. The graphical interfaces to my PhyLib contains an interface for soliciting training data from a user, visualizing features extracted from EMG data, and a mockup of a portable music player.

A second artifact I contribute is a wireless muscle-sensing armband. After several physical prototypes, this armband consists of three two-channel wireless EMG boards that together deliver six EMG sensing channels. These boards are mounted on the outside of a sports sweatband and are connected to twelve dry silver-chloride disc electrodes sewn to the inside of the armband (two electrodes per EMG sensing channel). I also created embedded firmware for the EMG sensing board for collecting the EMG signal, compressing the data, and sending it over the wireless connection.

My third artifact is the combined software and hardware to support several implementations of the Guitar Hero video game, including my own simplified no-music version of the game. The hardware components of this contribution are an XBOX controller and a Guitar Hero guitar controller modified to be software controllable via a USB to serial connection. The software element to this contribution includes software to control the modified controllers, a network-based key relay for controlling the PC version of Guitar Hero, and my own simple version of Guitar Hero where there is no music and the notes are random sequences.

My fourth and final artifact is the software for relaying and combining the touch gestures of the Microsoft Surface with my muscle-computer input gestures to enable new interactions on interactive surfaces. One component of this software runs on a desktop computer where the muscle signal is being processed and the other component runs directly on the Microsoft Surface. They communicate over the network to provide the sensor fusion for the touch-plus-muscle-computer interface.

1.2.3 Experimental Results

At successive stages of the development of my muscle-computer interface system, I employed laboratory experiments as a means for evaluating the system's ability to classify finger-level gestures. From these experiments, my dissertation makes several contributions in the form of experimental results. The first experimental result from my work is that it is possible to classify finger level gestures using only EMG from the upper forearm. Another high-level experimental result is that we can classify many different finger-gesture sets including when hands are resting on a horizontal surface, in free space, and when holding objects. Additionally, my research contributes the finding that it is possible to classify pinching gestures from several forearm rotation positions (supination and pronation) if training data is gathered from all of those points of rotation. Lastly, I also demonstrate that it is possible to classify gestures based on EMG even when training the system with only a user's data from a previous electrode application on a previous day.

1.3 Research Approach

My research approach is an iterative process of investigating the possibilities and limitations of electromyography, developing a gesture classification technique, testing that technique, then trying to expand on that technique and apply it to interactive systems. More specifically, I started my work by investigating what type of information can be extracted from EMG when EMG electrodes are placed over muscles on various parts of the body. This eventually led to asking the question: is it possible to infer finger-level movement from a band of electrodes placed in a uniform ring on a person's upper forearm? I first answered this question by exploring several gesture sets in the constrained scenario of a person resting their hand on a table. This included gathering data, developing a technique consisting of signal processing, feature extraction, and machine learning, then testing that technique with more data gathered from pilot subjects. Finally, I conducted a formal evaluation of the technique with external participants. I iteratively followed this basic process throughout the rest of my dissertation research. I iterated on my initial findings by loosening the physical constraints, allowing the person's hand to come off the table and then giving the person objects to hold, and then refined my classification techniques to work in those situations. The final component of my research approach has been to start matching the constraints of my classification techniques to the interaction design of applications for muscle-computer interfaces. For example, I identified the Guitar Hero video game as a good demonstration of the latency and accuracy of my approach to muscle-computer interfaces and refined my techniques to work well within the scope of that application.

1.4 Dissertation Outline

The rest of this dissertation describes concepts, techniques, and artifacts I have created as well as the experiments I have conducted in the course of developing an upper-forearm-based muscle-computer interface. This research is an important building block toward always-available mobile computing. In Chapter 2, I survey related work in the areas of natural user interfaces, on-body and wearable interfaces, and muscle sensing with electromyography.

Chapter 3 describes an offline classification approach to detecting finger-level gestures when a person's hand is resting on a horizontal surface. I demonstrate the feasibility of this classification approach through a laboratory experiment with thirteen people. Next, in Chapter 4, I describe an updated real-time classification system for finger-level gestures that is effective when hands are in free-space (not resting on a horizontal surface), even when people are holding objects. I evaluate this real-time system through a twelve-person laboratory experiment.

I describe a novel wireless-EMG armband in Chapter 5 that moves closer to the vision of muscle-computer interfaces as practical interfaces for always-available mobile computing. I also demonstrate the ability to use this wireless armband to classify a person's finger-level gestures without any new training or calibration when they re-don the armband two days after originally training the system.

Chapter 6 describes two applications of muscle-computer interfaces. The first application is an interface to the video game Guitar Hero where I used my muscle-sensing system to create an "air guitar" experience for the game. The second application I describe is augmenting the touch-based interactions on an interactive surface using my muscle-computer interface to create new interactions.

I explore several areas of future work in Chapter 7, including on-body input, output, and several application areas. Finally, I conclude in Chapter 8 with a description of this dissertation's contributions and limitations as well as several reflections on the relationship between interaction design and the limitations of physiological sensing-based interfaces and also the challenges of conducting experiments with "brand new" interaction techniques.

Chapter 2
Related Work

In this chapter, I discuss previous work related to this dissertation in the areas of natural user interfaces, on-body interfaces, wearable computing, and electromyography.

2.1 Natural User Interfaces

In this section, I briefly review two types of so-called "natural user interfaces" that are relevant to the muscle-computer interfaces that I present in this dissertation.

2.1.1 Speech

Speech is a powerful mode of communication that people employ throughout their lives to interact with the people around them. Speech can come in many forms including whispering between two adjacent people in a movie theater or a single person speaking to thousands in an auditorium through the assistance of an audio amplification system.

A more recent use of speech is for people to be able to communicate with a computer. In a desktop computing environment, speech is primarily used for either dictation (substituting for a keyboard) or commands to control applications (substituting for a mouse). Dictation can be carried out using several commercial speech recognizers such as the Windows Speech Recognizer (Microsoft a) available in the Windows operating system since the Vista version and Dragon NaturallySpeaking (Nuance Communication) from Nuance Communications. Typically, speech recognizers can successfully recognize speech without any training from a new user, but these systems can also have their accuracy improved with per-user training. There are a variety of methods for using speech in the command and control of the desktop user interface and in applications. Three common methods are say-what-you-see, show-numbers, and mouse-grid. In these approaches either text or numbers are mapped to particular items or regions of the screen (Kamel and Landay 1999). Users use speech through those mappings to access those portions of applications. Another method of using speech for interacting with a desktop computing environment is a “speech cursor.” In this approach, people use spoken commands to steer the mouse cursor in one of eight cardinal and ordinal compass directions (“up”, “down”, “upper right”, “lower left”, etc) at varying speeds (“faster”, “very fast”, “stop”) and then click on the screen (“click”).

Another closely related technology to speech is non-linguistic vocalization. A non-linguistic vocalization is characterized by a person making a sound with their voice that is not directly part of a spoken language. One way to harness non-linguistic vocalizations for computer input is to map specific sounds to ordinal directions and use that mapping for steering in a user interface (e.g., steering a mouse cursor). The Vocal Joystick project pioneered the detection of and the mapping of vowel sounds from English for this purpose (Bilmes et al. 2006). Their steering mapping can be used for many applications including steering a mouse in a desktop computing environment (Harada et al. 2008), used for an image painting program (Harada, Wobbrock, and Landay 2007), or in computer games (Sporka, Harada, and Kurniawan 2007). In addition to mapping vowels to ordinal directions, they have also explored using vowel sounds to control continuous parameters in an interface in conjunction with a mouse or stylus (e.g., controlling the color, width, or opacity in a drawing program)(Harada, Saponas, and Landay 2007).

Speech can also be used outside of the desktop computer environment to interact with computing. For example, speech recognition is used for voice dialing on mobile phones (either through a headset or as integrated in a car’s audio system) and for automated call centers’ interactive voice response systems. Another set of speech-recognition applications are spoken language interfaces for use with conversational agents (including robots of many forms) (Bolt 1985; Bohus and

Horvitz 2009). Conversational agents have the potential to provide personal and informational assistance like functionality including providing help scheduling an appointment, navigating an area, triaging medical questions, and maintaining a to-do list.

These uses of speech recognition and non-linguistic vocalization have increased the situations in which people can interact with computers to include many situations where a keyboard, mouse, or touchscreen are inconvenient or awkward to use. However, speech as an input modality has several drawbacks that reduce its usefulness in some contexts. In locations where other people are present, it is frequently socially unacceptable to use speech for computer input. For example, in an open cubicle setting, during a meeting, while riding a bus, or when traveling by airplane, speaking out loud can be distracting and annoying to other nearby people. In other circumstances, speech can be inconvenient because it is somewhat difficult to speak clearly, such as while exercising or wearing a medical mask. Lastly, noise in the form of ambient sound can make speech recognition difficult to use in loud places.

2.1.2 Computer Vision

Computer vision techniques attempt to extract structure or information about the physical world from the output of camera technologies (Forsyth and Ponce 2002). A simple example of a computer vision technique is examining pixels of each frame in a video stream from a camera and determining if the camera is panning left or right. More complex examples include identifying where in an image are faces of people (Viola and Jones 2002; Rowley, Baluja, and Kanade 1998; Schneiderman and Kanade 2000) or trying to identify what object is in the frame (Huber et al. 2004). Computer vision techniques have been used to create many different types of interfaces for situations where handheld input devices are not desired. For example, hand tracking and pose estimation techniques enable hand orientation and gestures to be used for input (Bretzner, Laptev, and Lindeberg 2002; Lenman, Bretzner, and Thuresson 2002; Lenman, Bretzner, and Thuresson 2002; Rehg and Kanade 1994).

In the above examples, computer vision is typically used in conjunction with cameras mounted in the environment. Cameras can also be worn on a person pointed out toward the surrounding world and used with computer vision to provide several types of input. One example of this approach is using a lapel-mounted camera for egocentric recognition of handheld objects for detecting the activity in which a person is engaged (Ren and Philipose 2009). Another example of a person-mounted camera is a camera and projector combination hung from the neck for recognizing hand and finger movements for direct interaction with a projected display (Mistry and Maes 2009). Lastly, computer vision from person-mounted camera has also been explored in the form of a pendant with an infrared camera and illuminator combination for detecting hand gestures for interacting with the surrounding computing environment (Starner et al. 2000).

Computer vision is also used in some multi-touch interactive surfaces. One of these approaches that scales well to large display sizes is frustrated total internal reflection. In this approach, the user-interface is a sheet of acrylic with an LED light source injecting light into the side of the acrylic. A person interacts with one side of the acrylic and a projector projects an image on the other side. A camera on the projected side of the acrylic detects where people are touching the display by capturing the light from the LED light source scattered by a person contacting the screen (Han 2005).

Computer vision can also be used in combination with other sensors to enable richer experiences. A notable example of this is the use of several types of depth sensing technologies including stereoscopic systems (Czernuszenko et al. 1997; Agrawala et al. 1997; Wilson 2004), time-of-flight cameras (Iddan and Yahav 2001; Schuon et al. 2008) and infrared cameras in conjunction with a continuously-projected infrared pattern (PrimeSense). In the application area of hand and body recognition for user-interfaces, depth sensing technologies have been primarily used for manipulating a virtual 3D projected environment (Wilson 2007; Hilliges et al. 2009; Benko and Wilson 2009). The forthcoming commercial product Kinect for the XBOX 360 console gaming system promises to enable whole body interaction with gaming and possibly other types of applications using only computer vision and voice (Microsoft b).

Computer vision also plays an important role in human-robot interaction. In these systems, computer vision is used to recognize the faces of people and track the location and presence of people relative to a robot (Zhang, Rui, and Way 2006). This capability enhances the experience of interacting with a robot by allowing features such as the virtual eyes of a robot to follow the person with whom the robot is engaged in dialog.

While computer vision technology can enable compelling and immersive experiences, it suffers from several limitations that reduce the number of situations in which it is an effective mechanism for interaction. A primary drawback of this approach is that with the exception of pendant systems that hang around a person's neck or are pinned to a person's shirt, computer vision-based systems require the installation of camera sensors in the environment. This constraint limits computer vision-systems to the locations where it is appropriate and convenient to place cameras. Another constraint is that environmental factors such as occlusion and lighting can also greatly reduce the scenarios where computer vision is effective for tracking body movements and recognizing gestures.

2.2 On-Body and Wearable Interfaces

In this section, we survey related work in the areas of wearable computing and on-body interfaces. These two areas of work overlap in the sense that on-body interfaces can generally be used for controlling wearable computers. But on-body interfaces can also be used outside of a wearable computing context. Here, we review them in separate subsections using the distinction that on-body interfaces are those that directly sense something about the body.

2.2.1 Wearable Computing

The idea of people wearing some type of computing or imaging technology goes back at least as far as Vannevar Bush's 1945 article "As We May Think" (Bush 1945), where he suggested that in the future camera technology would become as small as a walnut, record up to 100 photos, and be worn continuously. We could also think of wearable computing being as old as Patek Philippe's 1868 invention of the wrist watch (Patek Philippe Museum), John Harrison's invention of the first marine watch in 1759 (Betts 2006) or even Peter Henlein's invention of the pocket watch in 1505 (Dohrn-Van Rossum 1996). Steve Mann, a pioneer in this area of research, defines the term "Wearable Computing" to mean computing that is "subsumed into the personal space of the user, controlled by the user, and has both operational and interactional constancy, i.e. is always on and always accessible" (Mann 1998). Over the last three decades many new human-computer interfaces have been created to support this general concept of having computing on us and with us all the time.

A component of many wearable-computing setups is a head-mounted display (HMD) (Baber et al. 1999). HMDs can come in many forms including both stereoscopic glasses that completely occlude a person's field of view and a thinner version that a person can peer above and below to see the physical world (Spitzer et al. 1997). Another HMD that does not suffer as much from field-of-view occlusion is the monocular display. Monocular displays typically come in the form of a small rectangular display that can clip onto one side of a traditional pair of glasses and offer a display to the side of one eye. More recently, researchers have also investigated self-assembly techniques to create very small LED based visual output on contact lenses (Saeedi, Kim, and Parviz 2008).

In addition to visual output, wearable computers also need some mechanism for input. For text input, corded keyboards offer people the ability to type with one hand in their pocket (Lyons, Plaisted, and Starner 2004). For smaller throughput needs, researchers have also investigated several other techniques that do not require a handheld device. A simple example of this is making use of gestures that consist of a person whacking their phone in their pocket with different parts of their hand (Hudson et al. 2010). A touch-sensitive bezel on a watch can also enable people to quickly touch and interact with their computing environment while engaged in other non-computing tasks much faster than pulling a phone out of their pocket to use a touch screen (Ashbrook et al. 2008; Ashbrook, Lyons, and Starner 2008). Similarly, Harrison, et al. has shown the possibility of interacting with a watch form factor without touching it at all by detecting a magnet that could be contained in a ring on a person's contralateral hand (Harrison and Hudson 2009).

While the concept of wearable computing has matured over several decades of investigation, the vision has yet to be fully realized. This dissertation presents a new method of computer input that integrates well into wearable computing and helps to enable the desired always-available nature of wearable computing. The approach in this dissertation improves on previous work on human-computer interaction for wearable computers by introducing an approach that can work in hands-busy situations and without needing to physically manipulate an input devices with one's hands.

2.2.2 On-Body Input

On-body input is the class of computer-input techniques that are physically attached to the body and attempt to detect, classify, or quantify something about a person's physiology for either implicit or explicit input to a computer.

There are many parts of the body that can be tapped for input. One such place is the brain. Systems that attempt to measure a signal in the brain for computer input are called brain computer interfaces (BCIs) (Tan and Nijholt 2010; Nijholt et al. 2008). There are many technologies that can be employed in detecting brain activity for input. A non-invasive technique that has been extensively explored is Electroencephalography (EEG). EEG is the method of detecting electrical activity associated with neuron firing within the brain from electrodes detecting small voltage differences on the scalp (Smith 2004). In HCI, EEG has been used for applications including task classification (Lee and Tan 2006), classifying working memory load (Grimes et al. 2008), object categorization (Kapoor, Shenoy, and Tan 2008), asynchronous control of a virtual car or motorized chair (Galán et al. 2008; Leeb et al. 2007; Zhao, Zhang, and Cichocki 2009), and even slow text entry (Scherer et al. 2004; Obermaier, Müller, and Pfurtscheller 2003). Another approach to BCIs that has received attention from researchers is functional near infrared spectroscopy (fNIRS). An fNIRS sensor consists of a series of near infrared emitters and a detector. The detector measures properties of the infrared light after it passes through brain tissue to measure oxyhemoglobin (HbO_2) and deoxyhemoglobin (HbR) in the blood. This technique can measure the content of these compounds in the blood with a temporal accuracy in the range of a few milliseconds and a spatial accuracy of approximately five millimeters (Sassaroli et al. 2008). In HCI, fNIRS have been primarily used for detecting mental workload levels (Solovey et al. 2009; Hirshfield et al. 2009; Girouard et al. 2009; Sassaroli et al. 2008).

Another type of on-body input is the Skinput system by Harrison *et al*. that appropriates the surface of the skin as an input device by detecting the propagation of mechanical waves caused by abrupt contact with the skin (Harrison, Tan, and Morris 2010). An example of how this approach can be used for input is projecting a phone’s number pad on a person’s arm and tapping on the projected buttons to dial a number. Alternatively, a system could also map a different input to each finger and a person could then enter the corresponding input by tapping on the ends of those fingers with their contralateral hand.

A simple physiological measure that can be used for indirect computer input is a person’s heart rate. One use of heart-rate measurement is to add a new mechanism for controlling interactive games (Nenonen et al. 2007; Magielse and Markopoulos 2009). Heart rate has also been used to enhance music-based personal trainers for use during exercise such as jogging (Oliver and Flores-Mangas 2006).

Another on-body input technique that could also be called an in-body input technique is harnessing the use of tongue movement for human-computer interfaces. This approach is of particular interest for providing effective modes of input for people with traumatic brain and spinal cord injuries or medical conditions such as amyotrophic lateral sclerosis (also known as

Lou Gehrig's disease). The most obvious way to exploit direct control with the tongue is to provide physical transducers the tongue can actuate or manipulate. For example, both Peng et al. and Salem et al. have created a joystick-like device that a user can control with their tongue (Salem and Zhai 1997; Peng and Budinger 2007). Similarly, a commercial device from New Abilities Systems has embedded pressure sensitive buttons into a dental retainer placed on the roof of a user's mouth (New Abilities Systems). Researchers have also explored techniques that do not require people to curl their tongue up to manipulate controls on the roof of their mouth; such systems attempt to track complex movements by instrumenting the tongue with metallic piercings or magnetic attachments (Struijk 2006; Huo, Wang, and Ghovanloo 2008). The movement of the attached elements within the mouth can then be detected either by a dental retainer worn in the mouth or by a separate device worn outside the mouth. These tongue-based approaches enable people to interact with their computing environment even if they have no other motor ability.

This dissertation adds to the above literature of on-body input techniques a new approach that enables detection of finger-level gestures through only muscle-sensing on a ring of electrodes on the upper forearm. Our muscle-computer interface approach fills a gap left by the above on-body input techniques of enabling people to use their hands for input even when they are looking away, holding objects, and wearing gloves.

2.2.3 Tactile Output

Our ability to perceive touch has been scientifically studied for at least two centuries. During the 1830's E.H. Weber, who coined the phrase Just Noticeable Difference (JND), used Aesthesiometers to measure the tactile sensitivity of the skin (Evans 1991). Over the last 50 years, researchers have extended previous understandings of human's sense of touch to include how and what kind of information can be conveyed using vibrotactile interfaces (Gallace, Tan, and Spence 2007). For example, many have shown that we perceive vibration differently at various sites around the human body due to the variation in tactile sensitivity in our different regions of skin. In particular, it has been shown that finger tips are much more sensitive than our palms or thighs (Cholewiak and Collins 2003). In fact, our ability to differentiate two points from one another range from 2mm on the glabrous skin of our fingertips to 40mm on the 'hairy skin' of our backs. On the arm, people can better localize vibration near the elbow and wrist (Cholewiak and Collins 2003). Around the body, the optimal patterns, directions, and timing for using linear vibrotactile arrays to draw lines have also been studied (Cholewiak and Collins 2000). In the shoulder area, Toney et al. examined the type and format of data suitable for tactile array interfaces (Toney et al. 2003). Similarly, Gemperle et al. built a shoulder harness with vibrotactors (Gemperle, Ota, and Siewiorek 2001).

There has been a large focus on tactile feedback for haptic interfaces for our hands and fingers because that is where we have the greatest ability to perceive "touch". Traditionally this has been used for simulation or virtual environments (Wall and Harwin 2001; Haluck and Krummel 2000; Lindeman et al. 2004). Recently, Li et al. developed techniques for creating sensations of rubbing and tapping (Li et al. 2008). The sense of touch in our fingers has also been used for mobile applications; Hoggan et al. have showed high recognition rates using rhythm and location of vibrotactile motors on a mobile device in a person's hand (Hoggan, Anwar, and Brewster 2007). This technology has been used for identifying callers in tactile mobile phone alerts (Brown and Kaaresoja 2006). Chang, et al. have also explored mapping pressure sensors on one person's handheld device to vibration on another person's device for private communication (Chang et al. 2002).

Much of the work in tactile interfaces has originated in using our sense of touch for accessibility technologies, such as sensory substitution for people with a sensory impairment (Kaczmarek et al. 1991). For example, researchers have explored adding or substituting senses. Bach-y-Rita *et al*. utilized the tongue as a human-input channel for sonar-like vision at night or for the blind (Bach-Y-Rita et al. 1969; Bach-Y-Rita, Tyler, and Kaczmarek 2003). Other work has explored using tactile interfaces for hearing impairments (Bernstein et al. 1991) and visual impairments (Easton 1992; Tang and Beebe 1998). Tactile feedback has also been used during rehabilitation to improve human motor learning (Lieberman and Breazeal 2007).

In addition to the hand-based haptic and accessibility uses of tactile feedback, there have been many efforts to use the human body's over 20 square feet of touch receptors for other human-computer interfaces. Three groups have created belts that employ vibration to indicate direction (Tsukada and Yasumura 2004; Van Erp et al. 2005; FeelSpace Belt Project). These belts overload a person's "touch" sensation around the waist to actually create a new sense of direction. This sense of direction is in the form of a constant awareness of their orientation to a place or active direction for navigation. Similarly, researchers have evaluated vibrotactile feedback on the torso for in-vehicle (Van Erp and Van Veen 2004) and in-building navigation (Lindeman et al. 2005).

Piateski and Jones have compared vibrotactile feedback on the forearm and torso for navigation, finding people much better able to recognize vibrotactile patterns on their torso (Piateski and Jones 2005). Yeow and Cheung have explored a tactile vest to convey direction, position, and velocity signals for pilots controlling rotary wing aircraft (Yeow and Cheung 2005).

Previous work in tactile output has quantified aspects of human perception, identified ideal positioning and methods of vibrotactile feedback, and suggested many applications on and around the body. This research can inform the design of future tactile output for mobile computing, which may become essential when the technologies for always-available input in this dissertation are more commonplace.

2.3 Muscle Sensing with Electromyography (EMG)

In this section, we briefly describe EMG and then survey related work employing EMG in prosthetics and robotics as well as the use of EMG in human-computer interaction.

2.3.1 Sensing Muscles with EMG

Human skeletal muscles are made up of muscle fibers attached to bone by tendons. These muscles contract to create skeletal movement. To contract a muscle, the brain sends an electrical signal through the nervous system to motor neurons. These motor neurons then transmit electrical impulses known as *action potentials* to the adjoining muscle fibers, causing the muscle fibers to contract. The combination of a motor neuron and the attached muscle fibers are known as a *motor unit*. Each muscle is made up of many motor units. During muscle contraction, some subset of a muscle's motor units is activated. The sum of all the electrical activity in a motor unit during contraction is referred to as a *motor unit action potential* (MUAP).

Electromyography (EMG) measures the MUAP as an electrical potential between a ground electrode and a sensor electrode (Kleissen et al. 1998). EMG can measure signals either directly within the muscle (invasive EMG) or on the skin above a muscle (surface EMG). Invasive EMG is very accurate in sensing muscle activation, but is impractical for human-computer interaction applications as it requires needle electrodes to be inserted through the skin and directly into the muscle fibers. Surface EMG, while less accurate, only requires that conductive sensors be placed on the surface of the skin. Surface EMG is fundamentally noisier than invasive EMG since MUAPs must pass though body tissues such as fat and skin before they can be captured by a sensor on the surface. Due to the high sensitivity of EMG sensors required to detect these signals, they also typically detect other electrical phenomena such as activity from other muscles, skin movement over muscles, and environmental noise.

The EMG signal is an electrical potential, or voltage, changing over time. The raw signal is an oscillating wave with an amplitude increase during muscle activation. Most of the power of this signal is contained in the frequency range of 5 to 250 Hz (Merletti and Parker 2004). A typical statistic computed over the raw EMG signal for diagnosis of muscle activity is the windowed root mean squared (RMS) amplitude of the measured potential. This measure has typically been employed for diagnostic purposes such as evaluating muscle function during rehabilitation after a surgery or for measuring muscle activation to assess gait (Lanyi and Adler 2004). RMS amplitude is a rough metric for how active a muscle is at a given point in time. For a full review of processing techniques used in previous work, see Naik et al. (Naik et al. 2006).

2.3.2 EMG for Prosthetics and Robotics

For over three decades, researchers have been working on using EMG as a means for amputees to use remaining muscles to control prosthetic limbs (Kiguchi, Tanaka, and Fukuda 2004; Jacobsen and Jerard 1974). Most research in this domain has focused on using the muscles of the upper arms and shoulders to control the gross orientation and grasp of a low-degree-of-freedom prosthetic device for manipulating objects (Jacobsen and Jerard 1974). Each measured upper arm muscle is typically mapped directly to one degree of freedom of the prosthetic. For example, tricep contraction could be used for rotation while bicep flexion might close or open the prosthetic. Yatsenko *et al*. has explored using 22 electrodes on the upper forearm to model the special distribution of EMG energies for single contractions (hand open/close, wrist pronation/supination, flexion/extension), pairs of contractions such as wrist pronation + extension, and triple contractions (Yatsenko, McDonnall, and Guillory 2007).

More recently, researchers have begun to look at the potential of using the forearm muscles in hand amputees to control a multi-fingered prosthetic hand. While I know of no fully functional hand prosthetic, this is clearly a promising new area of EMG research. One of the challenges for creating hand prosthetics is that there is not a trivial mapping of individual muscles to finger movements. Instead, many of the same muscles are used for several different fingers (Schieber 1995).

In tackling these problems, Jiang et al. used wavelet transforms combined with a neural network to classify thumb, index finger, and middle finger movements from several EMG sensors placed on the upper arm and forearm (Jiang et al. 2005). Similar research by Peleg et al. has examined the possibility of differentiating among individual finger movements (Peleg et al. 2002). They use auto regression features combined with a K-nearest-neighbor classifier to identify which of the five fingers is pressing a button. They fastened sensors to users' lower and upper forearms and immobilized the participants' arms by attaching their arms to a board. Tenore et al. explored using a neural network approach to classify 12 finger movement types with 4 bipolar EMG electrodes on the upper arm and 28 electrodes blanketing the forearm (Tenore et al. 2007). These three techniques require sensors in multiple places on the arms (lower forearm, upper forearm, and upper arm) and require users to be in a fixed posture.

2.3.3 EMG for Human-Computer Interaction

EMG sensing has been explored in HCI research for sensing emotion or affect through facial muscle activity (Mandryk, Inkpen, and Calvert 2006). This work utilizes the state of human facial muscles as a mirror for the human emotional state. Since the user is not intentionally controlling facial muscle activity, this work also showcases *implicit* computer input using EMG. In fact, since EMG is sensitive enough to pick up activity that is too small to result in actual muscle movement, this work demonstrates the potential to capture emotional state via EMG even when there is no externally visible representation.

Along similar lines, Costanza et al. have investigated subtle EMG-based interfaces (Costanza, Inverso, and Allen 2005; Costanza et al. 2007). Users of their system cycle through an auditory voice mail menu on a cell phone by flexing one or both biceps. They demonstrate that people can use this muscle input to discretely interact with a device without being detected by others. Wheeler, et al. explored using a sheath of EMG sensors on the forearm to recognize joystick movement and typing on a virtual number pad (Wheeler, Chang, and Knuth 2006; Wheeler and Jorgensen 2003). Naik et al. built a system that used EMG sensors distributed across the forearm to classify wrist, finger, as well as combined wrist and finger flexion (Naik et al. 2006). While these systems attain relatively good results, I believe that there are opportunities for extending gesture detection beyond the gross movements explored in their work. Also, wrist movements, used by this last project, are only suitable in a subset of applications, and finger activity will be a more natural mechanism for interacting with most computer systems. Other researchers have explored the space of finer motor movement by creating EMG input devices requiring only finger movement (Manabe 2004). However, this work requires that devices be worn directly on the fingers, which is somewhat intrusive and might interfere with normal finger activity. Similarly, the approaches above that require a sheath of sensors on the forearm or sensors in multiple areas including the bicep and wrists, could be difficult to combine into a comfortable and discrete wearable device.

In one of the only explorations into classifying finger gestures for potential use in mobile device control, Ju et al. has explored several machine learning approaches to classifying the pinch gesture (Ju, Kaelbling, and Singer 2000). They collect EMG data from electrodes near people's wrist while the participants performed pinching gestures with each of their four fingers and their thumb. They collected data from people on eight successive days. The results of their offline analysis demonstrated pinch classification accuracies as high as 78% when the gesture recognizer is trained with data from the same person in the same session. However, they found that in all of their approaches, recognition accuracies dropped off substantially when trying to train the system with data from other sessions or other people. This work was primarily an exploration of machine learning approaches to finger gesture classification using EMG data. In their work they only explored one gesture set and only with offline analysis. They did not explore any issues relating to interaction including any type of interactive system or potential use of gestures for a human-computer interface.

While previous work in EMG sensing of hand and finger movements has successfully demonstrated the ability to recognize several classes of gestures, we still know very little about how to use EMG sensing for human-computer interaction. There are very few attempts to employ EMG sensing in any real-time control of a computing application. None of the previous work addresses important HCI issues including whether their techniques will still work when people are physically active or rotate their arm into different postures. Previous work also does not explore how EMG might be used for input in mobile computing applications where people might wear the EMG sensors all day but only want to initiate EMG control of their device at particular times. All of these issues remain open problems for using EMG for HCI.

2.4 Summary

This dissertation extends and integrates with the previous work described above to move human-computer interaction closer to the goal of enabling people to use computing during and in support of every activity in their life. This dissertation’s muscle-computer interfaces are not meant to replace every previous form of mobile human-computer interface. Instead, they are meant to provide a new method of interaction suitable in more situations than was previously possible.

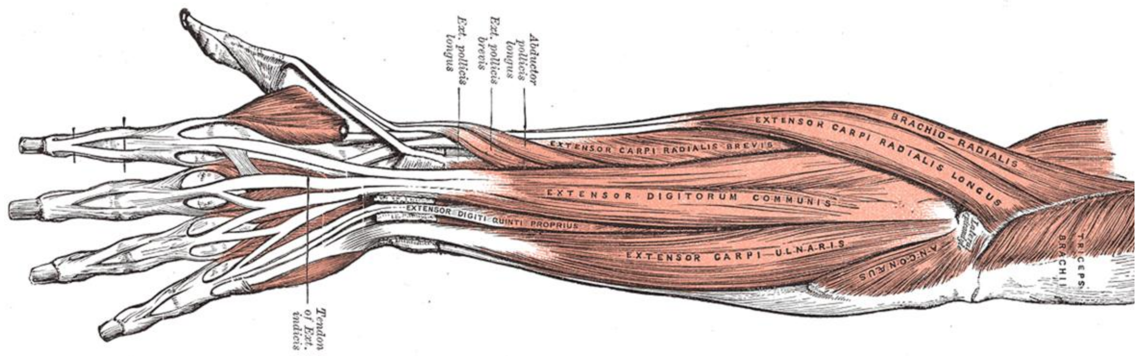

Figure 3-1: Posterior surface of the left forearm. Superficial muscles. Figure #418 from Henry Gray's Anatomy of the Human Body (Gray and Lewis 1918).

Chapter 3
Offline Classification of Finger-Level Movements from the Upper Forearm*

3.1 Introduction

Indirectly sensing finger movements offers us the opportunity to leverage the natural abilities of our fingers for computer input without actually encumbering our hands with physical devices. There are multiple approaches to this problem, such as computer vision or physiologic sensing. In this dissertation, I explore using a particular physiologic sensing technique: muscle sensing through forearm electromyography (EMG).

In this chapter, I describe our first experiment demonstrating the feasibility of using forearm EMG for muscle-computer interfaces. In our work, we chose to explore muscle sensing around the upper forearm. While there are many places throughout the arms and fingers where electrodes could potentially sense electrical activity related to finger movement, the upper forearm contains many muscles associated with finger movement (see Figure 3-1). We believe the form factor of a wireless band just below the elbow is feasible in the near future for muscle-computer interfaces

* Parts of this chapter are adapted from (Saponas et al. 2008).

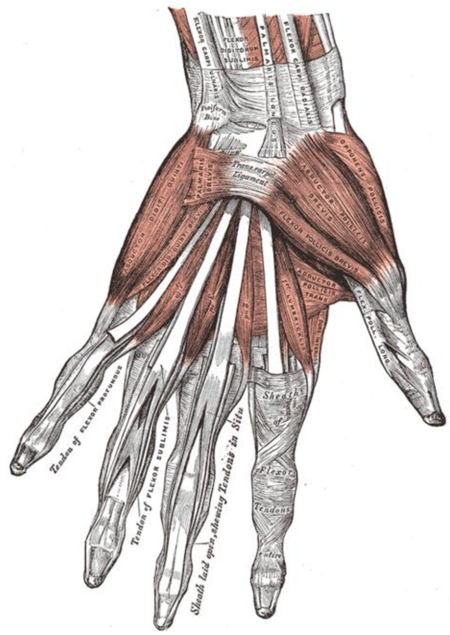

Figure 3-2: The muscles of the left hand. Palmar surface. Figure #427 from Henry Gray's Anatomy of the Human Body (Gray and Lewis 1918)

(see Chapter 5 for a first prototype of such a system). A band could be unobtrusive and worn either openly as a fashion accessory, or concealed beneath clothing. In either scenario, it would provide an input capability where traditional input devices are not feasible or desirable. We note that this form factor does not guarantee identical sensor placement as the user removes and replaces the armband; the results presented in this chapter assume that sensors do not move between training and decoding.

The experiment described in this chapter specifically explores offline classification of finger gestures with the hand resting on a horizontal surface. In this first experiment, we required users to rest their hand on a horizontal surface because we wanted to eliminate additional sources of noise in the EMG signal from arm movement. In later chapters, we explore a variety of gesture sets that do not require that a user rests their hand or arm on a horizontal surface.

3.2 Gesture Sets

We explored four distinct sets of finger gestures. We created these gesture sets to broadly explore how well our basic apparatus and analysis techniques could discriminate among various characteristics in finger movements that might be useful in real interface applications where someone’s hands are resting on a surface. Specifically, we tested whether we could classify

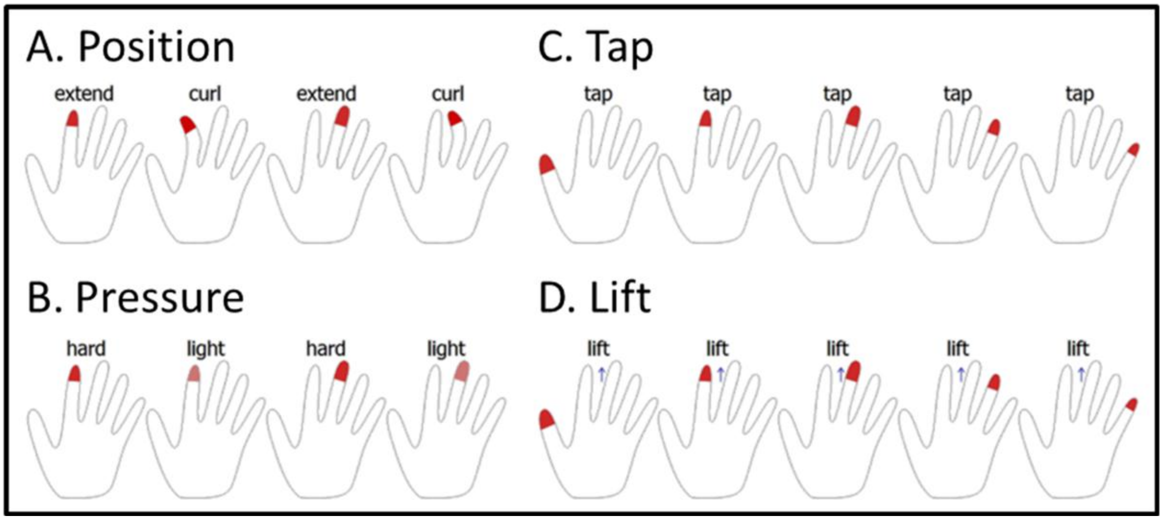

Figure 3-3: Stimuli representing the four sets of finger gestures tested during the experiment.

position and pressure of finger presses, as well as differentiate among all fingers during tapping and lifting.

Gesture sets that allow the independent use of all five fingers are particularly interesting in this exploration. Physiologically, since not all the fingers have independent muscle bundles that run down through the forearm (e.g., control of the thumb is mostly done through a strand that runs only to the palm and wrist. See Figure 3-2), we expected this to be a very difficult classification task, as demonstrated by multiple researchers who have tried before us and resorted to using only gross movements for control. However, we believe that the ability to classify among gestures performed by all five fingers independently is powerful in designing muscle-computer interfaces. Of course, not all gestures are easily performed independently by all fingers, and our choice of which fingers to study for particular gestures reflect these differences among fingers.

For an overview of the four gesture sets we used in our experiment, see Figure 3-3. The remainder of this subsection describes each individual gesture set.

3.2.1 Position: Index & Middle Finger

The four gestures in the *Position* set involved touching a surface with either the index or middle finger in one of two positions: extended or curled (see Figure 3-3a). Touching in the extended position is characterized as touching the surface with a finger while the hand is in the rest position. A curled position involves sliding the pad of the finger closer to the palm, which also causes the knuckle to rise, and touching the surface in that position. Participants were instructed to touch the surface firmly but not excessively hard.

3.2.2 Pressure: Index & Middle Finger

The four gestures in the *Pressure* set involved pressing lightly or firmly on a surface with either the index or middle finger (see Figure 3-3b). This was done in the extended position (i.e., from rest). We told participants that when "pressing lightly", they should apply enough pressure to dent a tomato, while "pressing hard" would break the skin of the tomato. Including the *rest* state, this gesture set has three levels of pressure.

3.2.3 Tap: Each of the Five Fingers

The five gestures in the *Tap* set involved individually tapping each of the five fingers on a surface from the rest position (see Figure 3-3c). This involved raising the finger slightly and then returning it onto the surface. Participants were asked to do this comfortably and not to exaggerate the gesture.

3.2.4 Lift: Each of the Five Fingers

The five gestures in the *Lift* set involved individually raising each of the five fingers in the air and holding it there (see Figure 3-3d). We asked participants to raise their finger only to the point where further movement would require additional force and not to exert any more force than was required to lift the finger off the table.

3.3 Experiment

Previous EMG-based input methods have classified gross movements such as wrist flexion (Wheeler and Jorgensen 2003; Naik et al. 2006) or bicep activation (Costanza, Inverso, and Allen 2005; Costanza et al. 2007), and hand prosthetics research has placed sensors on the forearm to detect finger movements (Tenore et al. 2007; Peleg et al. 2002; Jiang et al. 2005). However, these approaches involve a restrictive setup procedure including fixing the hand to a board or placing sensors at several places on the arm while recognizing only a few finger movements.

The high-level goal of this first experiment is to determine whether muscle-computer interfaces are even feasible using current EMG technology. This experiment is my first step in trying to employ EMG technology in such a way that muscle-computer interaction can be comfortable, unobtrusive, and useful for computer input.

Figure 3-4: Artist rendering of a forearm band with EMG sensors that could be used for muscle-computer interfaces.

Our approach is to place EMG sensors in a narrow band formation on the upper forearm. Our goal in starting with this sensor placement is to simulate an eventual platform resembling a thin wireless band worn just below the elbow (see Figure 3-4). As a starting point for muscle-computer input research, we attempt to computationally classify the four gesture sets described above. We compute three simple sets of features over an eight-channel EMG signal. These features are then used for classification using a statistical machine learning technique (Support Vector Machine).

We evaluate the extent to which we can classify gestures in this new muscle-computer input space through a laboratory experiment in which participants performed these gestures. We deliberately chose a simple EMG setup and off-the-shelf machine learning techniques in order to determine a lower bound on classification accuracy and ensure that our techniques are accessible to other researchers. If this simple setup results in reasonable classification ability, we should only be able to do better with improved hardware and more sophisticated classification algorithms.

3.3.1 Participants

Thirteen individuals (8 female) from the Puget Sound region volunteered for the experiment. Participants ranged from 20 to 63 years of age with an average age of 46. Participants were between 5′1″ and 6′1″ (1.55m to 1.85m) and weighed from 125 lbs to over 225 lbs (57kg to over 102kg). Most were daily computer users and played video games infrequently. None of the participants reported any existing muscular conditions or skin allergies, and all were right-handed. None were colorblind and all had 20/20 or corrected-to-20/20 vision. The experiment took approximately 90 minutes and participants received a software gratuity.

One participant remarked several times during and after the experiment that he “zoned out” and did not respond to the stimulus for sizable stretches of time. As such, we did not use this participant’s data, discarding it before we began our analysis. In the following sections, we report on analysis of the data collected from the remaining twelve participants.

See Appendix A for the complete demographic information and the demographic questionnaire.

3.3.2 Equipment and Setup

We used a *BioSemi Active Two* system for performing EMG sensing on participants’ forearms (Biosemi). This device employs a two-electrode active grounding system that drives the average potential of the participant as close as possible to the amplifier ground and reduces noise levels. The device samples eight sensor channels (labeled EX1 through EX8 in Figure 3-5) at 2048 Hz. While the capability of the Active Two to sense EMG is secondary to its primary electroencephalograph (EEG) function, the signals from this research device are comparable to most mid-end commercial surface EMG units used in medical settings. Sensor data was recorded to hard disk via a USB connection.

Before placing sensors on participants, we had them clean their upper forearms with an abrasive skin scrub, which helped ensure good conductivity and easy attachment. We then applied conductive gel to each sensor and attached them to the participants’ skin with a small adhesive circle around the gel. This cleaning procedure and the use of conductive gel are artifacts of the particular EMG equipment we used. Both can be obviated if dry electrodes, available with other manufacturers’ equipment, are used instead (e.g., see www.neumed.com).

Traditionally, EMG sensors are placed on the muscle belly, the largest mass of muscle that powers the particular movement of interest, in order to robustly measure the action potentials. Pairs of sensors are typically placed about an inch (~2.5cm) or so apart, in line with the muscle

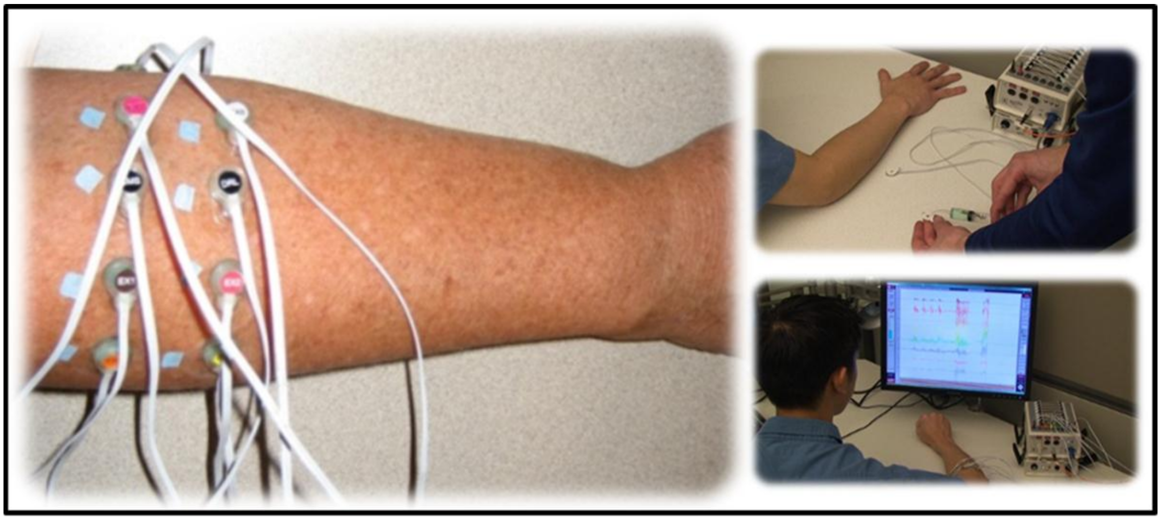

Figure 3-5: Sensors placed in a narrow band around a participant's arm using adhesive and conductive gel. The picture on the left is taken from an actual experiment.

fibers. Most current work also alludes to requiring intimate knowledge of surface muscle anatomy to get good sensor placement and clean signal.

In our pilot studies, we found that we could get reasonable signal even when we only approximately placed the sensors, especially when we had multiple sets of them placed around the forearm. Hence, we decided to place pairs of the eight sensors and the two ground electrodes in a narrow ring around each participant's upper right forearm (see Figure 3-5). We evenly spaced each pair around the portion of the forearm that did not rest on the table, with the ground electrodes in the middle. This configuration has encouraging implications for potential form factors of an approximately-placed armband EMG device (as illustrated in Figure 3-4). We draped sensor cables over the tops of participants' arms, allowing them to rest their arms comfortably on the table. On average, setup took a little less than 10 minutes.

Participants found a comfortable sitting position and viewed visual stimuli on a 21″ Samsung SyncMaster 214B LCD display. While we advised participants to minimize extraneous movements, we did not interrupt or stop them when they moved. Many participants occasionally stretched their hands, rocked their chairs back and forth, or tapped their feet. We believe that testing in such an environment is important as it mimics the realistic usage conditions of future muscle-computer interfaces.

3.3.3 Tasks

Participants performed four tasks corresponding to our four sets of finger gestures. Each participant began each of the tasks with his hand in a relaxed palm-down position on the table, which we refer to as the *rest* position. For most participants, this meant that fingers were mostly extended, but slightly curled. We asked participants to return to this rest position between gestures. Each task consisted of then performing gestures from that task's corresponding gesture set rapidly as prompted by a visual stimulus.

3.3.4 Design and Procedure

A within-participant design was used, with each participant performing all four tasks: Position, Pressure, Tap, and Lift. We counter-balanced the order of tasks across participants using a Latin square design.

We employed a randomized block design within each task. Each block consisted of a single repetition of each gesture within a set, presented in random order. We refer to each instance of a performed gesture as a trial. Participants performed multiple consecutive blocks for each task before proceeding to the next task. This block design is important as it ensures that gestures were relatively well distributed in time and that temporal bias (e.g., measurement drift) does not artificially inflate classification results.

For each trial, the participant was presented a visual stimulus that indicated the gesture to be performed (still images from Figure 3-3 presented on a computer screen). Participants were told to perform the appropriate gesture for the duration the stimulus stayed on the screen. In the Position, Pressure, and Lift tasks, stimuli were presented for two seconds. Tap stimuli were only shown for three quarters of a second, because a tap is a discrete gesture and cannot be "held down" or "held up." Participants were told to only tap their finger once per stimulus. A random delay between one and two seconds was inserted between stimuli to allow participants to return their hand to the rest position and refocus attention. Before each task, participants were given instructions and had approximately two minutes to practice performing the gestures with the prompting of the stimuli.

For each of the Pressure, Lift, and Tap tasks, participants completed 50 blocks with a two-minute break after 25 blocks were completed. Thus, each gesture for each of those tasks was performed 50 times by each user. In our pilot studies, we found the Position gesture set to be the most difficult to classify. Hence, we collected 75 blocks instead, resulting in 75 trials for each Position gesture, in order to have more training data for classification.

3.3.5 Data Analysis Techniques

To classify the signals measured with our EMG setup, we first perform some basic signal processing to transform the time series data into a time-independent data set. We then compute a set of features, which we use to train a support vector machine (SVM) (Burges 1998) and perform the classification. We discuss two metrics that quantify classification quality, one based on the accuracy with individual samples, and the other based on classifying entire trials. These steps are described in the following subsections.

Basic Signal Processing

We first convert the raw EMG data into a time-independent dataset. To do this, we adopt a technique used in EEG work (Kiguchi, Tanaka, and Fukuda 2004): we divide the raw EMG signal into 250ms segments and treat each 250ms segment as a single *sample* of EMG data. We then apply a band-pass filter to each sample between 2 Hz and 102 Hz, as pilot studies indicated that this is where most of the useful signal resides. We also filter out the 55 Hz to 65 Hz band to remove the 60 Hz noise that exists in most computing environments (e.g., from power lines and appliances). We use this approach as opposed to traditional EMG processing approaches used in prosthetics because finger level movements are driven by multiple muscle groups shared with the control of other fingers. Thus, it is important to capture the relationship between many EMG channels.

Feature Generation

For each 250ms sample, we generated three classes of features, which we use for training and testing the classifier. These classes were chosen based on prior work suggesting that they may be discriminative of various activities within electroencephalography and EMG signals (Merletti and Parker 2004; Raez, Hussain, and Mohd-Yasin 2006; Lee and Tan 2006).

- *Root Mean Square* (RMS) amplitude of the EMG potential is indicative of the amplitude of muscle activity close to a particular sensor. We first fully rectify the signal by taking its absolute value and then compute the RMS amplitude of each of the 8 channels. From these 8 base RMS features, we create another 28 by taking the ratios of the base RMS values between each pair of channels. These ratios make the feature space more expressive by representing relationships between channels, rather than treating each as being completely independent.

- *Frequency Energy* is indicative of the firing rate of muscle activity. The energy is often thought to be significantly affected by muscle energy and fatigue (Merletti and Parker 2004), and our pilot studies suggest it has high discriminative power in the activity space as well. To derive the frequency energy feature, we compute the fast Fourier transform (FFT) for each sample and square the FFT amplitude, which gives the energy at each frequency, and sum the energy of all channels into 10 Hz bins. This yields 10 frequency energy features for each sample.
- *Phase Coherence* measures the extent to which the EMG channels in a sample have a fixed relationship to each other (e.g., firing in similar manner). This is used extensively in electroencephalography work (Lee and Tan 2006) and pilot studies suggest it has discriminative power for EMG as well. As with the RMS values, we create 28 features by taking the ratios of the average phase between all channel pairs.

These calculations result in 74 features per sample (a one dimensional feature vector of length 74).

Classification

In our pilot studies, we explored several machine learning techniques for classifying the EMG signal into gestures. In those studies, we found that SVMs seemed to perform well (Burges 1998). SVMs are a set of supervised machine learning methods, which take a set of labeled training data and create a function that can then be used to predict the labels of unlabeled data. The labeled training data typically consist of input vectors of feature values and desired outputs. At a high level, SVMs map the input vectors to a high-dimensional space and attempt to create a set of maximal separating hyperplanes between the output variables, or classes. For our experiment, we used the Sequential Minimal Optimization (SMO) version of support vector machines (Platt 1998) as implemented in the Weka toolkit (Witten and Frank 2005). SMO is a fast method for training SVMs that breaks the typically large quadratic programming problem into a series of the smallest possible problems that are then solved analytically. This optimization minimizes computation time and memory use.

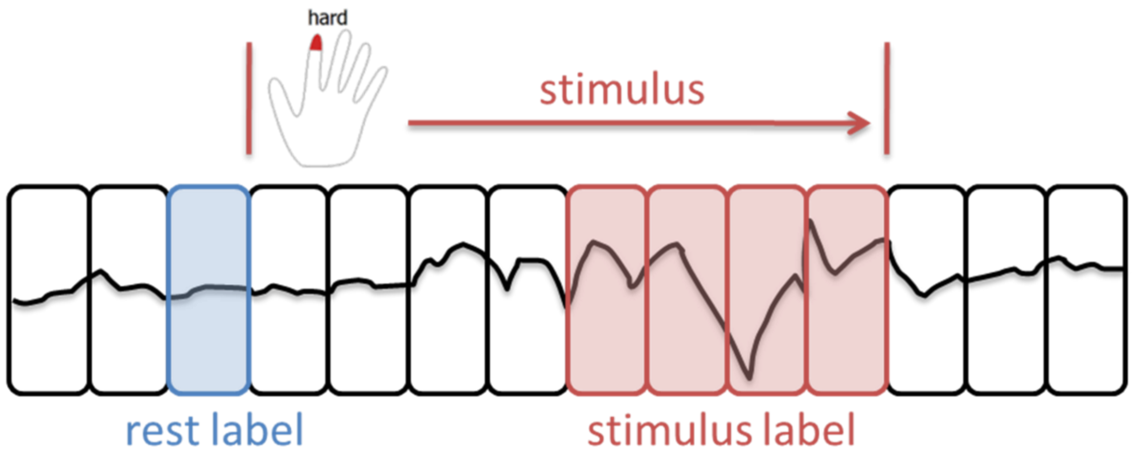

Figure 3-6: To mitigate variability introduced by participant reaction time, we trained and tested using only the sample before each stimulus was shown and the last four samples during stimulus presentation.

We used the default parameters for Weka's implementation of the SMO algorithm. Specifically, we set Weka's complexity parameter to 1.0 and the epsilon for round-off error at 1.0E-12, and we used a polynomial kernel function with exponent 1 and tolerance parameter of 0.001.

In labeling the training data we collected, we could have used values derived directly from the stimulus presentation (i.e., when the stimulus was showing a particular value, the data would be labeled accordingly), but we observed that participants responded to the stimulus with varying delay. Therefore, labeling based on the presented stimuli is more accurate toward the end of the stimulus presentation period. Hence we discard the first four 250ms samples (a total of one second per trial) and use only the next four samples for training. To acquire rest samples, we take the single sample that immediately preceded each gesture stimulus, as this was projected to be the time in which the hand had maximally recovered from the previous gesture but had not yet begun the next. Doing this also allowed us to have equivalent amounts of data from the rest and active conditions. See Figure 3-6 for an illustration of the data samples used. Our classifiers in this first experiment were trained and tested independently on data from each user from a single user session (except where stated otherwise).

3.3.6 Classification Results

We conducted a ten-fold cross-validation for tabulating classification accuracies for all four of our gesture sets. In each of the ten folds, 90% of the collected data was used for training and 10% was used for testing. The testing data were always taken as a single continuous chunk in time, as this is the most realistic scenario for testing such applications. The ten continuous chunks were chosen by splitting up the data by the first 10% temporally, then the following 10%, and so on. Performing a random holdout set, as is traditionally done, could have artificially boosted our results since test data samples would have been temporally adjacent to training samples. Using the training and testing methodology described above, we evaluated our classification accuracies for the four gesture sets.

Single-Sample Classification

Our first metric is the accuracy of single-sample classification. This metric provides a sense of the overall performance of the classifier. For the same reasons described above with respect to our training data, we only classify the sample immediately before each stimulus and the four samples at the end of the stimulus (see Figure 3-7). We do not try to classify the first four samples within the stimulus presentation period because we do not know when the physical gesture actually began due to variance in participant response times. The four samples we do classify are treated independently of each other. Classifying the sample immediately before stimulus presentation allows us to include the non-active or rest state as a condition in our classification.

In classifying single samples, our Position classifier performed at an average accuracy of 71% (sd: 9.0%), while Pressure classified at 76% (sd: 6.1%). In both cases, these results are much higher than the prior probability (the expected performance of a random classifier). For both of these tasks, prior probability is 20% because the classifiers were deciding among five conditions

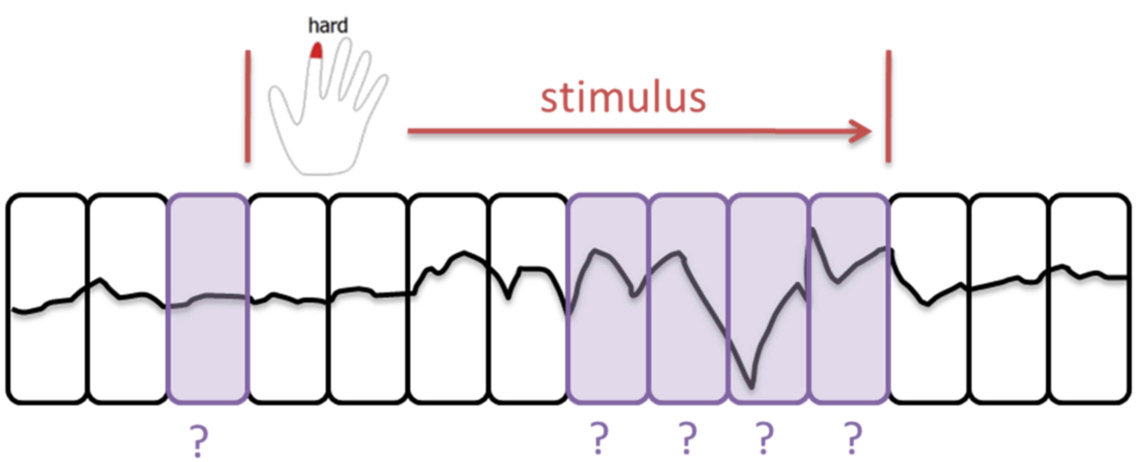

Figure 3-7: In single-sample classification, we classify each of the data samples independently.

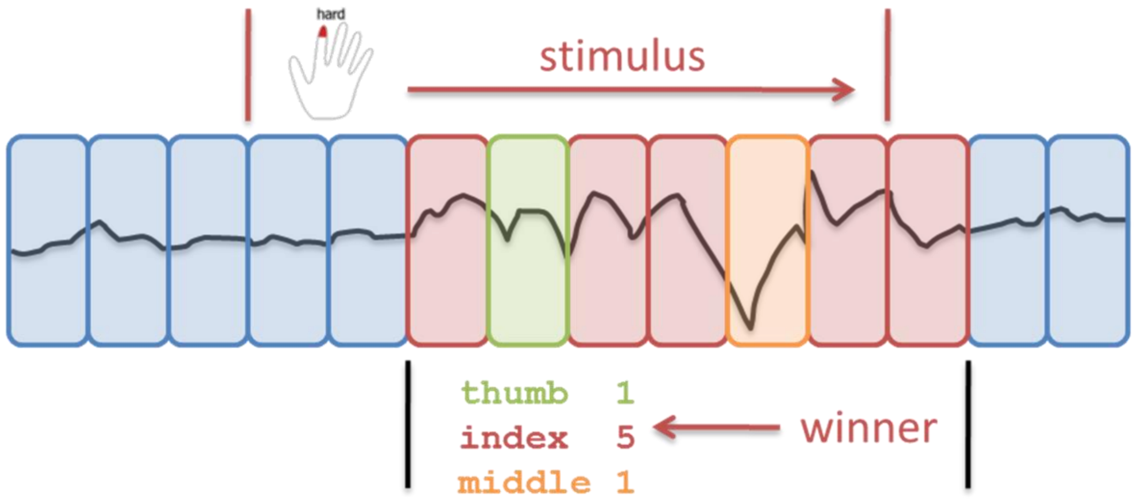

Figure 3-9: In whole-trial classification, we take a majority vote among classifications of individual samples in the largest non-rest region.

(four gestures and rest), each with an equal number of samples.

The Tap gestures were classified at an accuracy of 75% (sd: 6.9%) and Lift at 87% (sd: 7.0%). Again, these results are far above the prior probability, which was 17% since they were differentiating among six conditions (five gestures and rest). Per-participant accuracies for these results are shown in Figure 3-8.

Whole-Trial Classification

Our second metric is the accuracy of whole-trial classification. In a muscle-computer interface, we probably care more about classifying an entire gesture, or trial, rather than single data samples. As such, this metric provides us with a more ecologically valid measure of classifier performance. We consider a trial as the period starting three samples before stimulus presentation and ending three samples after (Figure 3-9). To classify a trial, we take a majority vote among classifications of individual samples in the largest non-rest region. When there is a tie, the vote that occurred later in time wins.

Accuracy increases for all tasks when we use the whole trial classification metric. With this metric, Position gestures are recognized at 78% (sd: 9.0%), Pressure at 84% (sd: 4.7%), Tap at 78% (sd: 10.5%), and Lift at 95% (sd: 6.4%). Figure 3-10 summarizes this data.

We note that if the classification problem is simplified by removing some of the gestures that are difficult to disambiguate, accuracy increases. For example, when all of the light pressure trials are removed from the Pressure task data, and the system only has to differentiate the firm index finger press, firm middle finger press, and rest conditions, it performs at an accuracy of over 90% (sd: 3.5%).

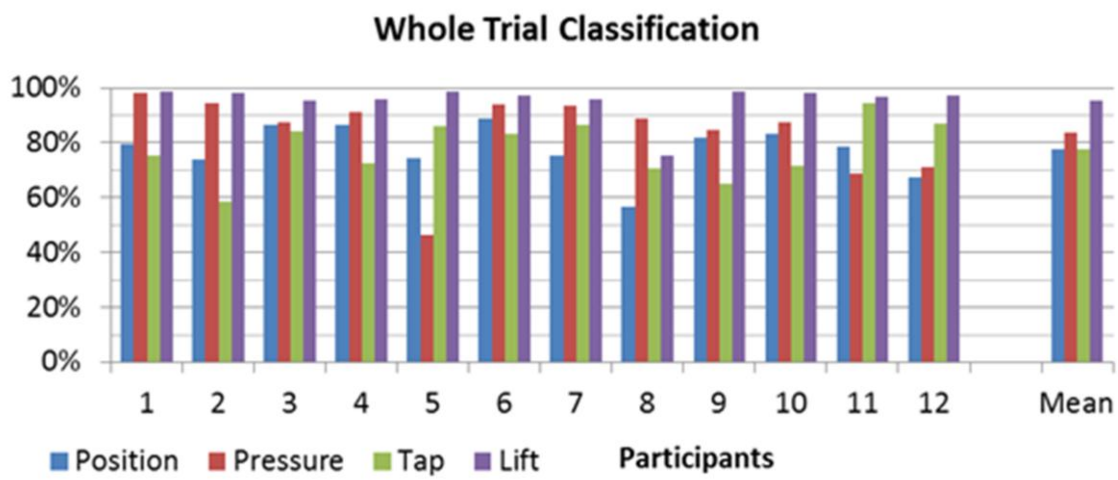

Figure 3-10: Whole-trial classification accuracies for all four sets of gestures, broken down by participant, with the mean result on the right side of the graph.

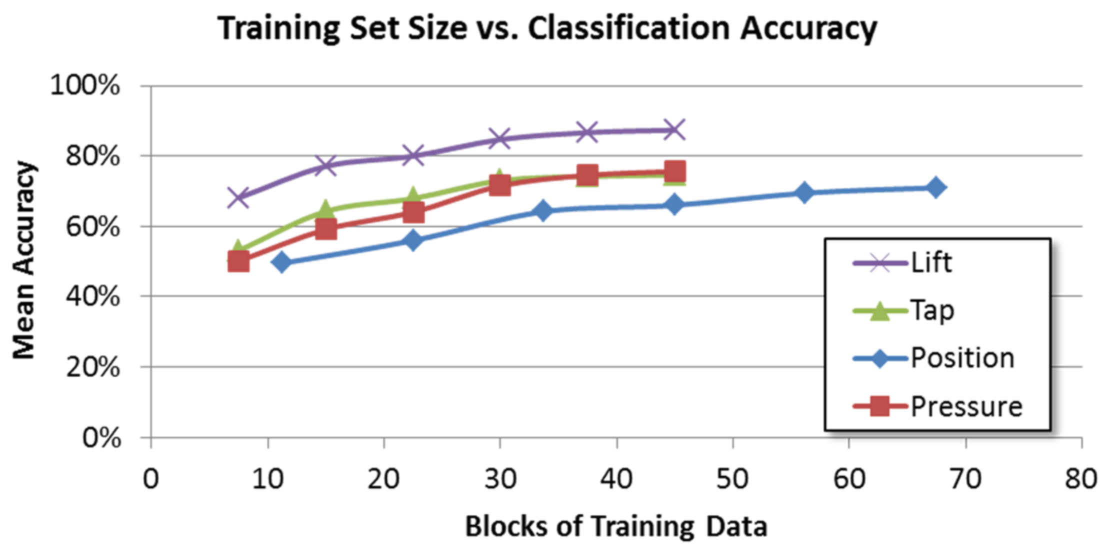

Figure 3-11: Decreasing the amount of training data degrades classification accuracy for all four gesture sets, but performance remains surprisingly high even with very little training data. A block of training data for a four-gesture set (position and pressure sets) takes approximately 14 seconds to collect. A block of a five-gesture set (tap and lift sets) takes approximately 17.5 seconds to collect.

Quantity of Training Data and Classification Performance

In the above evaluations, we used 90% of the data collected for training. We believe that it is important to understand the time-performance tradeoffs that exist in collecting training data, which has implications for the time required to configure muscle-computer interfaces in the future. Hence, we analyzed our data with various amounts of training data, incrementally removing samples from the end of the training data.

We found, not surprisingly, that decreasing the amount of training data decreases the accuracy of the classifier (Figure 3-11). However, we were pleasantly surprised at how slowly the results degrade as training data is reduced and how robust they are to the amount of training data available. In fact, each classifier's accuracy decreases only about 20% as we reduce the amount of training from 45 blocks to just 7 blocks. For example, Lift gestures are still classified at an accuracy of almost 70% with only two minutes of training data.

3.4 Discussion

In this section, we discuss several issues raised by the results of this first experiment.

3.4.1 Noisy Labels

One of the most significant limitations on our current classification results was the degree of inaccuracy in our labels when we trained and tested the gesture classifier. There were two main sources of noise in the labels.

The first type of noise is introduced by a variance in delay between stimulus presentation and participant response. This delay created many mislabeled data samples because the stimulus presentation could not be assumed to match the actual muscular response. While we mitigated the effect of this noise by using only the latter samples within a trial, we do not think this resulted in entirely correct labels, as variance between participants was quite large, with at least one participant who seemed to respond well beyond the one-second window of data that we did not classify.

The second source of noisy labels comes from participant error: in some cases, participants performed a different gesture than that instructed by the stimulus. Some participants remarked that they would try to anticipate upcoming stimuli and would make mistakes when they did not guess correctly. Hence, some percentage of our classification "errors" may actually be correct based on the gesture actually performed, but we do not know how frequently this occurred.

Both these problems could have been alleviated had we instrumented the participant or surface with physical switches, gesture sensors, or a camera so we would know exactly when a finger pressed down or lifted. We chose not to implement such a setup because we wanted participants to perform gestures unencumbered and in a manner that was comfortable and intuitive for them in order to maintain ecological validity with respect to our proposed application scenarios. While performing a study with such a control remains future work, we are encouraged by the surprisingly good results even in the presence of what we believe are fairly noisy labels, which only hurt our current results. We also believe that many of these problems would be minimized in an interactive system where users would have feedback indicating the system's classifications of their actions (e.g., the setup used in the experiment described in Chapter 4).

3.4.2 What Are We Measuring?

While traditional EMG work has been careful to ensure that the measured signal is derived from muscle unit action potentials, it is important to recall that these sensors do also detect other electrical signals. These signals include firing from distant muscles, such as the heart, environmental noise, and most importantly gross body movements. Many of these body movements cause the skin on which the sensors are attached to move as well, creating electrical signal.

While many of these extraneous signals would be considered too unreliable for use in traditional medical settings, we believe that some portion of our measured signals may actually be due to some of these "artifacts." We assert that these signals are perfectly legitimate, if not desirable, for use in muscle-computer interfaces that aim to classify overall activity. We believe that this approach of utilizing every bit of available signal is one of the reasons we are able to classify independent movement of all five fingers with such high accuracy, even though not all the fingers have muscle bundles that run down the forearm.

3.4.3 Cross-User Classification

In the results presented above, classifiers were trained and tested independently on data from each user with a fixed-placement of electrodes. That is, a new classification model was created for each user. In this scenario, every user has to train the system before it can recognize his or her gestures with each placement of electrodes.

To extend this result, we explored how well we could create cross-user models: models that are trained on data from one set of users and applied to classification on another. Being able to do this has implications for the robustness and invariance of the features we are using, and also for the potential of creating systems that require little or no user-specific training data. As an initial exploration of this problem space, we performed a twelve-fold cross validation in which we held out data from one user and trained on data from the remaining eleven. We attained reasonable though not stellar results. For example, we classify Lift gestures at an average accuracy of 57% (chance was 17%). It is not surprising that this accuracy is lower than for single-user models, especially since we did not do anything in sensor placement, experimental design, or data analysis to provide specifically for cross-user transfer. Despite these limitations, the classifier performed considerably better than chance, suggesting that there exists potential for building cross-user classifiers in this domain.

A more sophisticated approach to cross-user modeling might involve cross-validating internally within the training data to select the most invariant features from which to build the classifier. This remains future work.

3.5 Summary

The classification results presented in this chapter are based on single-handed input. A similar sensor band could be placed on the other forearm to provide even more input possibilities if both hands were resting on a horizontal surface. Since the classifiers would work nearly independently, we expect that this would approximately double the gesture space with little or no drop in accuracy. In fact, if we consider compound gestures involving both hands, we could imagine the input space growing quite significantly.

The index and middle finger are arguably the most viable initial candidates for use in muscle-computer interface applications that rely on position-based and pressure-based gestures, as it is difficult to independently control the fourth and fifth fingers for such gestures, and the thumb's abilities in this regard are qualitatively quite different from the other fingers. As such, our work focused on the index and middle fingers when evaluating position and pressure. The promisingly high classification accuracies we obtained indicate that simple muscle-computer interfaces could be designed that allow for the control of one or two input channels using the index and middle fingers. Examples include two-position sliders or switches controlled by extending and curling these two fingers, and three-state buttons controlled by varying the pressure exerted by these fingers (with button states corresponding to the "rest", "pressing lightly" and "pressing hard" gestures we evaluated in our experiment). The beauty of doing this with a muscle-computer interface is that these gestures can be performed anywhere and against any surface, whether on a table where one is sitting, one's own lap, or a car's steering wheel.

In contrast to position and pressure, simpler gestures such as tapping and lifting are reasonably independently performed by all fingers; hence our experiment evaluated all five fingers for these two gestures. Tapping and lifting with all five fingers are of particular interest, as our results indicate that we could conceivably design a muscle-computer interface that allowed any surface to be turned into a virtual keyboard operable by all five fingers (and potentially all ten fingers if a similar sensor band was worn on both arms). This has significant implications for text input where traditional keyboards may not be viable. Imagine sitting in a meeting where you are typing away with your fingers drumming on your lap with visual feedback obtained through your eye-glass display while others in the room are entirely unaware of your secondary activity. In fact, visual feedback might well be provided by a standard laptop screen. While typing on a laptop's keyboard might be distracting to other meeting attendees, typing via your muscle-computer interface and receiving feedback via a standard laptop display perhaps would not.

Our techniques and experiment show potential for unobtrusively sensing and decoding muscular activity for computer input. It is important to note that our results with classification accuracies as high as 95% in some conditions were achieved with off-the-shelf machine learning techniques, and with casually placed EMG sensors. It is reasonable to expect that even better classification accuracies could be achieved with more highly tuned machine learning algorithms and purpose-built sensors. We note, however, that our results were obtained with participants in a relatively sedentary position and gestures were performed beginning from a well-defined rest state. Chapter 4 investigates how classification accuracy changes in situations where the user is less sedentary and when gestures are performed while users are holding objects.

Chapter 4
Real-Time Classification of Free-Space and Hands Busy Gestures*

Previous work exploring muscle-sensing for input has primarily focused either on using a single large muscle (rather than the fingers) (Costanza et al. 2007; Englehart and Hudgins 2003; Farry and Walker 1993; Wang et al. 2006; Yatsenko, McDonnall, and Guillory 2007), which does not provide the breadth of input signals required for computer input, and/or on situations where the hand and arm are constrained to a surface (Wheeler, Chang, and Knuth 2006; Yatsenko, McDonnall, and Guillory 2007; Tenore et al. 2007; Peleg et al. 2002; Englehart and Hudgins 2003; Farry and Walker 1993), which is not a realistic usage scenario for always-available input devices. Chapter 3 demonstrated the feasibility of using offline machine learning techniques to interpret forearm muscle-sensing and classify finger gestures on a surface. This chapter extends these offline classification results to achieve online classification that enables using muscle-sensing for always-available input in real-world applications that are not constrained to a surface. The goal of this chapter is *not* trying to better understand or measure the physiology of human

* Parts of this chapter are adapted from (Saponas, Tan, et al. 2009).

musculature, but rather the goal is to further explore sensing muscle activity to enable interaction. Specifically, in this chapter:

- We leverage existing taxonomies of natural human grips to develop a gesture set covering interaction in free space, including when the hands are busy with objects, and even when hands and muscles are under load.
- We develop a procedure for rapidly and robustly calibrating an activation signal, present a system that classifies our gestures in real-time, and introduce a bi-manual "select and activate" paradigm that enables use in interactive systems.
- We demonstrate the feasibility of our approach through a laboratory experiment. Results show average classification accuracies of 79% for pinching, 85% while holding a travel mug, and 88% when carrying a weighted bag, all for four-finger gesture sets. Results suggest further generalizability across different arm postures.

4.1 Natural Human Grips

Most of the input devices we use for computing today take advantage of our ability to precisely operate physical transducers like buttons, knobs, and sliders. While this is an excellent approach when a computing device is one's primary focus, as in desktop computing, physical devices can be difficult or impossible to use when a user's hands or body are devoted to another activity. For example, a jogger may strap a music player to her arm or waist. However, even simple tasks such as changing songs, channels, or volume can be a struggle, requiring a user to reach across her body, possibly stop running, find the right button, and manipulate it. In circumstances such as these, where a user prefers to keep their hands free or is already holding something other than an input device, we propose that muscle-sensing offers an opportunity to take advantage of our manual dexterity without requiring physical actuation of a device.

To guide the design of muscle-sensing-based interaction techniques, it is important to consider the space of natural human grips and hand postures that we might leverage for gesture design. Over the last century, many grip posture classifications have been developed for biomechanical modeling, robotics, and therapy (MacKenzie and Iberall 1994). Schlesinger put forth the most well-known of these taxonomies (see Figure 4-1), characterizing six different manual grasps (Schlesinger 1919):

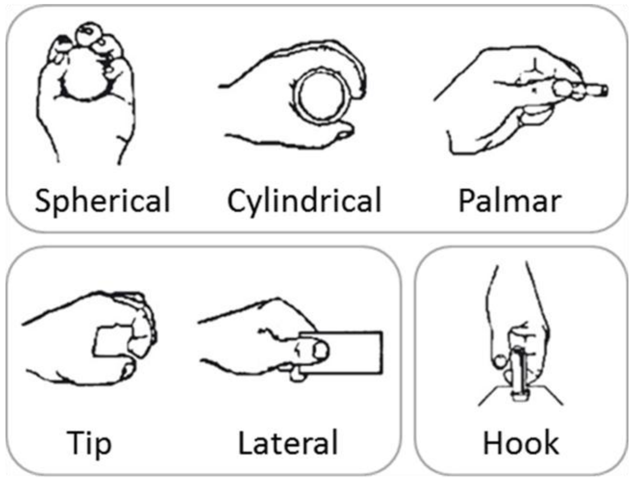

Figure 4-1: Schlesinger's natural grip taxonomies (Schlesinger 1919) as depicted in (MacKenzie and Iberall 1994). Groupings indicate the three similarity classes that guide our gesture set.

- **Spherical**: for holding spherical tools such as balls
- **Cylindrical**: for holding cylindrical tools such as cups
- **Palmar**: for grasping with palm facing the object
- **Tip**: for holding small tools like a pen
- **Lateral**: for holding thin, flat objects like paper
- **Hook**: for supporting a heavy load such as a bag

We explore techniques that will enable people to interact with computers when their hands are already being used in one of these grips, or when their hands are unencumbered but a handheld device is impractical. We divide these grips into three classes: small or no object in hand (tip and lateral), tool in hand (cylindrical, spherical, and palmar), and heavy load in hand (hook). Based on these three classes we suggest finger gestures, detect and classify these gestures in real-time using forearm muscle sensing, develop a two-handed interaction technique that allows for these gestures to control applications, and experimentally demonstrate the efficacy of these gestures.

4.2 Gesture Sets

In this chapter, we explore three separate but related gesture sets. Each gesture set contains four gestures corresponding to the four fingers: index, middle, ring, and pinky. The first gesture set consists of pinching one's thumb with one of the other four fingers (see Figure 4-2a). In our experiment, we specifically explore pinching with an empty hand held in the air near shoulder level with the arm at the side and elbow bent. This gesture could also be performed in other scenarios such as when wearing gloves, having hands at one's side, or possibly while jogging.

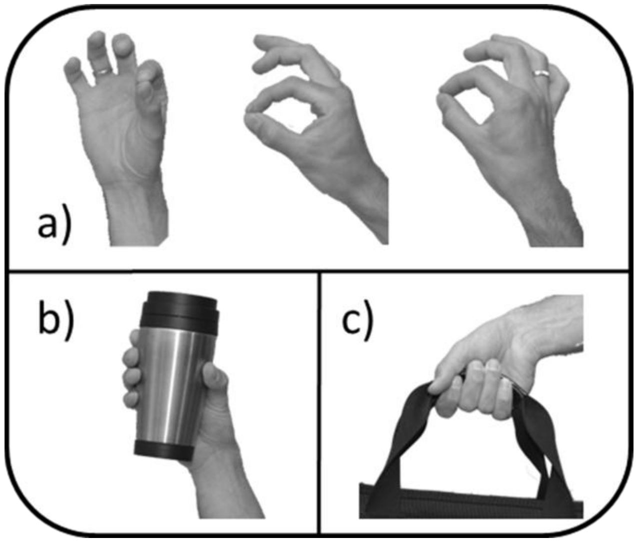

Figure 4-2: Our finger gesture sets. a) pinch gestures performed in three different arm postures b) fingers squeezing a travel mug c) fingers pulling up against the handle of a carried bag

The second and third gesture sets both consist of holding an object in one's hand while squeezing that object with one of four fingers. In the second gesture set, users hold a travel mug in their hand and each finger gesture is accomplished by squeezing harder with one finger than the others (see Figure 4-2b). Similarly, the third gesture set consists of holding a weighted briefcase by the handle and executing a finger gesture by pulling up with one finger more than the others (see Figure 4-2c).

4.3 Gesture Classification

In Chapter 3, we performed gesture classification offline. In this chapter, we classify gestures in real-time and begin to use our muscle-computer interface as a part of an interactive system. In moving to real-time classification and also not relying on people's hands always resting on a surface, we introduce several components to our classification technique including a calibration step and making our gestures bimanual. This subsection describes these additions and the improvements we made to our system subsequent to the experiment in Chapter 3.

4.3.1 Calibration

Before beginning any of the tasks in each session, we performed a short calibration step. Participants squeezed a ball for four seconds and then relaxed for another four. This calibration provided us with approximate maximum and minimum values across each channel and feature, which we used for normalizing the signal from each channel. Our normalization process was to scale the signal from zero to one based on the observed maximum and minimum value.

4.3.2 Activation Detection and Bimanual Gestures

In a real-world interactive system, determining when a user is performing a gesture and when he is not is crucial for preventing spurious detection of gestures and precisely labeling gesture onset or offset. This is particularly true if there is a strong timing component to the application, such as in games. Even in applications that do not have an intrinsic timing component, such as text entry, ambiguities in timing can yield incorrect results. For example, when switching from pinching with the index finger to the ring finger, a user passes through intermediate states, which may cause spurious or incorrect classifications of user intention.

Our approach to differentiating gesture from rest, and to simultaneously increasing the precision of gesture timing, is to introduce an explicit activation gesture. To do this, we use a second muscle-interface source, making a fist and squeezing the contralateral hand (in our experiments, we have our participants use their non-dominant hand for this activation gesture). Squeezing is an attractive choice as an activation gesture because it is a large multi-muscle action that can be robustly detected with consistent timing. Squeezing in itself is not sufficiently complex for most applications, thus we only use it as an activation gesture. By combining rich gestures performed with one hand and a robust but simple gesture performed with the other hand, we enable reliable and precise muscle-based interactions.

In addition to making the timing of input more predictable, using the non-dominant hand for gesture activation also allows the user to rapidly re-execute a single gesture many times in a row. For example, when scrolling through a list, the “down” gesture can be held for a second while the non-dominant hand makes several quick squeezes. This bimanual “select and activate” paradigm is the one we used in the testing phase of this chapter’s experiment.

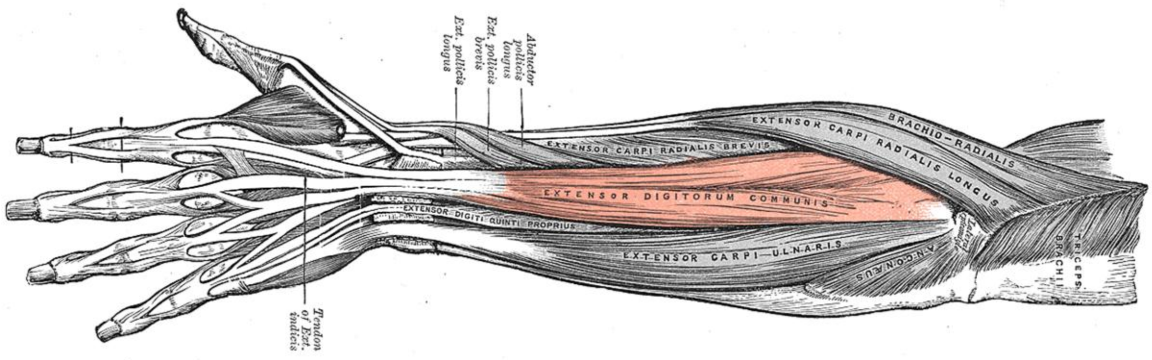

Figure 4-3: Posterior surface of the left forearm. Extensor digitorum communis highlighted. Modified from Figure #418 from Henry Gray's Anatomy of the Human Body (Gray and Lewis 1918).

Activation (Squeeze) Detection Technique

Our physical setup for detecting squeezing consists of placing two electrodes on the upper forearm: one on or near the extensor digitorum communis (see Figure 4-3) and another on the anterior side of the forearm near the elbow where there is little superficial muscle surface. We subtract the signal from these two sensors to achieve a resultant signal that is a low noise signal of the extensor digitorum communis. The extensor digitorum communis is used in extension of the index, middle, ring, and pinky fingers (which includes most forms of squeezing). This approach removes noise such as a person's cardiac electrical activity, giving a good estimate of the total muscle activity in the upper forearm. We then take the signal and discretize it into 32 non-overlapping segments per second (or approximately 31 millisecond segments). This results more frequent classifications over a shorter window data as compared with our approach in the previous chapter. This decrease in window size can make the system react faster in an interactive system. For each segment of time we compute the root mean square of the signal. The system then took any value above 40% of the maximum value seen during calibration to mean that the hand had been squeezed. We empirically selected 40% from results in pilot studies. The system would then "sleep" for a quarter-second before attempting to detect another hand squeeze. We enforced this "silent" period to prevent unintentional rapid sequences of selections.

4.3.3 Real-Time Classification Technique

To classify individual finger gestures from the EMG signal, we used a similar approach to the offline approach presented in Chapter 3, performing basic signal processing, computing a set of features, using those features to train a support vector machine (SVM) (Burges 1998), and then using that SVM to classify finger gestures. In this chapter, we use six EMG sensors instead of eight for detecting finger level gestures because two of our possible eight sensors are being used for squeeze detection on the other arm (the limit of eight sensors is an artificial limitation of the data acquisition box we used for this experiment). We also demonstrate that this approach can be used in a real-time system. The rest of this sub-section describes this process in detail.

Sliding Window Segmentation

Our first step is to convert the raw EMG data into a form suitable for our machine learning algorithm. Just as with squeeze detection, we divide the signal into 32 segments per second (about 31ms per segment). By dividing the data into segments, we transform it into a time independent dataset. We can then treat each of these segments as a single sample of data where we are trying to determine the gesture being performed.

Feature Extraction

For each 31ms sample, we generate three classes of features, which we use for training and testing the classifier.

The first set of features is the *Root Mean Square* (RMS) amplitude in each of the six channels, which correlates with magnitude of muscle activity. From the six base RMS features generated by sensors around the upper arm, we create another fifteen features by taking the ratio of the base RMS values between each pair of channels. These ratios make the feature space more expressive by representing relationships between channels, rather than treating each as being independent.

The second set of features is *Frequency Energy*, indicative of the temporal patterns of muscle activity. To derive these features, we compute the fast Fourier transform (FFT) for each sample and square the FFT amplitude, which gives the energy at each frequency. We create 13 bins over the 32 Hz sampling range for each of the six channels on the arm. This yields 78 frequency energy features per 32ms sample.

The third set of features is *Phase Coherence*, which loosely measures the relationships among EMG channels. We create fifteen such features by taking the ratios of the average phase between all channel pairs on the arm.

These calculations result in 114 features per sample for finger-level gesture classification.

Statistical Machine Learning

As in the experiment from the previous chapter, we employ support vector machines (SVMs). SVMs are a set of supervised machine learning methods that take a set of labeled training data and create a function that can be used to predict the labels of unlabeled data. For our experiment, we used the Sequential Minimal Optimization version of SVMs (Platt 1998).

Because people respond to a stimulus with varying delay, there is some amount of mislabeled information early within each stimulus presentation. As in Chapter 3, we combat this issue by discarding all samples from the first half of presentation and saving only the latter half as training data for our system. While classification results were generated 32 times a second, the system determined the currently recognized gesture at any given time as the last gesture classified three times in a row. For example, if the previous four samples were classified as “index, index, index, middle”, the system would use “index” as the currently recognized gesture even though the most recent classification result would be “middle.” We chose this approach to reduce sensitivity to momentary fluctuations in classification. Throughout this chapter, our classifiers were trained and tested independently on data from each participant during a single participant session.

4.4 Experiment

We conducted a laboratory experiment to investigate using forearm EMG to distinguish finger gestures within the three classes of grips: (1) small or no object in hand, (2) tool in hand, and (3) heavy load in hand. We collected gesture examples to train our recognizer and then asked participants to complete tasks using those gestures in a two-handed technique. For each part, we examine the average accuracies our system achieved in classifying finger gestures.

While each part of the experiment was conducted with a set of four finger gestures, we also present an offline analysis for each gesture set of a gesture recognizer that only uses the first three fingers (index, middle, and ring) to demonstrate the potential tradeoff of gesture richness against classification accuracy. We chose the pinky finger as the finger to remove in this analysis because participants reported that it was the most uncomfortable to manipulate.

4.4.1 Participants

Twelve individuals (5 female) volunteered to participate in the experiment. Participants ranged from 18 to 55 years of age with an average age of 36. All were daily computer users, and came from a variety of occupations. None reported existing muscular conditions or skin allergies, and

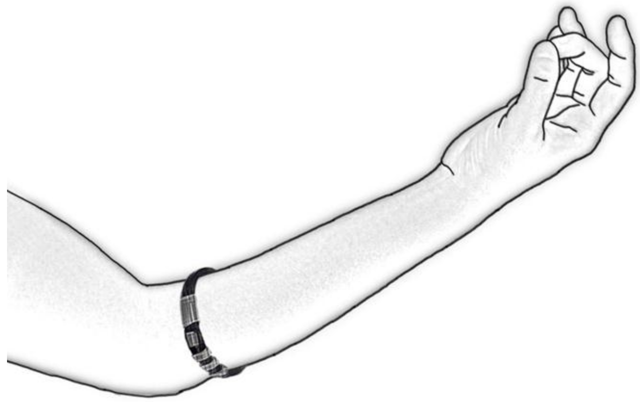

Figure 4-4: Artist rendering of a forearm muscle-sensing band.

all were right-handed. None were colorblind and all had 20/20 or corrected-to-20/20 vision. The experiment took 1.5 hours and participants were given a small gratuity.

See Appendix B for the complete demographic information and the demographic questionnaire.

4.4.2 Equipment and Setup

We used a *BioSemi Active Two* system as our forearm EMG sensing device (Biosemi). This system samples eight sensor channels at 2048 Hz. We first had participants clean their forearms with a soft scrub solution while we prepared the BioSemi sensors with conductive gel and adhesive.

To get the best possible signal, EMG sensing is traditionally conducted with two sensors spread an inch apart on a muscle belly. However, our results in Chapter 3 showed that it is possible to obtain reasonable results even when not precisely placing sensors. As such, we chose to place six sensors and two ground electrodes in a roughly uniform ring around each participant's upper right forearm for sensing finger gestures. We also placed two sensors on the upper left forearm for recognizing left-hand squeezes, or activation intent. This configuration mimics potential use with an approximately-placed armband EMG device, as illustrated in Figure 4-4. Setup took about 15 minutes.

4.4.3 Posture Independent Pinch Classification

The first part of our experiment explored performing finger gestures when the hands were not holding anything. Each participant performed pinch gestures with the thumb and one of the other

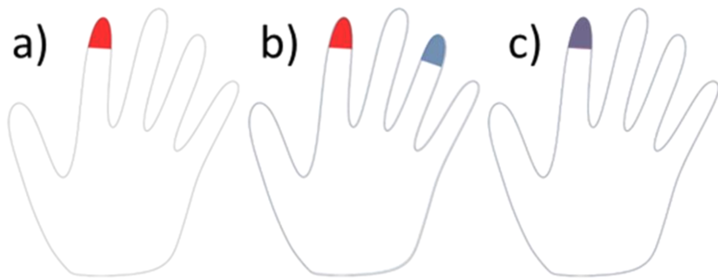

Figure 4-5: (a) A red highlight indicates that a gesture should be performed with the given finger; (b) a blue highlight indicates the currently recognized gesture; (c) a purple highlight indicates that the correct gesture is being performed.

fingers of their dominant hand. The gesturing arm was held in a comfortable position with a bent elbow and the empty hand held at about shoulder height (see Figure 4-4 and Figure 4-2a).

Without the constraint of a surface to rest on, people naturally move and rotate their arms and wrists between gestures. Doing so moves muscles under the skin relative to the attached sensors, creating changes to the observed EMG signals and potentially impacting classification. Most previous work has carefully constrained arm posture to avoid this scenario (for example, securing people's arm to a surface). However, this is an unreasonable constraint if muscle-computer interfaces are to be used for real-world interaction. Hence, we set out to examine whether or not our decoding techniques generalize to variable postures, and more importantly, how we can improve our techniques to better support posture variability. This sub-section describes an experiment to assess the viability of our approach to this problem.

Design and Procedure

We chose three different postures to explore: the two extremes of comfortable rotation of the forearm toward and away from their shoulder (pronation and supination) as well as a "natural" midpoint position (see Figure 4-2a).

Training Phase

Participants sat in a chair facing a desktop display. The system prompted participants to pinch each of their fingers to their thumb by highlighting the appropriate finger on an outline of a hand (see Figure 4-5a). We asked our participants to press "a comfortable amount". If they asked for clarification, we told them to "press hard enough to dent a tomato, but not hard enough to rupture the skin." They were told to relax their fingers when nothing was highlighted. Fingers were highlighted for a second, with a break of three-quarters of a second in between each stimulus.

We employed a block design, with each block comprising one trial each of an index, middle, ring, and pinky finger gesture, presented in random order. We gathered 25 blocks of training data for each of the three arm postures, the order of which was counterbalanced across participants.

Testing Phase

In the testing phase, participants performed 25 blocks of gestures in each of the three arm postures. As in the training phase, participants received their cues via a highlighted finger on the display. However, rather than timing their responses to the timing of the stimuli, participants were asked to perform the finger gesture with their right hand and "lock it in" by squeezing their left fist. To aid participants in this, we provided a small ball that they could squeeze with their left hand. The gesture could have just as easily been performed without the prop, as we demonstrate in the Hands Busy Gesture Classification part of the experiment (see Section 4.4.4). When the system recognized a squeezing movement with the left hand, it classified the gesture being performed with the right hand using the muscle-sensing data immediately preceding the squeeze.

Locking in a gesture by squeezing made the finger highlighting disappear for half a second, after which the system advanced to the next gesture. Since detecting the activation gesture is quicker and more robust than that of individual finger gestures, the bimanual paradigm allows for rapid selection of the same gesture multiple times in a row, as well as a robust way to avoid false positives.

Results

As describe above, variability in arm posture (particularly twisting of the forearm) presents a challenge for accurate finger gesture classification. To explore this issue, we trained the gesture recognizer in each of three postures independently, and performed an offline analysis testing each recognizer with the test data from the other two postures.

Table 4-1: Classification accuracies among pinch postures, averaged across all users. Chance classification for this four-gesture problem is 25%.

Train	**Test**		
	Left	Center	Right
Left	**78%**	72%	57%
Center	70%	**79%**	74%
Right	68%	73%	**74%**

As shown in Table 4-1, the system performed best when classifying pinch gestures using training data that was gathered in the same posture. Furthermore, training transferred more effectively between postures that were more similar. This can be seen by grouping these results by distance between training and testing postures (in amount of arm rotation). Distance zero represents training and testing on the same posture. Distance one represents a small rotation away, that is, either of the extremes to the midpoint or vice versa. Distance two represents training on one of the extreme positions and testing on the other.

The mean accuracy for distance zero is 77%, while distance one classifies at 72% and distance two at 63%. A univariate ANOVA on classification accuracy with rotation distance as the only factor shows a main effect of distance ($F_{2,105}$=5.79, p=0.004). Posthoc tests with Bonferroni correction for multiple comparisons show this effect driven by significant differences between distance zero and distance two (p=0.003) and marginally between distance one and distance two (p=0.086). Note that a random classifier would be operating at about 25% for the four-finger gestures.

However, when all of the training data is used (75 blocks) to train the gesture recognizer, instead of training data from a single posture, the average accuracy over all of a person's test data is 79% with a standard deviation of 13% (see Figure 4-6). This demonstrates that training in a variety of postures could lead to relatively robust models that find the invariants and work well across the range of postures. Exploring more complex methods of modeling posture independence remains future work. Reducing the gesture recognizer to just the first three fingers increased this accuracy to 85% with a standard deviation of 11%.

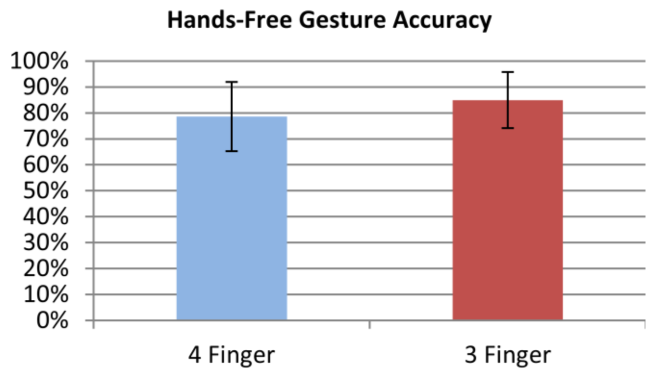

Figure 4-6: Mean classification accuracies for pinch gesture. Error bars represent standard deviation in all graphs.

4.4.4 Hands Busy Gesture Classification

The second part of our experiment explored performing finger gestures when the hands are already busy holding an object.

Design and Procedure

We looked at two different classes of objects for this part of the experiment. First, we used a travel mug to represent small tool-sized objects held in the hand. For this task, participants sat in a chair and held the mug in the air as one might naturally hold a beverage (see Figure 4-2b). Second, we tested larger and heavier objects being carried. Participants stood in front of a desk and carried a laptop bag in each hand (see Figure 4-2c). Each bag held a book weighing approximately one kilogram.

For both object types, we conducted a training phase and a testing phase. These were done one object type at a time and the order of the two object types was counterbalanced across users.

Training Phase

As before, participants performed 25 blocks of finger gestures in response to stimuli. The same stimuli highlighting fingers in the outline of a hand were used. Participants were asked to exert a little more pressure with the highlighted finger than with the other fingers. With the mug, this meant pressing on it a little more firmly with the highlighted finger than with the other fingers. With the bag, this meant pulling on the handle a little harder with the highlighted finger than with the other fingers. At the conclusion of the training phase for each object, the collected data was used to train the gesture recognition system for use in the subsequent phases. Once training data is collected, training the system requires only a few seconds of computation.

Testing Phase

In the testing phase of this part of the experiment, participants used the two-handed technique to perform gestures as they did in the Posture Independent Pinch Classification part of the experiment (see Section 4.4.3). However, in this part of the experiment, participants completed the stimulus-response task twice: once *with* visual feedback about the real-time classification, and once *without* visual feedback. The order was counterbalanced across participants and objects to avoid an ordering effect.

The "no visual feedback" condition was in the same style as the previous testing phase; a finger was highlighted and a participant would perform that gesture then squeeze with their left hand. When holding the travel mug, participants squeezed an empty left hand with their fingers against the lower pad of their thumb to "lock in" the current right-hand gesture. When holding a bag in each hand, participants squeezed the handle of the left-hand bag to "lock in" the current right-hand gesture.

In the "with visual feedback" condition, we added a second component to the display of the hand. In addition to the red highlighting of the finger that should be used in the gesture, the system also continuously highlighted its current gesture recognition result in a semi-transparent blue (see Figure 4-5b-c). We explained to participants that this was the system's best guess at their current gesture. We asked participants to perform the red gesture and activate their response only when they were confident it was correctly detected. As a side effect, visual feedback also allowed participants to understand the system's recognition behavior and to tailor their gestures accordingly. The goal of this manipulation was to explore the importance and tradeoffs of having visual feedback while using a muscle-computer interface.

Participants completed 25 blocks of gestures for each object both with and without visual feedback. The order of the feedback manipulation was balanced across the order of participants and objects.

Portable Music Player Experiment

In addition to testing the accuracy with which our system was able to classify gestures performed by participants, we also employed these gestures for use in a more ecologically valid application: a portable music player interface.

Our simulated portable music player (see Figure 4-7) is controlled through a hierarchical menu interface similar to those found in many mobile computing devices. Our player contained eight songs and only the songs menu was populated. The menu system can be navigated using four directional arrows where the "up" and "down" arrows move a selection cursor up and down in the current menu, while the "left" and "right" arrows navigate backward or forward in the menu structure. Forward navigation is also used to indicate a final selection at the end of a series of navigations. In music players, this corresponds to selecting a song.

We asked participants to control the portable music player menu interface and complete a series of tasks using our real-time muscle-computer interface. The training data collected earlier was used since the hands were similarly loaded with either the mug or the heavy bag. The user's inputs were mapped to the directional controller of the portable music player by assigning the index finger of the right hand to left, the pinky finger to right, the middle finger to up, and the ring finger to down. As in the other parts of the experiment, the left-hand grasping gesture was used to activate the gesture being performed by the right hand. The system continuously highlighted in red the directional arrow corresponding to the system's current finger gesture recognition result. This visual feedback told a participant what action the system would take if they squeezed their left hand at that moment.

Participants completed three different tasks with the portable music player. They (a) navigated from the top of the menu structure to the list of songs and selected a specified song, (b) navigated

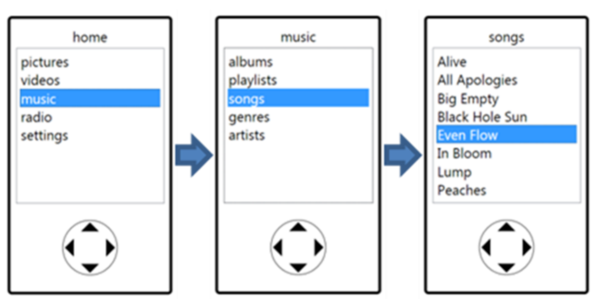

Figure 4-7: Software mockup of a portable music player

from a random starting point in the songs list to a particular song, and (c) advanced to the next song, starting at a random song in the song list. Task instructions, such as "Select Even Flow," were shown on the display above the music player interface. When a participant was presented with an instruction, they would then do a series of direction gestures to navigate the menu and select the song. Participants completed five blocks of these three tasks for each object (mug and heavy bag), for 30 tasks in total.

Results

Here, we describe the results from the hands-busy portion of our experiment.

Hands-Busy Finger Gesture Recognition

Participants performed finger gestures both sitting down with a travel mug in their hand and while standing with laptop bags in their hands. The system attempted to classify gestures both when the participants did and did not have visual feedback from the recognizer.

When participants held a travel mug in their hand, the four-finger recognizer attained an average accuracy of 65% without visual feedback (see Figure 4-8). Mean classification improved dramatically, to 85%, with visual feedback. A two-way ANOVA (finger × presence/absence of visual feedback) on classification accuracy revealed that the results with visual feedback were significantly higher than without ($F_{1,10}$=24.86, p=0.001). The system also classified much more accurately when only classifying among three fingers instead of four: 77% without feedback and 86% with feedback.

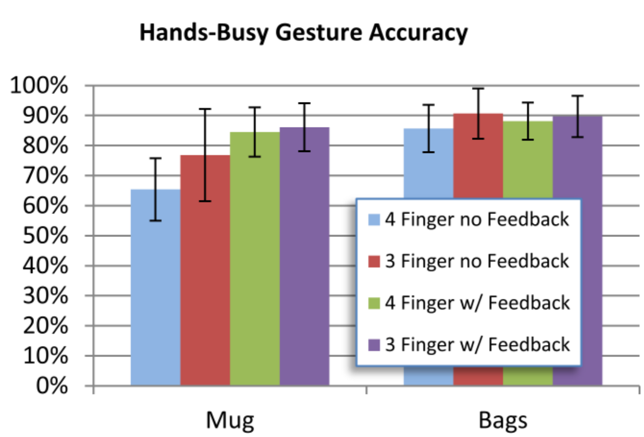

Figure 4-8: Mean classification accuracies of hands-busy gestures. Error bars represent the standard deviation.

Participants spent a mean of 1.61 seconds between gestures without visual feedback. This slowed to a mean of 3.42 seconds when they had visual feedback. An ANOVA revealed a main effect for feedback ($F_{1,10}$=13.86, p=0.004).

While holding a bag in each hand, the system classified participants' four-finger gestures at an accuracy of 86% without visual feedback and 88% with visual feedback (see Figure 4-8). When the classification was reduced to three fingers, the system's accuracy was better: 91% without visual feedback and similarly 90% with feedback.

On average, participants waited 1.69 seconds to squeeze their left fist when there was no visual feedback. This increased to 2.67 seconds when they had visual feedback of the system's current recognition result. A two-way ANOVA (finger × presence/absence of visual feedback) on completion time showed that the difference in feedback conditions was significant ($F_{1,10}$=19.77, p=0.001).

These results suggest that there is a time-accuracy tradeoff for visual feedback. Participants were probably spending time inspecting the feedback and making corrections to increase overall accuracy. In future work, we would like to explore less intrusive methods of providing feedback.

Portable Music Player Application Recognition

In the portable music player application, participants completed five blocks of three tasks with both the mug and bags. For each of these tasks, we recorded whether they selected the correct song, how many navigation steps they used above the minimum steps required to select the correct song, and how long it took them to complete each task.

In the travel mug scenario, two of the participants found that the system's classification of their pinky finger did not work well enough to effectively complete the portable music player tasks. We removed this data from our analysis.

When navigating the three-level hierarchical menu to select a song, participants on average selected the correct song 85% of the time with bags in their hands and 87% of the time while holding a travel mug. A failure was selecting any song besides the one specified. On average participants spent 45 seconds (median 39 seconds) navigating the menus through an average of 15 gestures per task with bags, and 58 seconds (median 40 seconds) navigating the menus while employing an average of 14 gestures with the mug. The goal of this phase of the experiment was to demonstrate that our real-time recognition system functioned well enough to be used in an interactive system. Among our participants some found it somewhat difficult to control the music player, while several stated that it worked very well for them and were interested when this might be released as a commercial product.

4.5 Summary

We have explored the feasibility of building forearm muscle sensing-based finger gesture recognizers that are independent of posture and shown that these recognizers performed well even when participants' hands were already holding objects. In this section, we discuss the implications of these results for application design.

4.5.1 Posture Independence

The results from the first part of our experiment suggest that while training data from one arm posture is most useful in recognizing gestures in the same posture, it is also possible to use our techniques to train a single gesture recognizer that works reasonably well in multiple arm positions. This suggests that electromyography-based interactions could be deployed without constraining wrist and hand positions. We feel that this is a major step toward enabling real-world applications, particularly applications in mobile settings. Users interact with mobile devices in a variety of body postures (seated, standing, walking, etc.), and we would therefore expect a similar variety of postures in the gesturing hand. Requiring a user to train a separate classifier for multiple hand positions would be costly; hence, we are encouraged by our results demonstrating the feasibility of cross-posture training.

4.5.2 Hands-Busy Interaction

Traditional input modalities take advantage of our dexterity, motor ability, and hand-eye coordination. However, in many scenarios we have to choose between our everyday behavior and manipulating a physical input device. In these scenarios, muscle-computer interfaces leveraging gestures that can be performed while our hands are already gripping an object provide an opportunity for computing environments to better support hands-busy activities such as when using a mobile phone while walking with a briefcase in hand or operating a music player while jogging. The results of hands-busy portion of our experiment demonstrate the possibility of classifying gestures involving individual fingers even when the whole hand is already engaged in a task, and even when the arm is supporting a heavy load.

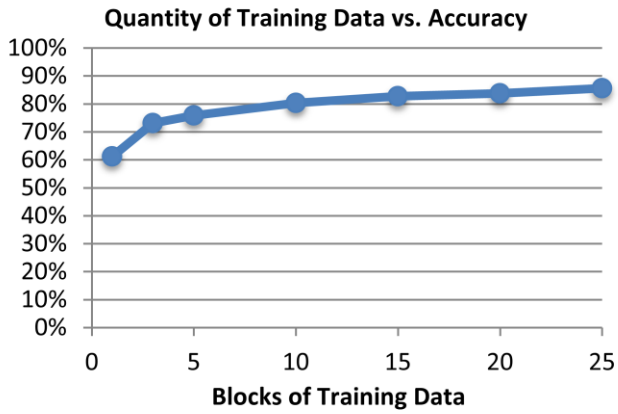

Figure 4-9: Classification accuracy versus blocks of training data for four finger gestures with bags in hand. Each training block takes seven seconds for a four finger classifier.

4.5.3 Quantity of Training Data and Classification Accuracy

Figure 4-9 shows that even with limited training data (10 blocks or approximately 70 seconds), average accuracies exceed 80% for four-finger classification, suggesting that the required amount of training for a muscle-computer interface would be on par with that typically required to train a speech recognition system. Future work will explore building cross-user models that would allow instantaneous use of our system without per-user training, leveraging per-user training only to enhance performance.

4.5.4 Cross-User and Cross-Session Models

We trained and tested our classifier for a single participant in a single session as is common with similar technologies such as brain-computer interfaces (Lee and Tan 2006). Future work will evaluate the degree to which classifiers can be re-used across sessions, and will focus on automatically configuring a classification system without careful sensor placement.

4.5.5 Interaction Design Issues

Even if a system can recognize individual gestures with reasonable accuracy, deployment in real-world scenarios still requires careful consideration of appropriate interaction techniques. Here we explore some of the design issues related to using muscle-computer interfaces for input.

Visual Feedback: Speed and Accuracy

Our experiments demonstrate that the proposed gesture set can be accurately recognized via muscle-sensing in the absence of visual feedback, which is critical to many applications, including nearly all hands-free mobile scenarios.

However, visual feedback makes the system more predictable and gives users an opportunity to adapt their behavior to that of the recognition system. For example, participants could experiment with finger position or exertion to improve recognition. This can be seen in the hands-busy part of our experiment where participants held a travel mug in their hands. The average accuracy of the system was much higher when participants had visual feedback. However, this came at the cost of reduced speed. On average, participants spent more time performing each gesture, as they adjusted their gestures until the system made the correct classification. This speed-accuracy tradeoff should be considered carefully in the context of an application. In applications where an error can easily be undone and the gesture repeated (e.g., in a mobile music player), the higher speed that comes from feedback-free gesture input may justify an increased error rate. In contrast, in applications where an incorrect gesture might be more costly (e.g., when controlling a mechanical device or playing a game), the decreased speed that comes from using visual feedback might be reasonable.

Engagement, Disengagement, & Calibration

A wearable, always-available input system needs a mechanism for engaging and disengaging the system. We do not want the system to interpret every squeeze or pinch action as a command. In our experiment, we used the left hand to support engagement and disengagement, and we feel that this separation of tasks across the two hands is a reasonable option for real applications. However, it would be worthwhile to look at how engagement and disengagement might be supported by sensing only one hand. In particular, is there a physical action unique enough to be robustly classified during everyday activity such that it can be used as an engagement delimiter? One example of such an action might be squeezing the hand into a fist twice in succession. In our limited exploration of this topic, a fist clench has appeared to be easily distinguishable among other typical movements, so this may be a starting point for future muscle-computer interfaces.

Multi-Finger Interactions

Our experiments focused on recognition of single gestures performed one at a time. The system's ability to recognize these gestures indicates that we could develop interaction techniques that rely on sequences of gestures. It would also be interesting to compare such sequenced interaction with simultaneous performance of several gestures at a time. For example, how does recognition performance compare when doing an index finger pinch followed by a middle finger pinch, vs. a simultaneous index and middle finger pinch. Apart from recognition performance, users' perception and performance of these different styles of multi-finger interactions must also be considered carefully.

4.5.6 Conclusion

This chapter demonstrates that muscle-sensing can be used to accurately classify a useful variety of finger gestures, even when the hands are under load. It also shows that classification can be done in real-time, thus making forearm muscle-sensing viable for human-computer interaction. Furthermore, it highlights the tradeoff between speed and accuracy that results from providing users with immediate visual feedback. Finally, it introduces a novel bimanual technique for accurate engagement/disengagement of the recognizer, a crucial aspect of making muscle sensing usable for interactive tasks. In addition to the formal experimentation and results, we have demonstrated more holistic interaction via our portable music player application and a prototype game.

Chapter 5
Making Muscle-Computer Interfaces More Practical*

The previous two chapters demonstrated the potential of human-computer interfaces based on finger gestures sensed from electrodes on the upper forearm. While this approach holds much potential, thus far we have given little attention to sensing finger gestures in the context of three important real-world requirements: sensing hardware suitable for mobile and off-desktop environments, electrodes that can be put on quickly without adhesives or gel, and gesture recognition techniques that require no new training or calibration after re-donning a muscle-sensing armband. In this chapter, we describe our approach to overcoming these challenges, and we demonstrate average classification accuracies as high as 86% for pinching with one of three fingers in a two-session, eight-person experiment.

5.1 Wireless Platform

In this section, we describe our muscle-sensing hardware, gesture sets, and classification techniques.

* Parts of this chapter are adapted from (Saponas et al. 2010).

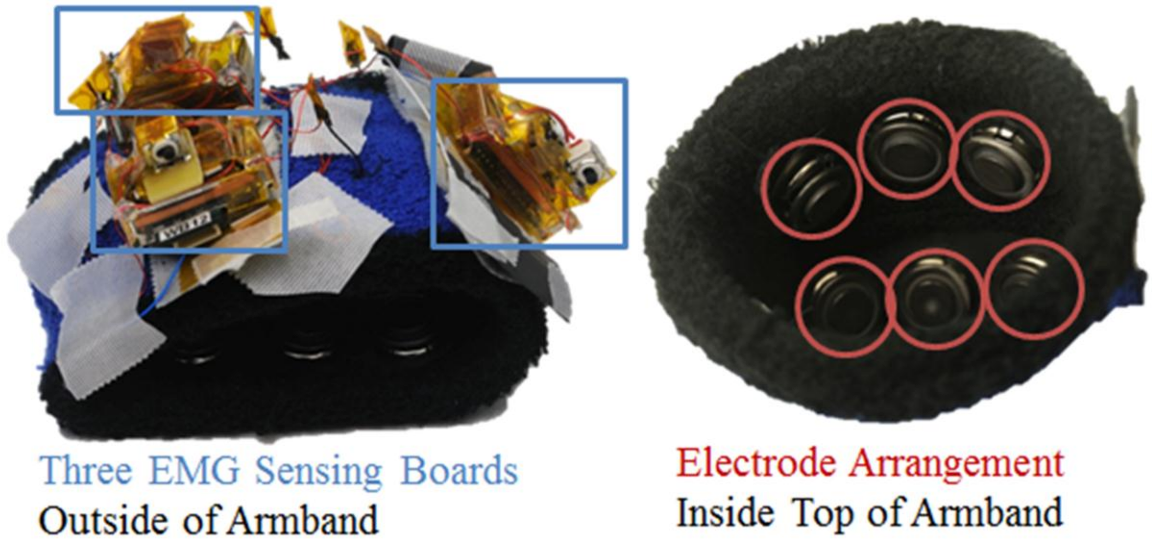

Figure 5-1: Our wireless-muscle sensing armband.

5.1.1 Wireless Muscle-Sensing Armband

Recall that our vision for muscle-computer interfaces is a thin armband worn on the upper forearm, capable of sensing a variety of finger gestures. To this end, we have created an embedded wireless muscle sensing device combined with electrodes and a sports sweatband (see Figure 5-1). Our embedded EMG sensing board has two analog muscle sensing channels and a Zigbee wireless radio (see Figure 5-2). During the development process, we discovered that six channels were more appropriate for robust gesture recognition. As a result, our right-hand armband carries three of our two-channel devices wired together (see Figure 5-1). Each sensing channel consists of one pair of silver-chloride-coated electrodes connected to a differential amplifier with very high input impedance (DC gain 20 dB). These are arranged into two sets of three channels; one set on top of the forearm the other set on the bottom. The armband streams raw data to a desktop computer where all processing takes place. In future iterations, some preprocessing might occur onboard and the rest of the processing could be carried out on a mobile device. Our board consumes approximately 35 mA constantly while sampling and streaming over the radio.

5.1.2 Gesture Sets

In this chapter, we explore two gesture sets for our muscle-computer interface that are distinct, yet related to the gesture sets described in Chapter 3 and Chapter 4. Our first gesture set consists of pinching one's index, middle, or ring finger together with their thumb. This can be performed

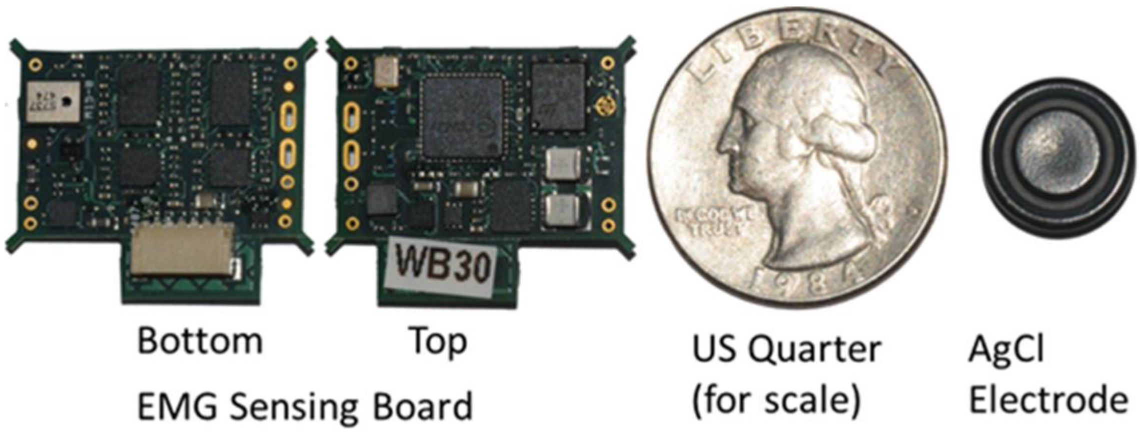

Figure 5-2: Our embedded wireless muscle sensing board and a silver chloride electrode pictured with a quarter for scale.

in a wide variety of situations and can even be done when already holding or carrying objects. In our second gesture set, users rest their hand on a surface and press down with one of their index, middle, or ring fingers. Pressing on a surface is a gesture that can be performed when someone is resting their hand on a table, the arm of a chair, or even on one's own leg. Both of these gesture sets leverage isometric muscle contraction; that is, during the gesture, a user's muscles continue to fire without finger movement.

Similar to the approach described in Chapter 4, we combine these gestures on the right hand with a squeezing action in the left hand to create a bimanual select-plus-activate compound gesture. This approach has three advantages. First, it gives the user an explicit method to indicate their intention to provide input. Second, a large squeezing action can be robustly detected with few false positives even during arm movement. Third, users can rapidly perform several of the same gesture in a row by pinching with their right hand while using their left hand to "pump" multiple successive squeezes.

5.1.3 Finger Gesture Training and Classification Technique

We follow a gesture classification scheme similar to that described in Chapter 4 with several enhancements. We continue to employ a machine learning approach using a support vector machine (SVM) that classifies gestures using 31 millisecond segments of EMG data. However, because of the limitations of sampling data with our microcontroller and transmitting them over Zigbee to a desktop computer, we only compute twelve classifications per second. Each classification result is obtained by first collecting 32 samples for each of our six sensing channels.

We then extract the following types of features from the window of data: the amplitude of each channel as the root mean square (RMS) amplitude of the fully rectified signal, the RMS ratios among all six channels, spectral power in several frequency bands, the ratio of high-frequency energy to low-frequency energy within each channel, and lastly the phase coherence among each pair of channels. All of these features are combined to create a 54-element feature vector over a 30-millisecond window. The values of the feature vector are normalized based on a four-second calibration step where users sequentially pinch each of their fingers. We use these feature vectors both for training the SVM as well as for classifying gestures.

Just as in previous chapters, we train our system by asking users to perform a sequence of actions while the system computes and records feature vectors from their muscle data. To aid in this process, the computer guides the user through the training phase by presenting a finger gesture to perform for four seconds, giving the user a one-second break, and then showing another gesture to perform. The system keeps the feature vectors computed over the second half of each gesture as “good” data for training.

After training an SVM, it can be used to classify subsequent gestures. In our system, we attempt to classify every feature vector, yielding approximately twelve classifications per second. While a user is performing a gesture, the system might correctly identify their gesture most of the time, but occasionally misclassify some segments of data. We attempt to be robust to these circumstances by adding a level of smoothing where the current recognized gesture at any time is computed as the majority vote of the previous six classifications. Unlike the smoothing approach in Chapter 4, here we add the logic that if the system observes three consecutive identical classifications, that result will override the vote. Due to the lower rate of classifications due to our wireless transmission bottleneck, we introduce this short circuiting to enable the system to respond more quickly.

As mentioned in the Gesture Sets subsection, our complete finger gestures consist of a bimanual select-and-activate action where a user, for example, might pinch with their right index finger and then squeeze their left hand. The user’s right-hand pinching gestures are continually classified as described above, but are only used as an input to the system at the moment they squeeze their left hand to *activate* the gesture. As another change from the method in Chapter 4, instead of just watching for the RMS to pass above a certain level, we instead infer left-hand activation using a gradient detector that watches for large changes in muscle signal amplitude. To reduce false

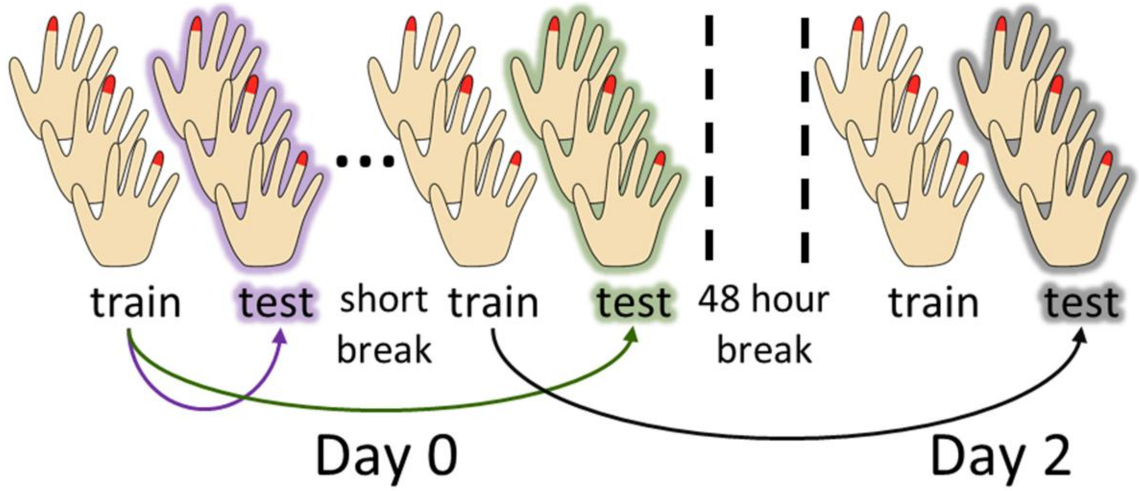

Figure 5-3: Experimental Setup: Eight participants engaged in three train/test sessions on two separate days.

positives due to motion artifacts caused by arm shifting and twisting, we further filter our gradient detector by requiring that activation be at least 35% of the maximum amplitude recorded during calibration as well as a ratio of at least two to one of low-frequency energy to high-frequency energy.

5.2 Cross-Session Classification Experiment

We evaluated the effectiveness of our wireless armband and classification approach in a multi-session experiment.

5.2.1 Participants

Eight participants (four female) from our research organization volunteered to participate in our experiment. They ranged in age from 23 to 31. Seven of our participants were right-handed and one participant was left-handed; however, each of our participants performed the pinching and pressing gestures with their right hand. None of our participants had any known neuromuscular diseases. We attempted to place the armbands in a similar position on every participant.

See Appendix C for the complete demographic information and the demographic questionnaire.

5.2.2 Design and Procedure

Each participant took part in two sessions occurring on two separate days. In the first session, participants first *trained* the system to recognize their finger gestures then *tested* the system's ability to recognize their gestures. Training entailed providing twelve examples for each finger

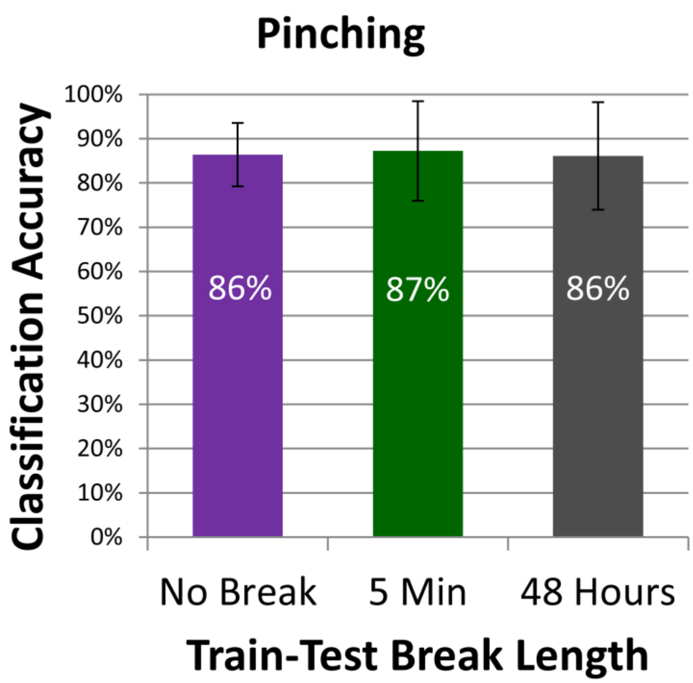

Figure 5-4: Summary of classification results for pinching index, middle, and ring fingers in air from sessions on two separate days. Error bars for the mean represent standard deviation.

gesture in both gesture sets. Testing included fifteen attempts of each gesture. Participants did this for both gesture sets, with the order counter-balanced across participants.

Following the first training and testing segment, we removed the armbands from participants' arms and gave them a short break to get up and walk around the room. After their break, we had them repeat the process of providing the system training data and then testing the recognition ability of the system. As illustrated in Figure 5-3, both testing phases during their first day used the training data they initially provided after putting on the muscle-sensing armbands. This tested the ability of the system to recognize gestures when the training and test data corresponded to identical electrode placement and when the armband was removed and re-donned between training and testing, leading to slightly different electrode placements. Approximately 48 hours later, participants came back for a second session and again engaged in a training and testing session. During testing on the second day, the system used the training data provided during the second part of the first day's session, challenging the system to classify their gestures with no new training or calibration from the placement of the armband on the second day.

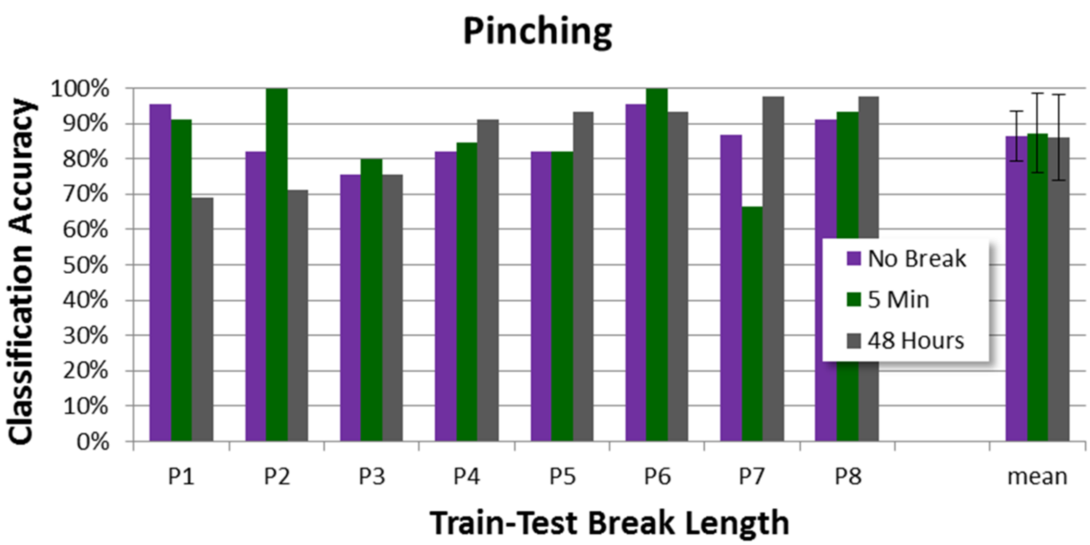

Figure 5-5: Per participant classification results for pinching index, middle, and ring fingers in air from sessions on two separate days. Error bars for the mean represent standard deviation.

5.2.3 Results

Our metric for performance is simply the accuracy with which our system can classify people's finger gestures. From our experiment, we can evaluate this ability based on three scenarios: when the system is trained immediately before use, when the system was trained earlier that day but the armband has been removed and reapplied, and lastly when no new training or calibration is provided that day.

Our system performed best at recognizing the pinching gesture, with mean accuracies for same session, short break, and one-day break of 86%, 87%, and 86%, respectively; chance was 33% (see Figure 5-4). Mean accuracies for each participant's pinching gesture can be seen in Figure 5-5. We found that our system classified pressing gestures slightly less well, with a mean accuracy of 76% when collecting training data immediately prior to testing, 73% after a short break, and 66% when using training from a previous day (see Figure 5-6). Results for each participants pressing gesture can be seen in Figure 5-7.

5.2.4 Discussion

Reflecting on the less robust classification of pressing-on-a-surface gestures, we think the primary source of misclassification is the variety of methods people can employ when pressing a single finger down on a surface. For example, they can relax their other fingers or they can pull them away from the surface. Similarly, they can use muscles in their hand, forearm, or shoulder to exert force. However, our system did perform well for two of our participants, suggesting that it may be possible with practice or feedback to classify these gestures.

The ability of our system to classify finger pinching gestures using only training data collected on a different day is an encouraging result toward the vision of a muscle-computer interface armband that can be quickly slipped on before starting a task in the lab or heading out the door for a run. In fact, for five of our eight participants, the system recognized their day-two pinching at an accuracy of 95%.

We believe there are two main reasons why our system performed much better for a little over half of our participants. First, as with any new input device or tool, people have different initial intuitions and abilities. In the case of our muscle-computer input technique, the more consistently a user gestures, the better the system will perform. We believe that over time, system performance would improve for any given user as they develop a more repeatable form of their gestures. Such changes over time might also warrant a method for periodically updating the

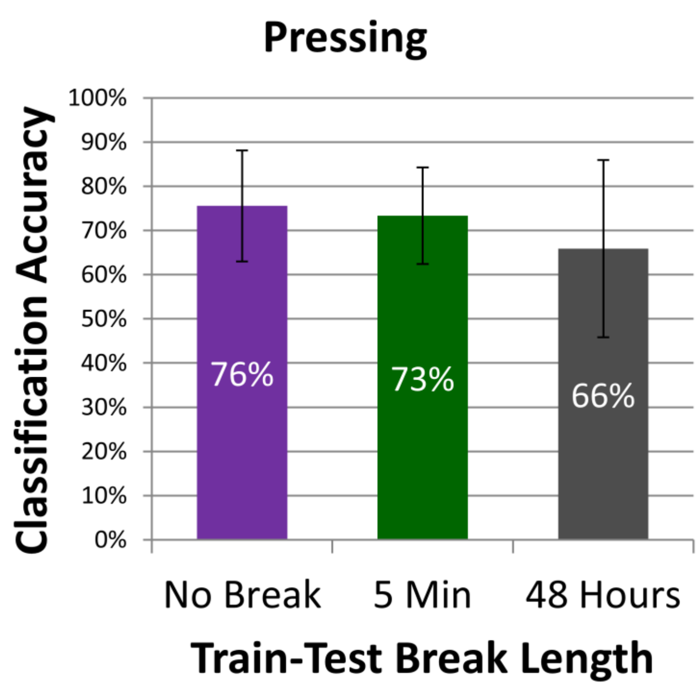

Figure 5-6: Summary of classification results for pressing index, middle, and ring fingers on a surface from sessions on two separate days. Error bars for the mean represent standard deviation.

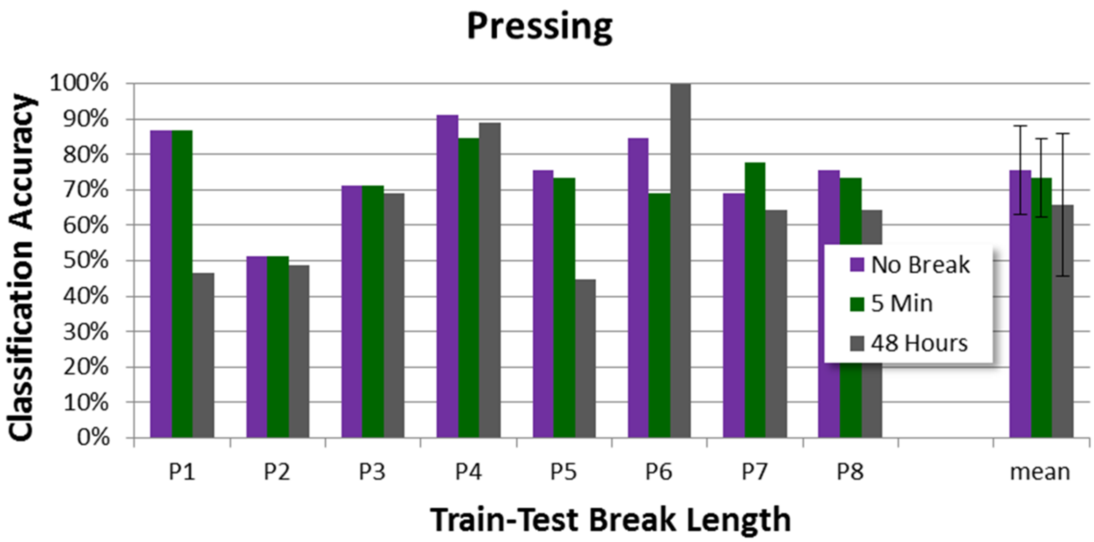

Figure 5-7: Per participant classification results for pressing index, middle, and ring fingers on a surface from sessions on two separate days. Error bars for the mean represent standard deviation.

system with more recent training. A second source of variance among users is the matching of their physiology to the layout of electrodes on the armband and our placement of the armband on their arm. While we did not observe an obvious systematic effect of body type in our experiment, it is a variable worthy of further investigation. We also think it may be possible to further improve on our results by reducing the impact of armband orientation by creating an armband with a dense array of electrodes where the system detects which subset of electrodes will be most effective.

5.3 Summary

In this chapter, we have presented a system for accurately classifying pinch gestures with no new training or calibration after re-donning a wireless armband 48 hours after initially training the system. Our work is a step towards making muscle-computer interfaces a more practical possibility for controlling computing devices in mobile, off-the-desktop situations where traditional input devices are inconvenient or impossible to use.

Chapter 6
Applications of Muscle-Computer Interfaces*

In this chapter, we introduce two applications of the muscle-computer interface techniques described in the previous three chapters. The first application is in the gaming space, employing bimanual pinching gestures to control the Guitar Hero video game. The second application of muscle-computer interfaces is augmenting the Microsoft Surface interactive tabletop with new interactions enabled by muscle sensing.

6.1 Air Guitar Hero

Guitar Hero is a video game for both consoles and PCs that engages players in a simplified guitar playing experience (see Figure 6-1). In the game, abstract musical notes from a song scroll toward the player on the screen. Each note corresponds to a key or a set of keys on a plastic guitar controller. The goal of the game is to hold down the correct keys and strum the guitar controller when each note passes the bottom of the screen. Some of the song audio plays constantly while other parts (such as the lead guitar part) are played when and if the user plays the correct note at the correct time. A player's score is loosely determined by how many notes they play correctly

* Parts of this chapter are adapted from (Benko et al. 2009).

Figure 6-1: Screenshot from the Guitar Hero video game.

including bonuses for getting many notes in a row correctly. The game can be played by one to four players. When multiple players are playing, they each play separate but complementary components of the song (e.g., lead guitar and bass guitar). As a demonstration of how quickly and accurately our muscle-computer interface systems can work, we built a muscle-computer interface to Guitar Hero that enables an “air guitar” experience. Instead of using a plastic guitar controller where the keys correspond to notes and strumming is done by flicking a momentary contact switch, we instead provide the same input through a muscle-sensing interface. The remainder of this subsection describes our implementation.

6.1.1 Mapping Finger Level Gesture Classification to Guitar Hero Controls

In our Air Guitar Hero input technique for the Guitar Hero video game, we map the notes (or frets) to pinching a player’s thumb with one of their index, middle, or ring fingers (only three notes are required in the “easy” mode of the game) and strumming to making a strumming motion with the players other hand. For a right handed person, this means having six muscle sensors on their left arm sensing pinching gestures and two sensors on their right arm sensing strumming. As an example, a player pinching with their left index finger corresponds to the left most button (the green button) being held down in the game. The player making an abrupt strumming motion with their left hand and arm corresponds to the strum button being tapped. Thus, when playing the

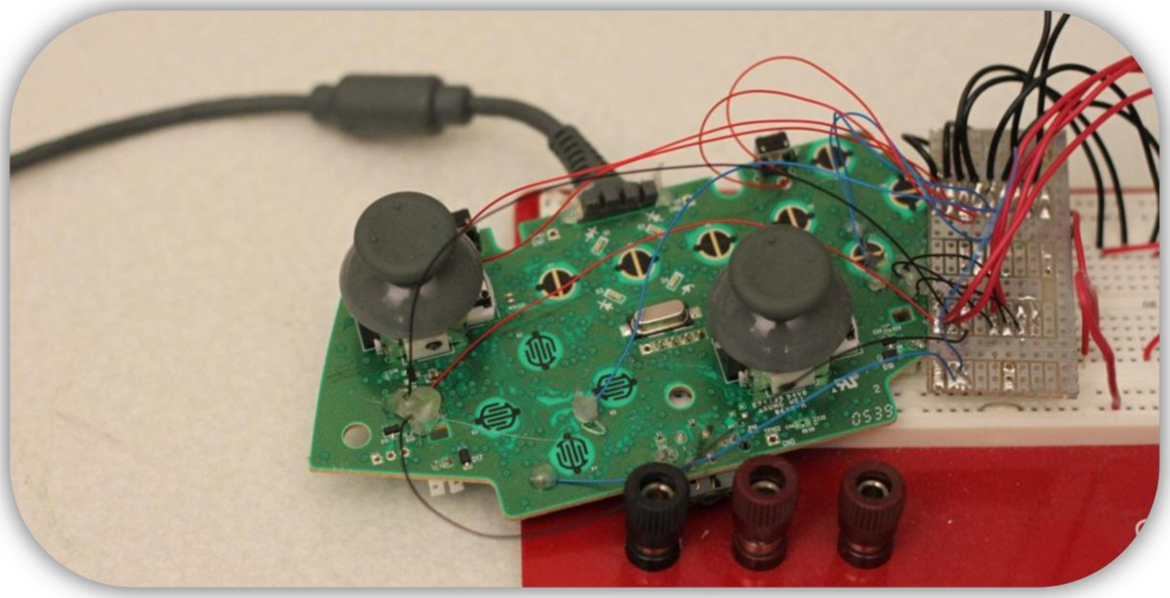

Figure 6-2: Modified XBOX controller for interfacing with the Guitar Hero III video game.

game, a player is attempting to pinch with the finger of their left hand corresponding to the next note to be played and then strum with their right hand at the time that note is meant to be played (when that note reaches the bottom of the screen).

We initially used the XBOX version of Guitar Hero III to build our system. To control the Guitar Hero game on an XBOX we modified a standard wired XBOX controller so that we could control when the buttons were being pressed down. To do this, we soldered wires to the test points corresponding to each side of the buttons on the XBOX PCBs and connected each button to a reed relay (see Figure 6-2). Connections to the test point were held in place by hot glue. The reed relay allowed us to "manually" complete the electrical connection for the button that would normally connect when pressed down and do so without interfering with the signal carried on that connection.

We later switched to the newer Guitar Hero 5 version of the game for XBOX and also to controlling the game by modifying an XBOX guitar controller for Guitar Hero. We followed a similar method for controlling this hardware device by first disassembling the neck of the controller, then disconnecting the neck's controller board from the buttons in the neck, and finally directly interfaced with that connection by connecting a reed relay between each button signal

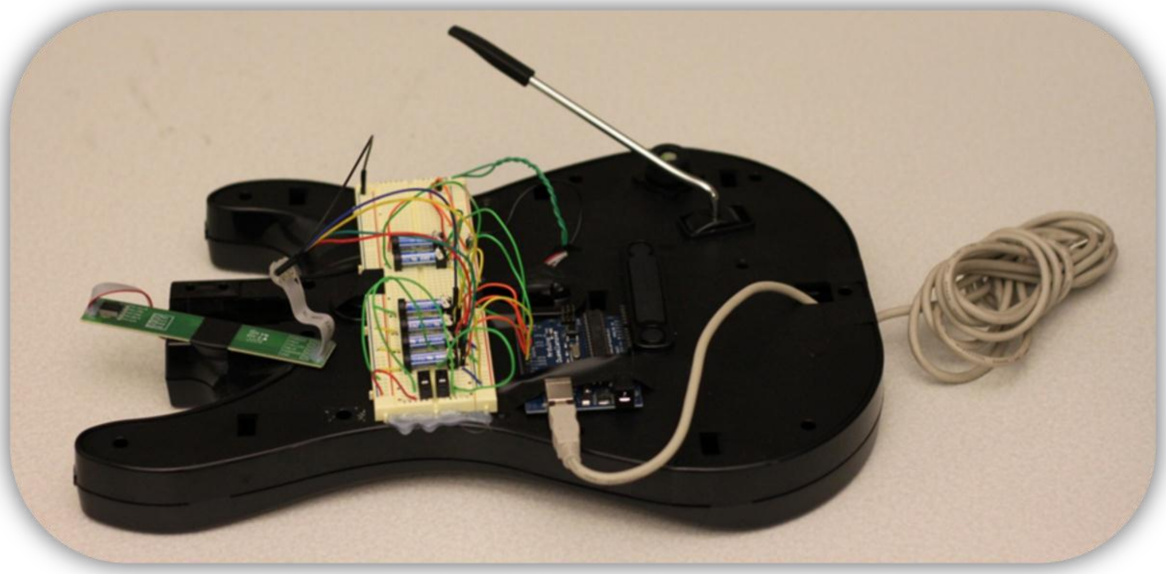

Figure 6-3: A Guitar Hero controller we modified to give input to the Guitar Hero 5 video game.

line and ground on the neck controller board (see Figure 6-3). We controlled the reed relays using an Arduino connected to a PC by a virtual serial port over USB. We also connected another reed relay to the connection point for the strumming momentary contact switch inside the body of the guitar controller. To control either of the controllers we modified, we just apply a positive voltage across the reed relay, to close the connection that is normally closed by pressing the buttons on the controller.

Wired Implementation

We first implemented our Air Guitar Hero interface using our wired EMG setup from Chapter 4. In this implementation, our physical setup consisted of a player sitting in a chair wearing an outer shirt over their normal clothing that housed the wires for the sensors such that they ran up their arm toward their shoulder and out the back of their shirt (see Figure 6-4). On their left forearm we had six electrodes corresponding to six EMG sensing channels and an additional two electrodes for the active grounding component of the data acquisition box. These six sensing channels were used for sensing pinching gestures with their left hand. On their right forearm, we had two electrodes corresponding to two sensing channels. These two sensing channels were used for detecting a strumming motion for controlling strumming in the game.

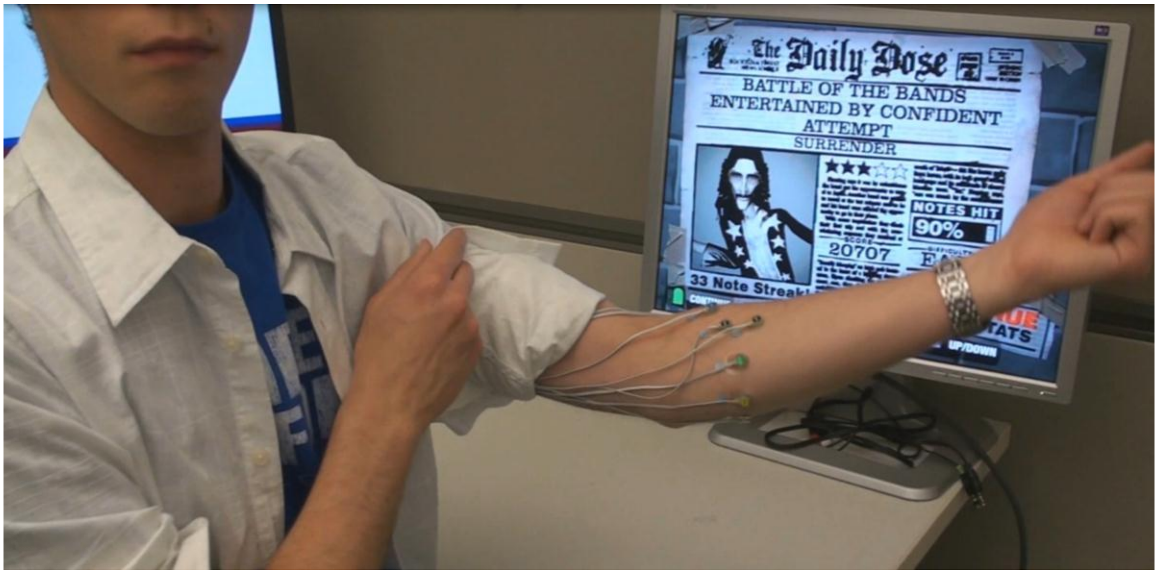

Figure 6-4: Wired implementation of our Air Guitar Hero interface. Wired sensors are run up the player's sleeve and out the back of their shirt. Players sit in a chair while playing. Video can be seen at: http://www.youtube.com/watch?v=X1r2TYvGpHo

In the wired implementation, we used a one-step smoothing process on top of the output from our SVM based statistical machine learning approach described in Chapter 4. Our smoothing approach was to set the current inferred note being held down to whichever finger was last inferred as being used for pinching three times in a row in the 30 classifications per second generated by our SVM. For example, the consecutive classification outputs of "index, middle, middle, middle, ring, ring, index," the smoothing step would declare "middle" or "the red note" as the current note as soon as the third middle in a row was inferred.

The strumming motion was detected using the same scheme as the activation detection described in Chapter 4. After calibration, the system watched for the activation level of the right arm to go above 50 percent of the maximum value observed during calibration. The activation level is calculated as the RMS value over the fully rectified signal resulting from subtracting the two EMG channels on the right arm. After each strum detection, there is a sleep period of 100 milliseconds to prevent too rapid of detection of the strumming motion.

Wireless Implementation

We later implemented a new version of our Air Guitar Hero interface using our new wireless muscle-sensing platform described in Chapter 5. In this iteration, players wear two armbands: one

Figure 6-5: Wireless implementation of our Air Guitar Hero interface. Players can stand and move around while wearing two wireless EMG armbands. Video can be seen at: http://www.youtube.com/watch?v=pktVSTwC8qo

wireless armband with six sensing channels on their left forearm and one wireless armband with two sensing channels on their right forearm. Pinch detection was also used in this implementation for playing notes in Guitar Hero. The system from Chapter 5 was used for inferring pinching. This included the smoothing approach of majority vote over the most recent six SVM recognition results with a short circuiting if two of the same inference results in a row are seen. Strumming detection was also done using the enhanced activation classifier of a gradient detector, with a minimum threshold, and a constraint regarding the ratio of energy in high frequency bands to low frequency bands. This new implementation enabled a more enjoyable "air guitar" experience because the player could be standing and moving around as they played (see Figure 6-5).

6.1.2 Discussion

Our Air Guitar Hero application of muscle-computer interfaces demonstrates that our system classifies quickly and accurately enough for the time-sensitive task of playing the Guitar Hero video game. This is challenging for several reasons. First, playing a videogame has a significant cognitive component as well as a time pressure component that together serve as a large distraction from concentrating on performing pinching gestures in a consistent and careful manner. This is a change from our laboratory experiments where participants primarily focus all of their attention on performing the pinching gestures. Second, the nature of the game requires the player to anticipate and perform pinching gestures at specific times as governed by the game. This is in contrast to our laboratory setup, where experiment participants would see a visual stimulus and then choose to perform gestures at their own pace. In the Guitar Hero context, they could be in a rush to perform a gesture, anticipate the next gesture, and then transition into a new gesture corresponding to the next note in the sequence presented by the game. Lastly, in our wireless implementation, players can move around and really “get into the game” by moving around to the music while enjoying their “air guitar” experience.

Was this application successful?

The Air Guitar Hero application using our approach to muscle-computer interfaces was successful in both demonstrating that our system is fast and accurate enough for gaming applications as well as showing that finger level gestures performed as a part of tasks with a cognitive component can be recognized accurately. We think this application is just the tip of the iceberg of potential interactive applications in both gaming and mobile computing scenarios. Chapter 7 discusses some of these potential new applications.

Why this interface works

One of the reasons our muscle-computer interface for Guitar Hero works well is that the latency requirements of our approach line up well with that of the latency requirements of the game. The most time sensitive component of Guitar Hero is strumming at the correct time. In our strum detector, we only need to examine the most recent 31ms window of time to catch the abrupt change in muscle signal caused by a strumming motion. Thus, our system’s quick response to the physical motion of strumming allows players to make their strumming motion at the same time as they would if they were using a tradition plastic guitar controller. Our mechanism for reliably detecting which finger a player is pinching requires more time because it applies smoothing over the last several classification results. However, in Guitar Hero, players typically begin pressing the button corresponding to the next note to be played in preparation before they strum. This common behavior fits well with the latency introduced by our smoothing system. Another advantageous aspect of our approach is that a player’s right and left arm are separate muscle systems and the actions being taken by one arm largely do not affect the sensing on the other arm.

A common component to a song in Guitar Hero is playing the same note several times in a row in fast succession. Another reason our approach is effective is that this is a type of input that is fairly easy to execute using our muscle-computer interface; players simply hold one pinch in their left hand and strum to the beat with their right hand.

Another property of our system that fits particularly well with Guitar Hero is that our method of collecting training data somewhat resembles playing the game, just very slowly. We collect training data by providing a visual stimulus to a user indicating which finger to pinch. After a user begins pinching with that finger they squeeze their contralateral hand. When they squeeze their contralateral hand the system then uses the signal recorded just prior to squeezing as training data for the system. This setup mirrors pinching one's finger to select a note and strumming with one's contralateral hand to play the note.

6.2 Augmenting Interactive Surfaces with Muscle-Computer Input

Interactive surfaces extend traditional desktop computing by allowing direct manipulation of objects in a way that draws on our experiences with the physical world. However, the limited scope of information provided by current tabletop interfaces falls significantly short of the rich gestural capabilities of the human hand. Most systems are unable to differentiate properties such as which finger or person is touching the surface, the amount of pressure exerted, or gestures that occur when not in contact with the surface. These limitations constrain the design space and interaction bandwidth of tabletop systems.

In this section, we explore the feasibility of expanding the interaction possibilities on interactive surfaces by sensing muscle activity via *forearm electromyography* (EMG). EMG allows us to infer additional information about each touch contact with an interactive surface, and provides novel information about hand and finger movement away from the surface. We employ muscle sensing in combination with the contact sensing of a standard multi-touch tabletop (the Microsoft Surface) and introduce novel interactions that emerge from this combination of sensor streams.

As demonstrated in previous sensor fusion work, the combination of multiple complementary streams can often be greater than the sum of the parts (Harada, Saponas, and Landay 2007; Oviatt et al. 2000). For example, in our work, we use muscle sensing to determine which finger is in contact with a surface, assess the level of pressure exerted by the user while they are pressing down, and even detect activity when a user's hand is not in contact with the surface. Combining these sensing modalities allows us to explore finger-specific input, pressure-sensitive interaction, and free-space gestures that complement traditional on-surface interactions.

6.2.1 Interactive Surface Sensing

While most available multi-touch systems are capable of tracking various points of user contact with a surface (Dietz and Leigh 2001), the problem of identifying particular fingers, hands, or hand postures is less well solved. Existing approaches to solving this problem include camera-based sensing, electro-static coupling, and instrumented gloves.

Several camera-based interactive surface systems have demonstrated the capability to image the user's hands, either above the display (Wilson 2005) or through the display (Benko and Wilson 2009; Wilson 2004), but none of these explore contact identification or freehand interactions in the space above the surface. Malik et al. (Malik, Ranjan, and Balakrishnan 2005) used two overhead cameras to detect hand postures as well as which finger of which hand touched a surface, but required a black background for reliable recognition. In general, camera-based approaches have two shortcomings: fingers and hands can easily be occluded and contact pressure is not robustly observable.

Techniques such as frustrated total internal reflection (FTIR) (Han 2005) are able to estimate contact pressure by detecting changes in the shape of a contact that are often indicative of pressure changes. However, this approach has limited precision. FTIR systems cannot reliably discriminate contact shape changes due to posture adjustments from those due to pressure variation. FTIR systems also cannot reliably identify contacts as belonging to particular fingers.

Benko et al. (Benko and Feiner 2007) demonstrated a multi-finger interaction technique which required users to wear instrumented gloves for finger identification. Gloves have also been extensively used in virtual reality research. For example, Cutler et al. (Cutler, Frölich, and Hanrahan 1997) used gloves for above-the-surface 3D interactions. While simple and reliable, gloves suffer from many issues, including hygiene, comfort, access time, and a reduction in the directness offered by "direct touch" interfaces.

Interaction in the space *above* the interactive surface has also been explored with styli (Kattinakere, Grossman, and Subramanian 2007), video cameras (Wilson 2004; Wilson 2005; Wilson 2006), and depth-sensing cameras (Wilson 2007; Benko and Wilson 2009). The use of depth-sensing cameras is particularly of interest, as it facilitates precise 3D hand positioning and gesture-tracking without requiring the user to wear on-body sensors. However, low sensing resolution, finger visibility, and occlusion issues make such approaches potentially more errorprone than the approach described in this paper. In addition, neither depth-sensing or standard video cameras are able to directly sense contact pressure and require gestures to be “within sight” of the surface. Other technologies such as Izadi et al.’s SecondLight (Izadi et al. 2008) permit projection onto objects held in the space above the surface. While supporting an interesting set of interactions, this does not allow input away from the surface, only output.

While not in the domain of surface computing, Sugiura and Koseki (Sugiura and Koseki 1998) demonstrated the concept of finger-dependent user interface elements and interactions. They relied on a standalone fingerprint reader to determine which finger was used and assigned data and specific properties to each of the user’s fingers.

6.2.2 Combining Muscle and Touch Sensing

Touch-sensitive surfaces and muscle sensing provide complementary streams of information. Touch-sensitive surfaces provide precise location and tracking information when a user’s hand is in contact with the surface. They can also precisely record temporal information about the arrival and removal of contacts. Our approach to muscle-computer interfaces can detect which fingers, are engaged in an interaction. It can also approximate the level of activation of muscle groups, which allows the estimation of contact pressure. Furthermore, muscle-sensing can provide information about an interaction even when a user’s hand is no longer in contact with a surface. However, muscle-sensing does not provide spatial information, and is not as reliable as touch-sensing for temporally-sensitive gestures. We thus introduce a multimodal system that relies on surface input for spatial information and muscle sensing for finger identification, pressure, and off-surface gestures.

Equipment and Setup

Our system is implemented using a Microsoft Surface (Microsoft c) and a BioSemi Active Two EMG device (Biosemi).

As in Chapter 4, we place six sensors and two ground electrodes in a roughly uniform ring around the upper forearm of the user’s dominant hand for sensing finger gestures (see Figure 6-6). We also placed two sensors on the forearm of the non-dominant hand for recognizing coarse muscle activation.

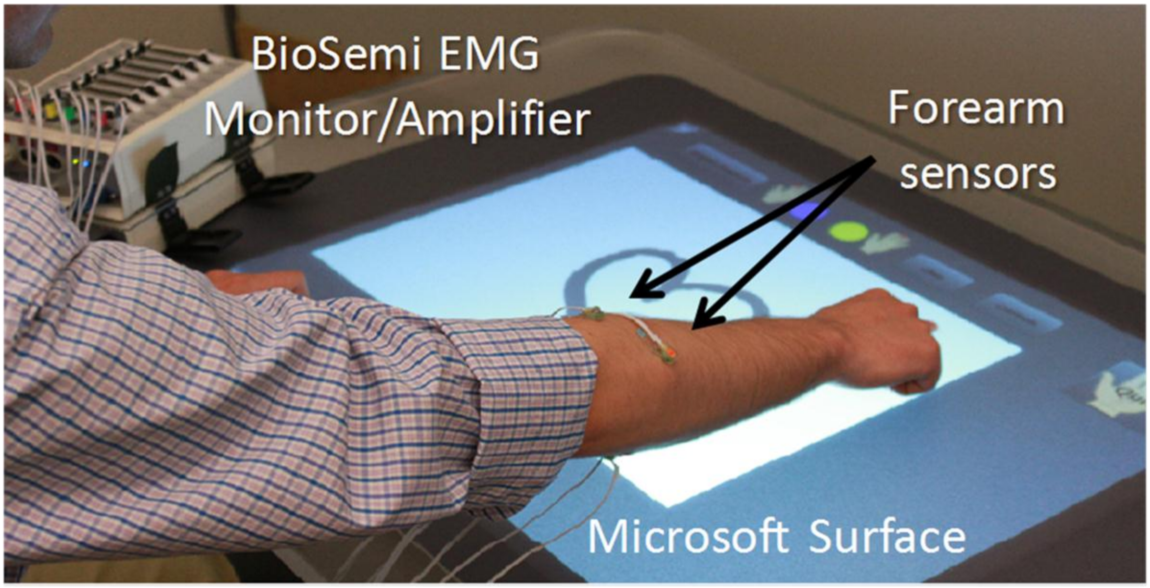

Figure 6-6: Our system employs electromyography (electrodes placed on the upper forearm) to infer finger identity, estimate finger pressure, and enable off-surface gestures. For a video overview see http://www.youtube.com/watch?v=0phjl804onU

Sensing Primitives

Our system employs muscle-sensing to provide four primitives to applications on the interactive surface. Calibrating and training our system for all four primitives requires approximately five minutes per user. We discuss incorporating these primitives into hybrid interaction techniques in the next subsection. The remainder of this subsection describes each of our four interaction primitives.

Level of Pressure

The *pressure* primitive is a smoothed, down-sampled representation of the raw level of muscle activation on the dominant hand. This feature requires no training, but only a ten-second calibration procedure that allows the system to scale pressure values appropriately. Pressure is calculated on a linear scale based on the RMS of each 1/30 ms window averaged across the six sensing channels on the dominant arm. The pressure values are also smoothed by calculating the current pressure as an average over the last several windows of time. The latency of pressure reporting is approximately 150ms. Approximately 20ms of this latency (and that of the other primitives) is driven primarily by our prototype setup: muscle-sensing is processed on a separate computer and then is sent over a network connection to the Microsoft Surface computer where it is correlated with touch contact information.

Contact Finger Identification

This primitive is based on the machine learning approach to finger level gesture identification described in Chapter 5. Specifically, we again train a support vector machine (SVM) based on features of frequency and amplitude information in the EMG signal to determine which finger is applying pressure to the surface. The primary difference between the work presented in previous chapters and this use of muscle-sensing is the training and use of finger gestures that include dragging a single finger on a horizontal surface (previous chapters have explored pressing and taping on a horizontal surface, pinching in free space, squeezing objects, and pulling up on handles). This primitive requires about two minutes of training for each user. In previous chapters, users have been asked to respond to various visual stimuli while the arm is in a fixed position to collect labeled data. In our training, we instead prompt users to use each of their fingers to draw freely on the surface. At the end of the training period, the system analyzes the training data to build a real-time classifier. Building the classifier requires less than five seconds. The latency of finger identification is approximately 300ms.

"Pinch" and "Throw" Gestures

A "pinch" gesture consists of bringing a finger rapidly against the thumb, and lifting away from the surface, the way one might pick up a small object from a table. The "throw" gesture consists of rapidly opening the fingers from the pinched state, as one might do when throwing an object held between pinched fingers. The "pinch" and "throw" gestures are detected by looking for characteristic changes in the muscle activation level of the dominant hand. *Detecting* these gestures requires no training, but *identifying* the fingers performing these gestures currently requires a two-minute training procedure identical to that described for contact finger identification, except that instead of drawing on a surface, the system asks the user to pinch specific fingers against his or her thumb in mid-air for five seconds at a time during a two minute training period (this is the same as the procedures described in Chapter 4 and Chapter 5). The latency of pinch detection and identification is also approximately 300ms.

"Flick" Gesture

The "flick" gesture consists of a simple wave of the hand. The "flick" gesture is detected by looking for characteristic changes in the muscle activation level of the hand. This primitive requires no training other than a ten-second calibration procedure that allows the system to scale amplitude values appropriately. The latency of "flick" detection is approximately 50ms.

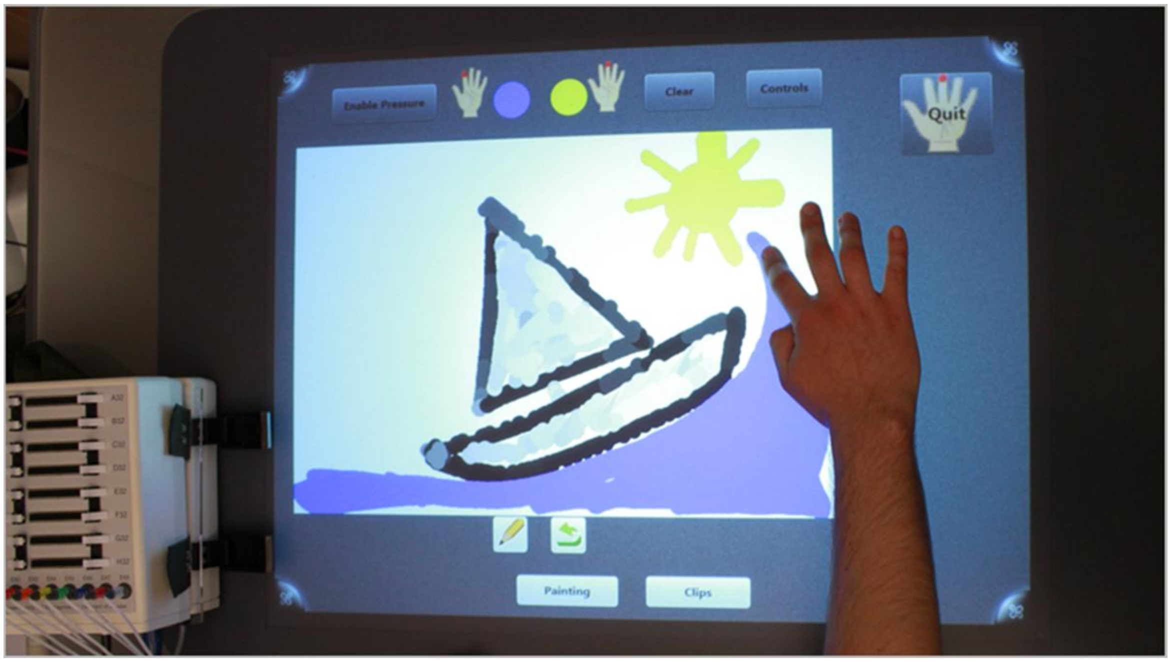

Figure 6-7: An example drawing demonstrates both pressure-painting and finger-dependent painting. A different color is mapped to each finger, and pressure controls stroke saturation.

Due to the equipment constraint of having only 8 EMG sensor channels and the resulting asymmetric setup, we bound each of the gestures to a specific hand. The dominant hand, with the larger number of sensors, could sense pressure, contact-finger identification, as well as the pinch and throw gestures. The flick gesture was restricted to the non-dominant hand.

Input Techniques

We have prototyped four interaction techniques to demonstrate and evaluate the utility of muscle sensing for interactive surfaces. These interactions are all prototyped within a simple painting and image-manipulation application.

Pressure-Sensitive Painting

To demonstrate our system's ability to estimate contact pressure, we associate different saturation levels in our painting application with different levels of finger pressure (more pressure results in darker strokes) (see Figure 6-7).

Finger-Aware Painting

To demonstrate our system's ability to associate surface contacts with specific fingers, we associate different brush colors with the index and middle fingers (see Figure 6-7). When the interactive surface detects a contact, it immediately queries the muscle-sensing system for the

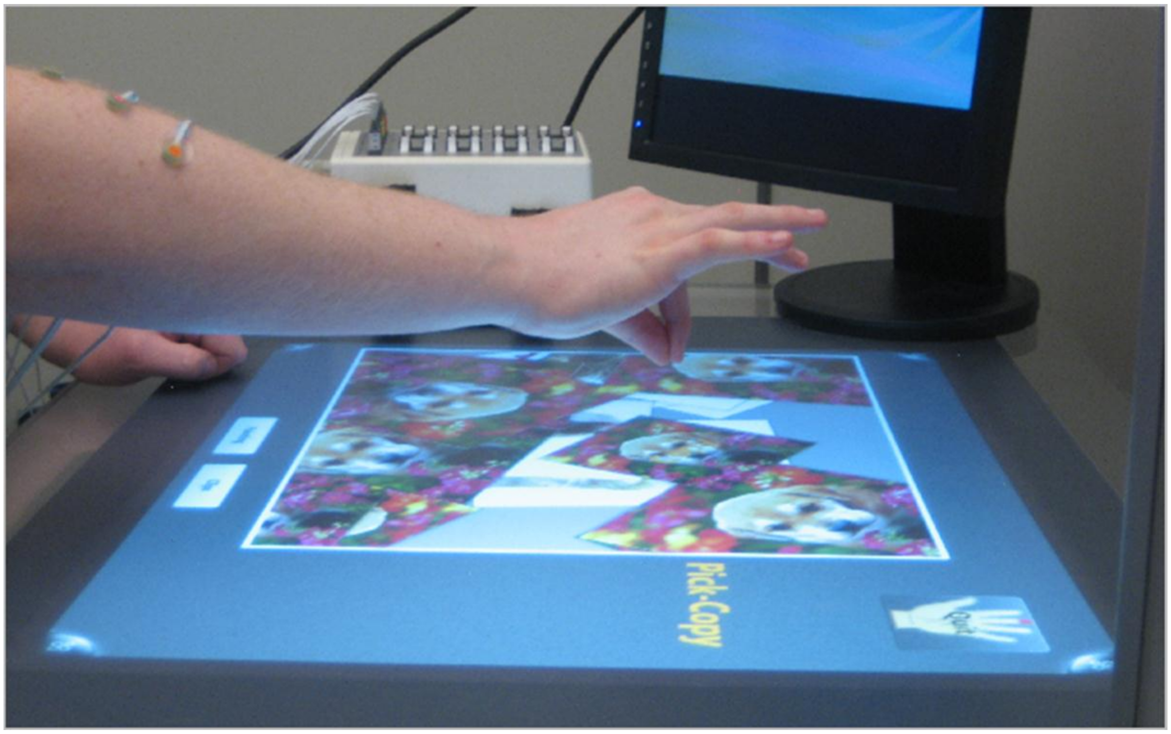

Figure 6-8: Performing the finger-dependent pick and throw interaction: A user picks up a virtual object by pinching it on the surface and lifting his hand away from the surface. Releasing the pinch returns the object to the current canvas.

identity of the active finger, and uses that color for the brush stroke associated with this contact. Because we have independent processing streams for touch and muscle sensing, we begin to draw a translucent stroke to maintain the sensation of responsiveness, and only fill in the color when the muscle-sensing system has returned with the finger it detects.

Finger-Dependent Pick and Throw

To demonstrate our system's ability to detect gestures more complex than simple touches, and to persist the state of those gestures even when the hand leaves the surface, we map the "pinch" and "throw" gesture primitives to "cut/copy" and "paste" operations on a simple photo canvas. Thus, the user is able to *pick* a photo up from the table and *throw* it back onto the canvas. Picking is initiated on the surface, by placing two fingers on the desired photo and then performing a pinch gesture (see Figure 6-8). By pinching with the index or middle finger, the user can specify whether to initiate a cut or a copy operation, respectively. A user holds on to a copied or cut photo by maintaining the pinch posture, even after the hand has left the surface, and pastes the object back onto the surface by executing the throw gesture. The user can perform arbitrary actions (e.g., switch between canvases) while she is holding the object and has not thrown it back.

Undo Flick

To demonstrate our system's ability to facilitate bimanual, off-the-surface interaction, we map the "flick" gesture performed by the non-dominant hand to the undo operation in our painting application. This action removes the most-recently-created stroke.

6.2.3 Experiment

To gather initial feedback on our system we brought participants into our lab to use several components of our system. The goals of our evaluation were to validate the basic feasibility of our system and interaction techniques, to assess their robustness and reliability, and to gather anecdotal responses from novice users about our proposed interaction techniques.

Participants

We recruited six participants (three female) from within our organization. Each participant spent approximately 90 minutes interacting with our system and was provided with a compensation valued at $10 for their time. See Appendix D for the demographic questionnaire.

Tasks

At the beginning of each participant's experimental session, we applied EMG sensors to the participant's arm as described in the previous section. We then asked the participant to make a tight fist and then relax, allowing calibration of the signal level for each hand. Introduction and the initial setup took approximately 15 minutes. Participants then completed the following five tasks (in order):

Task 1

Copy an image from a given paper template (see Figure 6-9a) using the pressure-sensitive painting technique. The image was presented on paper and contained varying levels of light and dark strokes.

Task 2

Copy an image from a given paper template (see Figure 6-9b) using the finger-aware painting technique. The image was presented on paper and contained blue and green strokes, which were mapped to the participant's index and middle fingers, respectively.

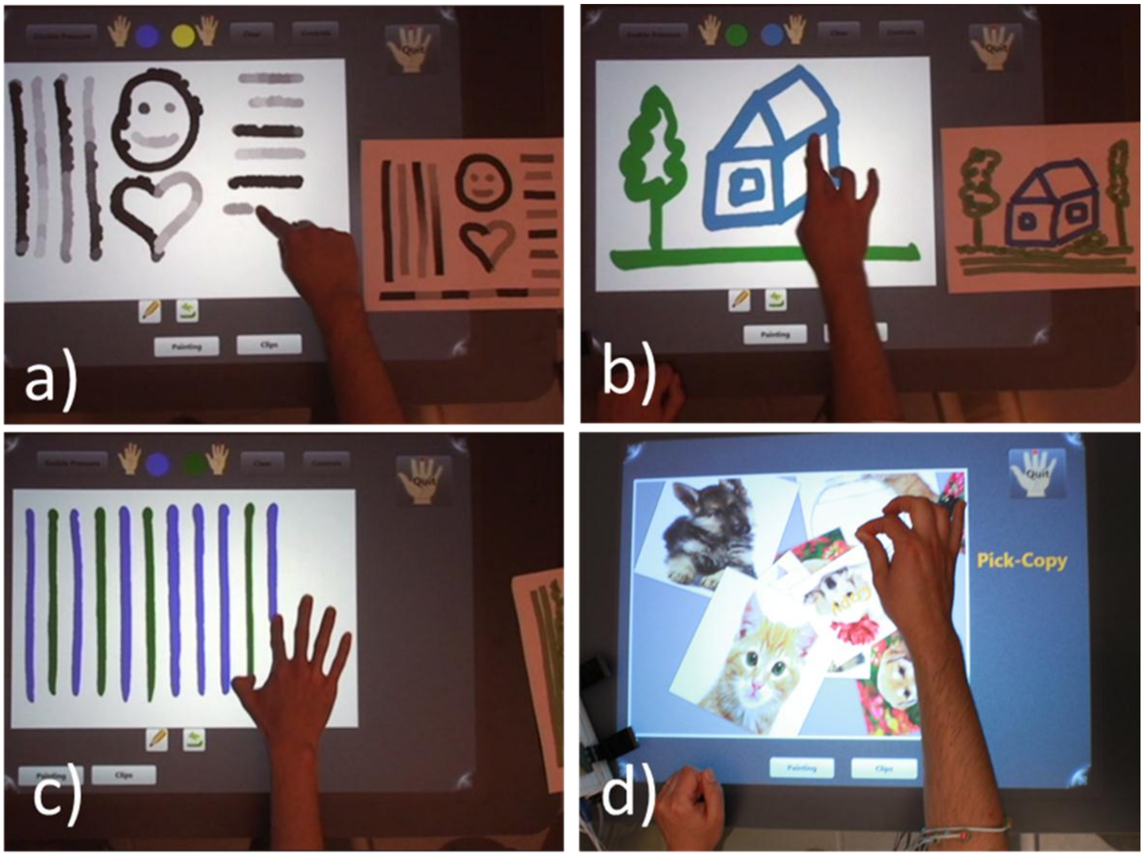

Figure 6-9: Four tasks from our user evaluation: (a) Task 1: copy an image using contact pressure to control saturation; (b) Task 2: copy an image using index and middle fingers to paint two separate colors; (c) Task 3: draw lines with alternating colors; and (d) Task 5: move three images and copy three images to a different canvas.

Task 3

Make a series of vertical lines across the surface, changing color with each vertical line (see Figure 6-9c). Each participant filled two canvases with vertical lines.

Task 4

Write the numbers from 1 to 10 on the surface, executing the “undo flick” gesture after each even number, but not after odd numbers. Correct execution of this task would leave only the odd numbers written on the surface. If an even number contained multiple strokes, participants executed the “undo flick” gesture as many times as was necessary to erase the number completely.

Task 5

Presented with a pile of six images on a canvas, either copy or move each image to another canvas, depending on the image category. Specifically, they had to copy images of cats and move images of dogs (see Figure 6-9d). Participants picked up images using our “pick” gesture, where the index finger initiated a “move/cut” operation and the middle finger initiated a “copy” operation. While the image was held in their dominant hand, participants pressed an on-screen button with their non-dominant hand to switch to the target canvas, and used the “throw” gesture to place the image on that canvas.

Additional Procedures

There were two additional training sessions: First, before performing Task 2, participants spent two minutes training the system to recognize finger-specific contacts. Second, participants spent another two minutes training the finger-specific “pinch” gesture before Task 5. Training in both cases consisted of repeated activation of a desired hand pose or gesture using a stimulus-response training method, i.e., the user was prompted with a particular pose/gesture on the screen, they performed it for 2 seconds, and then they relaxed their hand muscles.

Before performing each task, participants were given time to practice each interaction and ask questions. This practice session took no longer than 5 minutes. When comfortable with the interaction, participants proceeded to complete the specific tasks, which were untimed. On average, participants completed each task within one minute.

Figure 6-10: Pictures painted by participants in our experiment, where rows 1, 2, and 3 show the results of Tasks 1, 2, and 3 respectively. Task 1: copy the leftmost image using pressure-sensitive painting. Task 2: copy the leftmost image using image using index and middle fingers to paint in blue and green, respectively. Task 3: draw alternating blue and green lines using index and middle fingers, similar to task 2. The leftmost target images were provided to our participants on paper.

At the conclusion of the session, each participant completed a questionnaire that solicited feedback about each interaction.

Results

In this subsection, we present quantitative results from each of our tasks. Discussion of the implications of these results is presented in the following subsection.

Task 1

We analyzed Task 1 (copying an image using pressure-sensitive painting) by defining 22 features, such as “line 2 is lighter than line 1”, and “line 3 demonstrates the correct brightness gradient”, and coding errors on each of these features for each participant. The resulting drawings can be seen in the top row of Figure 6-10 and the per-participant correctness is shown in Figure 6-11. Across our six participants, the mean accuracy was 93.9% (sd = 4.7%). In short, *all participants were able to effectively manipulate pressure to control brush darkness in a drawing task.*

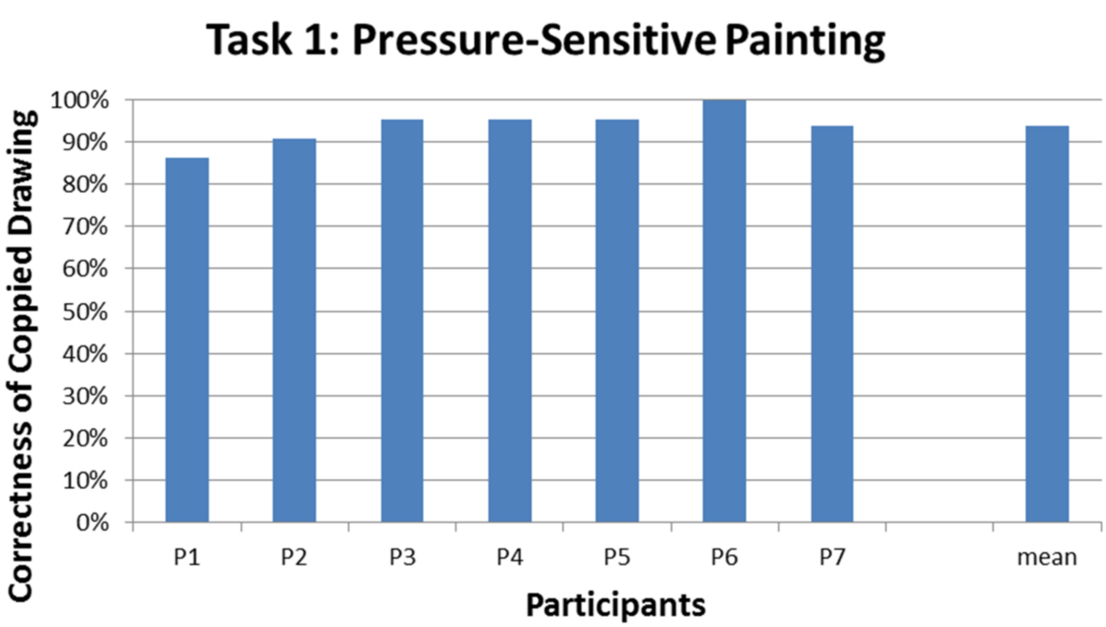

Figure 6-11: Results from analyzing the accuracy of partipants’ attempts to use pressure sensitive painting to copy an image in Task 1. Correctness is calculated over 22 features, such as “line 2 is lighter than line 1.”

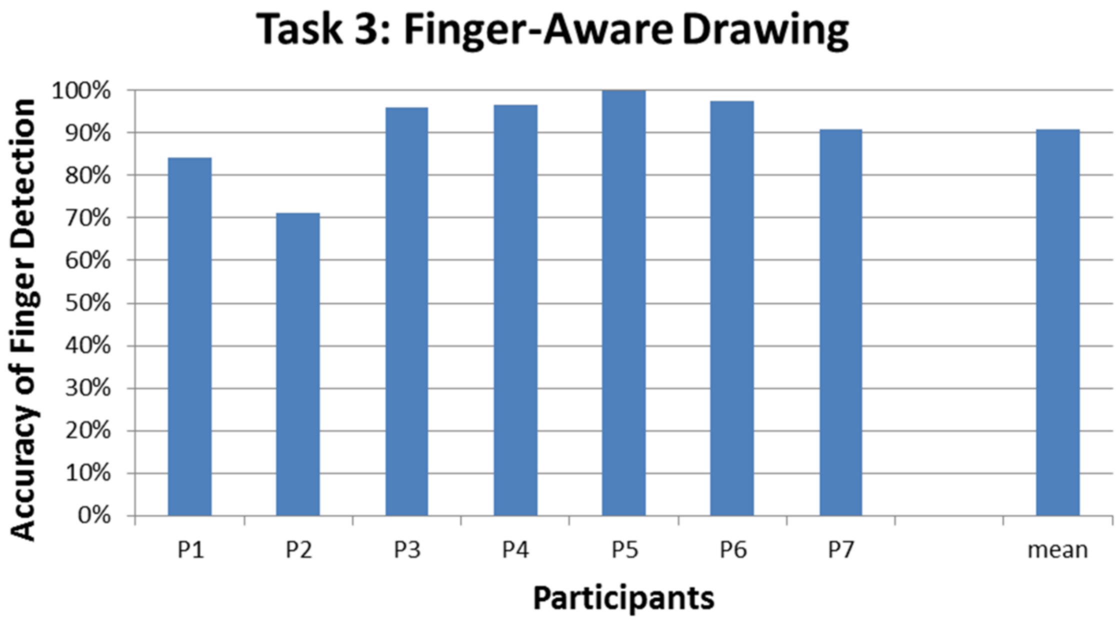

Figure 6-12: Results from Task 3: finger-aware drawing. Accuracy of finger detection is calculated as the percentage of lines that are drawn the correct color.

Task 2

The task of copying a multi-color image is more open-ended and therefore difficult to formally analyze, as participants used different numbers of strokes to complete the image. Anecdotally, success on task 3 (vertical lines) was indicative of users' ability to perform task 2: *while all six participants completed the target drawing* (middle row of Figure 6-10), *one had some difficulty reliably selecting the finger color*.

Task 3

We analyzed Task 3 (finger-aware drawing of alternating blue and green vertical lines) by computing the percentage of lines drawn in the correct color for each participant (see bottom row of Figure 6-10 and Figure 6-12). Across our six participants, the mean accuracy was 90.9% (sd = 11.1%). This includes P2 for whom finger classification did not perform at a level comparable to the other participants. In this errant case, the classification was biased toward one finger, resulting in an accuracy of only 71%. Without this participant, the mean accuracy overall was 94.8%. In short, *five out of six participants were able to effectively specify brush colors by painting with different fingers.*

Task 4

We analyzed Task 4 (writing numbers and selectively erasing half of them with the "undo flick" gesture) by counting the number of false-positive and false-negative "undo" operations performed by each participant. All participants but one completed this task with no errors. The one participant had two false positive errors. In short, *five out of six participants were able to reliably execute and control the "undo flick" gesture without any false positives.*

Task 5

We analyzed Task 5 (picking and throwing images) by counting the number of "mis-triggers" and "mis-classifications." Mis-triggers were instances where the system detected a pinch or throw gesture that the user did not intend, or failed to detect an intended gesture. Mis-classifications were instances where the system correctly detected the presence of a "pick" gesture but failed to correctly identify the gesturing finger. Three of our six participants performed this task without any errors of either type. Two of the remaining three participants experienced no mis-triggers, but had 2 and 3 mis-classifications, respectively. The remaining participant experienced 2 mis-triggers and 1 mis-classification. In short, *this was the most difficult of our interactions, but the three perfect executions of this task support its basic feasibility*. In the following subsection, we will discuss hypotheses surrounding the classification errors experienced by the other participants.

Discussion

Here we discuss the lessons learned from developing our system and testing our interaction techniques.

Calibration and Training

One goal when developing sensing and recognition systems is to construct an accurate model that requires minimal calibration and training. Our system currently requires gross calibration each time a user dons the EMG device sensors. This comes in the form of a making a tight fist and then relaxing each of the hands. Because of the variance in muscle activity across users and the inconsistency in sensor placement, even for repeated use on the same user, this is necessary to normalize the raw amplitudes and find the basic working range of the signal. This calibration provides sufficient information to model pressure gestures, pick and throw gestures, as well as our flick gesture, since these function based on thresholds set on the signal amplitude.

Other gestures such as distinguishing between different fingers require more training since the relationship between the raw signal and the desired recognition result is less obvious. In these cases, we have users perform tasks in which we collect labeled data that can be used by machine learning techniques to dynamically build the classification model. Subsequent to our experiment, we added support to our system for using the wireless muscle-computer interface armbands from Chapter 5. Informally, we have used the system successfully to interact with the system using only training data from a previous session on a previous day.

We believe that this training exercise must be carefully designed to collect data that is representative of real use scenarios. For example, our other muscle-sensing training methodologies have employed a stimulus-response paradigm, in which the user is told exactly which gesture to perform and when. In the case of finger identification, we could have gathered training data by asking users to press down with each of their fingers and hold them there in a single spot. This is not only potentially boring, but also provides data that is quite different from that which has to be recognized during drawing, where the finger is in constant motion.

In our tests, we had users paint images of their choice while using fingers of our specification, which was a much more compelling exercise that provided better training data. Even then, some users performed the training very differently than they did the task. While we cannot quantify this, our informal observations of the user that had poor recognition results leads us to believe that they were trying so hard to train the system correctly that their arm might have been abnormally tense when they did this, leading to the construction of a poor model.

These issues point toward a limitation of our current system. We did not explicitly tell users that they had to perform the tasks and gestures in any given way, and we found that users who deviated most from the way they trained the system generally had the worst recognition results. This is hardly surprising, but most users were able to naturally self-correct after the training phase, and with a few minutes of practice, quickly learned how to perform the gestures in a way so as to get reliable classification.

Classification Limitations

The recognition rates achieved by our system – for example the 90% mean accuracy for finger identification – might be considered low when compared to the error rates of standard input devices such as mice and keyboards. However, our accuracies are comparable to other “non-traditional” input modalities such as speech and gesture, both of which have achieved success in a variety of applications where the benefits of alternative modalities compensate for reduced accuracy. In addition, we believe that a more synergistic combination of touch sensing and muscle sensing would probably yield better recognition results. For example, we could consider the changes in the touch contact area as well as the outline of the hand in the hover zone to further aid our recognition system in detecting pressure and which finger is in contact with the table.

Gesture Sets

In this work, we only classify a single contact at a time. This is not an intrinsic limitation of the approach, but rather one of implementation. It remains future work to develop recognition techniques that deal with compound gestures, whether through training explicitly for these gestures or by inferring them from models of the individual gestures. If multiple fingers are touching the surface at the same time, the system could also use the relative position and ordering of the fingers and the information about which fingers are currently touching the surface to infer which finger is which. Even minute changes in pressure and finger flex could be correlated with minute changes in finger contact area and relative position to other contacts to precisely identify each finger in contact with the surface.

In our experiment, one of our explicit design decisions was to utilize only the index and middle fingers. This was a simplification since we sought to explore modality fusion rather than explicit muscle-sensing system performance. That said, in previous chapters we have shown that the recognition accuracy does not degrade drastically even when people use all five fingers. It should be noted that we expect that the natural way to use the thumb on a surface is probably not equivalent to the best-case scenario tested in our other work and we would likely see slightly degraded performance there; the muscles controlling the thumb are less accessible to a forearm EMG sensor than the muscles that drive the other fingers.

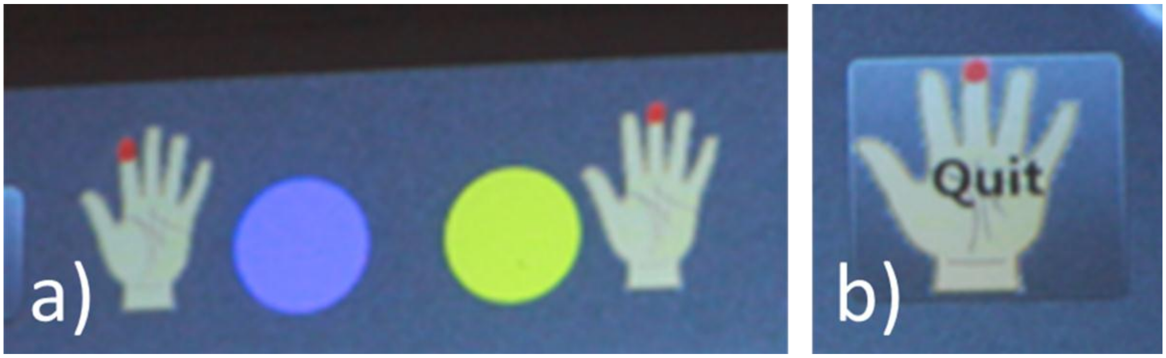

Figure 6-13: Finger-dependent UI elements: (a) finger ink wells for choosing the brush color of index and middle fingers, and (b) middle-finger quit button to reduce accidental activation.

Interaction and Interface Considerations

A slightly more optimistic idea is to make use of a unique property of muscle-sensing: it is sometimes possible to detect a physical movement event before it actually occurs. This is because before we make a motion, we have preparatory muscle activation that can be sensed by EMG. Hence, it may be possible to detect actions such as pressing a button slightly before the physical event actually occurs, which could perhaps be integrated into interaction techniques for tabletops to, for example, begin animating a change to an object that will be affected on screen.

In our prototype system, we implemented and evaluated each of our interaction techniques separately. However, these can obviously be integrated into a single system. Pressure sensing can be done simultaneous with finger identification and the surfaces ability to sense contact shape for hybrid interactions such as simultaneously controlling stroke shape, color, and saturation. Similarly, finger identification on the surface (e.g., painting) and finger identification off the surface (e.g., pinching) can be inferred simultaneously through separate classifiers while using surface contact information to determine how to use the results.

In addition to our hybrid interaction techniques, we explored the concept of finger-dependent user interface elements (see Figure 6-13), i.e., on-screen elements that can be activated only when touched with a specific finger; this is similar to the concept introduced in (Sugiura and Koseki 1998). We prototyped finger-dependent ink-wells for selecting the finger brush color, and middle-finger quit button for exiting our application. Such elements are harder to activate by mistake than standard widgets, which could be useful for actions with high cost of accidental activation (e.g., delete or quit).

6.3 Summary

In this chapter, we have presented two applications of our muscle-computer interface techniques: Air Guitar Hero and an augmented interactive surface. In our air guitar experience for the Guitar Hero video game, we demonstrated that our bimanual approach to classifying finger level gestures is both fast enough and robust enough for real interactive applications (even when the application itself was not designed specifically for a muscle-computer interface). Our application of muscle-sensing to interactive tabletops highlights the opportunity to enhance existing interaction techniques with novel sensing modalities. Our work shows that the interaction vocabulary for interactive surfaces can be increased in a useful way by using muscle-sensing to determine additional properties of users' hands and fingers, such as which finger is touching the surface, with how much pressure they are pressing, and if they are continuing a gesture while no longer in physical contact with the surface.

In both of these applications, the key to making them effective human-computer interfaces lies in coupling the interaction design to constraints of the sensing technology. Both systems attempt to design around the latency inherent in our approach to classification. In the case of Air Guitar Hero, we minimize the latency of detecting strumming because it is the most time sensitive aspect of the game, while players are able to anticipate note selection and begin pinching their fingers early enough that the latency of our system has minimal effect on performance. In our interactive tabletop application, the system begins drawing a semi-transparent stroke until the system is confident of the specific finger the user is employing and then later fills in the stroke with the correct color (usually while the user is still making the stroke).

In the next chapter we discuss several more potential application of muscle-sensing as well as several other on-body input and output modalities.

Chapter 7
Future Work

Creating new human-computer interfaces that expand the situations and environments in which people can effectively use computing is a large problem space with applications in many fields. In the previous chapters, I have explored one particular avenue of interaction that enables people to use computing in many situations where traditional interfaces are inadequate. This line of research could be furthered by taking muscle-computer interfaces beyond the laboratory through deployments with people in their everyday lives. Beyond muscle-computer interface, there is also a much larger related research agenda possible in developing more techniques to harness the untapped bandwidth of the human body for new interfaces to computing, including for output. In addition, I think employing more forms of physiological sensing has the promise not only to create human-computer interfaces that can be used beyond traditional office work settings and in everyday activities, but also to create better interfaces for people with physical impairments. In this chapter, I describe several areas of future work for further supporting everyday activities through always-available mobile computing.

7.1 On-Body Input

On-body input provides the ability to interact with a computer without necessarily requiring traditional physical transducers such as buttons and touch screens. Muscle-computer interfaces are one such on-body interface. This section describes ways the muscle-computer interfaces in this dissertation could be improved as well as extended using other types of sensors. I also describe another form of on-body input I have explored at the prototyping stage: tongue-computer input.

7.1.1 Improving EMG-Based Input Techniques

While we have had promising results classifying finger gestures in the lab using a wired device and have a wireless device, there remain several open challenges to making muscle-computer input feasible for everyday mobile use.

Orientation Detection: Multiperson Models and More Robust Multisession Models

In the ideal case, people would be able to quickly slide on their muscle input armband and immediately begin using it. Thus far we have been able to partially achieve this goal; while, classifiers were trained and tested independently on data from each user with a nearly fixed-placement of electrodes, we have demonstrated that if trained once, we can recognize finger gestures from that same person in later sessions. However, in all our work we have created a new classification model for each new user. In this scenario, every user has to train the system at least once before the system can recognize his or her gestures. In our simple attempts to build *cross-user* models, we have seen our accuracies drop significantly. One approach to improving our ability to build *cross-user* models and more robust *cross-session* models is to attempt to select a subset of features that are more invariant. Additionally, for a given feature set, it may be possible to do a simpler calibration step for some features instead of collecting a whole new training set. Similarly, if the armband has a dense array of sensors, it might be possible to use a robust signal such as a fist clench to determine the orientation of the armband relative to the muscle so a global model can be adapted to any shift in electrode location on the arm.

Leveraging a Physiological Model to Improve Recognition

In our work thus far on finger gesture classification, we have primarily used support vector machines to classify gestures. However, this approach does not leverage additional information available about the classification problem. For example, we have observed that when classifying pinch gestures, the middle finger pinch “signal” is so strong that it is very rarely misclassified as another finger. However, the recognizer often flips back and forth quickly between index and middle during an index finger pinch. Similarly, during a ring finger pinch, the classifier’s posterior probabilities for ring and middle finger are very close. A simple improvement here would be a rule based approach requiring middle finger pinch to be very likely to classify as such. In future work, it may be fruitful to explore more ways in which other knowledge about the problem, such as the physical layout of the fingers, can improve recognition results. One possible approach for incorporating structured knowledge into our gesture recognition approach is Transfer Learning and Advice Taking (Torrey et al. 2006). In Transfer Learning and Advice Taking, an SVM, for example, can be given soft constraints that provide additional guidance in classification beyond what is captured in training data.

7.1.2 More Sensing on the Arm

While we have demonstrated that muscle-computer interfaces by themselves can provide rich input, adding other types of sensors to our armband form factor could further increase the possible interaction vocabulary for an armband platform. For example, adding an accelerometer and gyroscope would give the ability to combine arm motions with finger gestures. For example, imagine changing a continuous value (such as the volume of a portable muscle player or mobile phone) by reaching forward, pinching with your index finger, and then slowly rotating your forearm clockwise or counter-clockwise until the desired volume is reached.

Another gesture sensing approach that could be combined in the same form factor of a forearm band is sensing the propagation of mechanical waves through the tissue and bones of the fingers, hand, wrist, and forearm. Recently Harrison, et al. demonstrated their Skinput system’s ability to resolve the location of finger taps on the forearm and hand using a novel arm-mounted cantilevered piezoelectric acoustic detection mechanism (Harrison, Tan, and Morris 2010). Combining this sensing approach with muscle-sensing could bring together the best of both sensing techniques for an even richer gesture input system. For example, muscle-computer interfaces are particularly good at sensing continuous muscle exertion and changes in that exertion (e.g., which finger is currently pinching, pressing, or pulling and with how much pressure) while the Skinput system is good at detecting the timing of instantaneous events (such as taps with fingers from the contralateral hand) and where those taps impact the arm and hand. Combining these sensing techniques together could create an interface that is both very good at

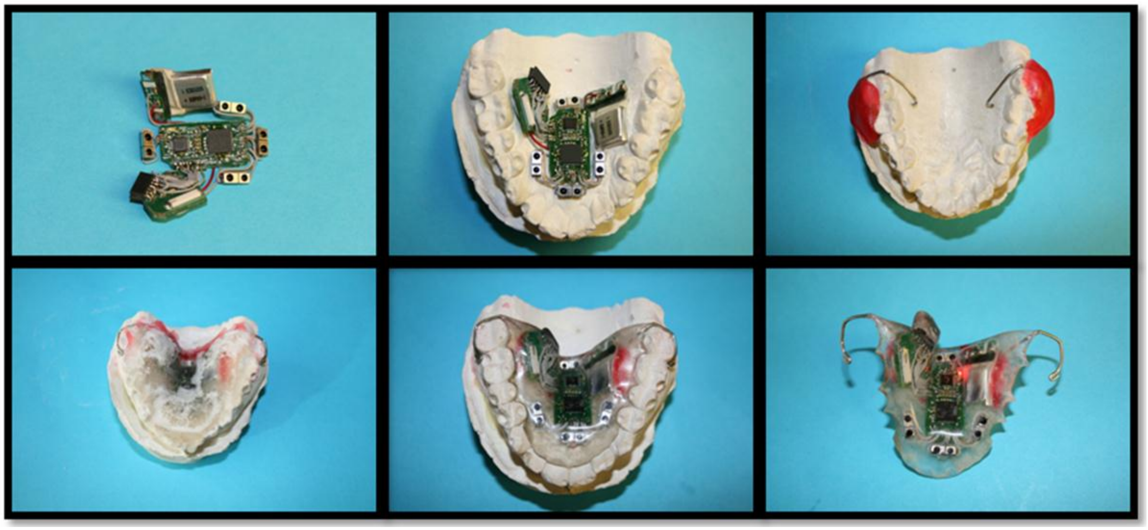

Figure 7-1: Process of building our infrared based wireless tongue computer interface embedded in an acrylic retainer.

time resolution of when a gesture begins as well as allowing continuous variation of a control and robust detection of the fingers involved.

7.1.3 Tongue-Computer Input

Physiological interfaces could also aid people who are bound to beds or wheelchairs in hospital, home or workplace settings. These interfaces could be useful across a diverse spectrum, from those with temporary limitations to those who have suffered debilitating spinal or brain injuries. People in these situations cannot make use of traditional human-computer interfaces. For example, the keyboard, mouse, and display of a laptop in their traditional forms are difficult to use when lying down, yet people who spend all or most of their time in a hospital bed (even if only for weeks or months) can benefit greatly from being able to use a computer. We have recently begun working on a new interface technology supporting this need. We have explored methods to optically sense tongue movements from the physical form factor of a dental retainer. Thus far, we have created a system based on infrared proximity sensing that can be entirely embedded in the acrylic of a retainer and sense movements for real-time control (see Figure 7-1 for the process of creating our infrared based wireless tongue computer interface embedded in an acrylic retainer; see Figure 7-2 for a larger photo of the finished retainer) (Saponas, Kelly, et al. 2009). As a first demonstration of our system, we have built an interface to the game Tetris. In our interface, we map several tongue gestures to the controls in the game: a game piece is moved left by swiping the tip of the tongue from right to left across the back of the front teeth; similarly,

Figure 7-2: Our infrared based wireless tongue computer interface embedded in an acrylic retainer.

moving a piece right is accomplished by swiping the tongue from left to right; rotating a piece clockwise is done by taping the tongue against the palate (top) of the mouth; and dropping a piece to the bottom of the screen is carried out by pressing the tongue against the palate and holding it there for one second. Our interface to the game of Tetris is meant both as a demonstration of the ability to use infrared proximity sensing in the mouth to detect tongue gestures as well as a mechanism for potential users to get familiar with using a tongue based interface.

We have also built a first prototype extending this capability to controlling the movement of a motorized wheelchair (see Figure 7-3). We modified a traditional motorized chair such that we can control the chair’s direction and speed via our tongue-computer input technique. A limitation of the communications protocol of the chair we modified is that it only allows us to turn the chair left, turn the chair right, move the chair backward, or move the chair forward. Unfortunately, it does not allow us to simultaneously turn the chair and move the chair backward or forward simultaneously so that it would enable a driver to execute a turn around a corner and object as a single continuous turn. In our implementation of a tongue interface for controlling the motorized chair, we took a similar approach to our interface to Tetris. We mapped swiping one’s tongue left or right to turning the chair left or right. However, instead of making it a single discrete action, we added the feature that after swiping their tongue in a lateral direction, a user then holds their tongue against that side of their mouth while the chair turns. To stop turning and leave the chair facing the current direction, the user simply returns their tongue to a rest position in the center of

Figure 7-3: A person using our wireless tongue input platform to drive a motorized chair through an obstacle course.

their mouth. To make a small correction by moving the chair backwards a couple feet, users must just tap their tongue against the top of their mouth. And to initiate forward movement, users can hold their tongue against the top of their mouth for approximately one second. Once forward movement begins, they can control the speed of the chair continuously by moving their tongue up and down in their mouth. As with the other types of movement, if they drop their tongue back down to a rest position, the chair stops all motion.

These first prototypes of tongue-computer interfaces highlight the opportunity to make use of the tongue for input as well as the larger opportunity to further utilize the untapped bandwidth of the human body for computer input.

7.2 Output for Always-Available Computing

This dissertation primarily explores one novel mode of computer input: muscle-computer input. However, supporting everyday activities through always-available mobile computing must eventually include novel solutions in the space of computer output to complement and support mobile input techniques such as muscle-computer input. In this section, we describe several opportunities for developing new modes of computer output especially appropriate for always-available mobile computing.

7.2.1 Visual Output

The most common and perhaps the richest form of computer output is visual output. Desktop and notebook computers make use of CRTs and LCDs to display graphical desktops, office productivity applications, web browsers, and more. This is most commonly paired with keyboard and mouse input for entering text and clicking on widgets. It makes sense that traditional computing makes so much use of visual feedback as humans' visual perception abilities are vast. We can quickly read or skim text, enjoy images and videos, as well as interpret charts and graphs.

In mobile settings, it may often not be possible to equip people in a practical way with displays of much resolution. Certainly, there are displays that can be clipped onto a pair of glasses with adequate resolution for some typical use of displays. In many situations, though, such a large device might not be necessary or ideal as the right type of visual feedback to pair with an always-available input technique such as muscle-computer interfaces. For example, if someone is out jogging and controlling their portable music player, just having a ten character display integrated in the lenses of their sports sunglasses would be enough to scroll by information such as the title of a song or the subject of an important email. This size display might be able to be integrated directly into the lens of sunglasses in the near future and have lesser requirements than trying to fit a "full" display on a pair of sunglasses. I think this is an important area of future work, extending on-body interfaces by pairing them with the right visual output.

7.2.2 Audio Output

In desktop computing, audio output is used for feedback after clicks, sound effects in games, and for listening to music. Arguably, software has made even more use of audio in mobile scenarios. For example, Bluetooth headsets use certain tones to indicate when a user has successfully turned the headset on or off, changed the volume, or has an incoming call. Future work in this area could build on previous work on non-language audio output, such as Earcons (Brewster, Wright, and Edwards 1992). This is, of course, in addition to being able to use the headset for listening to the voice of a caller or enjoying music. One could also imagine using a Bluetooth headset in conjunction with an always-available input technique such as muscle-computer interfaces to indicate when the system is busy and not ready for input, when the system has detected a gesture and is attempting to recognize that gesture, identify which gesture was detected, or when the system has been successfully turned on or off. These types of uses of audio output for mobile computing remain future work, but they could make for an important component of an always-available interface for mobile computing.

7.2.3 Tactile Output

Tactile feedback has been explored for many applications from sensory substitution (Bach-Y-Rita et al. 1969; Bach-Y-Rita, Tyler, and Kaczmarek 2003; Kaczmarek et al. 1991) to navigation (Van Erp et al. 2005; Tsukada and Yasumura 2004; FeelSpace Belt Project). An important area of future work is exploring how tactile feedback can be combined with muscle-computer input to create a full interaction loop away from the hands and eyes. An always worn tactile device could also be used for applications such as longer term ambient awareness. In this subsection, we highlight several opportunities for tactile output in mobile computing and suggest a vibrotactile form factor that could complement the forearm based muscle-computer input techniques presented in this dissertation.

Armband Vibrotactile Output

Similar to our muscle input armband prototype, one could build a wearable armband with embedded vibrotactile feedback. Our vision is to place two parallel rings of vibration motors around the forearm (see Figure 7-4a). In a first prototype of this vision, we explored creating a ring of four vibration motors sewn into a custom armband. We used an embedded Arduino microcontroller (Arduino), battery, and custom PCBs with vibrotactile motors mounted to one side of each board (see Figure 7-4b). We experimented with vibration intensities, frequencies, and vibrating the motors in spatial patterns. The conclusion of our first exploration is that this could definitely be a fruitful path for output in an always-available interface, but such an approach requires more actuators. We suggest for future work placing two rings of vibrotactile actuators approximately four centimeters apart. Within each ring, the vibrators could be approximately one centimeter in diameter and spaced two centimeters apart. These distance choices are partially based on the previous work where linear vibrotactile arrays have mostly used two centimeter spacing within an array. This spacing would also separate the two rings as much as possible while still fitting on a sports armband. We would suggest first creating this tactile armband as a separate platform from the muscle input armband. However, the eventual goal would be to integrate the two devices together into one armband suitable to be worn for an entire day under a person's shirt.

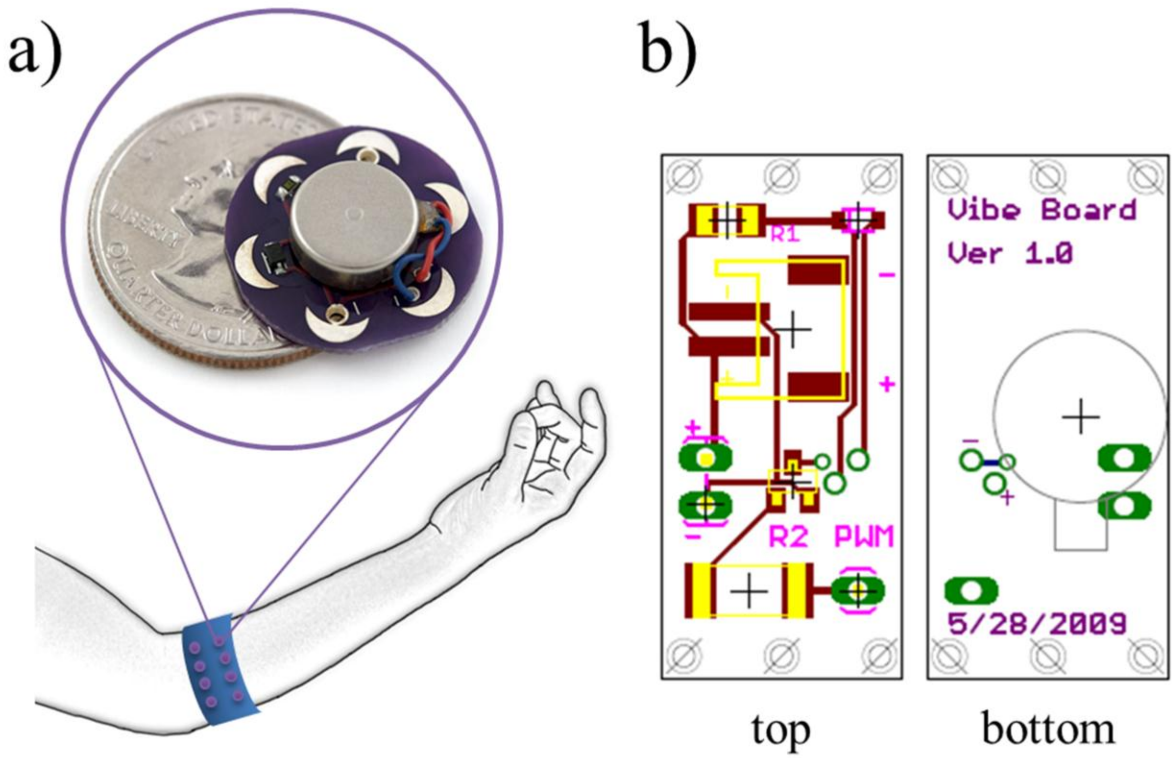

Figure 7-4: a) drawing of a potential vibrotactile armband b) layout of our vibrotactile satellite PCB

Interactive Feedback

When we touch buttons, sliders, keys and other physical input devices, we use our sense of touch in our fingers to understand the state of the system. We can tell, for example, whether a button is pushed all the way in. We use this feedback in a motor control loop to precisely and accurately manipulate these controls. In many cases, we combine this with our sense of sight to either look at a physical control or at a corresponding screen or indicator light to determine what action to take or if our actions resulted in the appropriate change in system state. Similarly, our muscle-sensing based input approach still requires some type of feedback mechanism so users can accurately control devices. In some cases, such as controlling a portable music player, there may be audio cues such as the song changing that indicate system state. However, in many potential scenarios a more immediate and direct form of feedback might better enable users to control the system with muscle-computer input.

In future work, we would like to explore ways to use our proposed vibrotactile armband to provide computer output such as alerts, indicating the system is on, indicating the system is busy, and indicating the gesture or command that was just recognized by the system. Some of the approaches we have considered include creating a sensation of motion, relative speed, or direction of a line “being drawn” using the rings of vibration motors. For example, one ring might be slowly going around in a circle to indicate the system is on versus going around quickly to indicate the system is processing. Similarly, the other ring might “draw a line” in one direction to indicate the “next” command was recognized and the reverse direction to indicate the “previous” command was recognized. Other techniques might include some type of pulsation to indicate an error state or that a command was not recognized. Similarly, the frequency of vibration could be used to indicate a relative value, such as the volume setting, after the value is changed. The goal of this type of future exploration would be to enhance muscle-computer input with an accompanying tactile feedback mechanism for eyes-free mobile interfaces.

Awareness

One important role computing can play in our lives is keeping us aware of long term state or physically distant events. For example, researchers have created systems that use ambient displays in the home to keep family members aware of the activity level of an elder (Consolvo, Roessler, and Shelton 2004; Mynatt et al. 2001). Others have used the background image on a mobile phone as an ambient display to provide awareness of one’s fitness level (Consolvo et al. 2008) or “greenness” (Froehlich et al. 2009). There are also a wide range of commercial ambient displays that convey the state of the stock market through glowing orbs on a desk (Ambient Devices) or the sidebar on a desktop display. In future work, we want to extend these ideas of ambient displays to “wearable” ambient displays.

There are several approaches one might take in employing our proposed vibrotactile armband platform for conveying awareness information. For example, the two linear tactile arrays could “display” state by being off versus continuously “drawing” lines with one or both rings, drawing lines in the same direction versus opposite directions, or drawing lines quickly versus slowly. An interesting direction of future work would be investigating the extent to which people can differentiate these different tactile sensations.

In addition to exploring whether these techniques can convey awareness information, it would also be important to determine whether certain approaches are distracting or whether people stop noticing them consciously and just incorporate them into their decision making process. In some cases, it might be as simple as presence; that is, utilizing our ability to ignore a tactile feeling until it is gone. For example, many people are not always consciously aware that their wallet is in their back pocket, but when they get up from their chair and accidently leave their wallet behind, they might notice as they walk away that their wallet is no longer there. This is because we get accustomed to feeling the presence of a wallet in our pocket without consciously noticing. For some people, they can subconsciously multiplex many of these feelings simultaneously: a wallet in a back pocket, keys and mobile phones in side pockets, a watch on the wrist, or rings on their fingers. Wearing an armband on each forearm might allow us to, at a minimum, create similar feelings of presence or absence that are not distracting but can convey some longer term state. Ideally, future work would uncover that our armband approach is capable of delivering many “bits” of awareness information unobtrusively.

7.3 Applications

In this section, we explore potential future work in the area of applications from the space of always-available mobile computing.

7.3.1 Note Taking

One need that pervades many people’s jobs is browsing previous notes and documenting tasks. The environments in which people conduct their work often limit their ability to use traditional note taking technologies such as a paper and pen, a keyboard, or voice annotation. For example, in many medical settings, it is important for clinicians to both consult notes and record events, yet their jobs require continuous use of their hands and also leave them with little or no down time to take out and use a computer. This example highlights an opportunity for future work to develop interfaces that enable computing to play an even broader role in occupations outside the office cubical.

7.3.2 Microinteractions

In addition to developing novel methods of interacting with computers in the workplace, I also want to create better support for everyday activities where the human-computer interfaces of a mobile phone or notebook computer are not suitable. These include home activities such as taking care of an infant and preparing food, hobbies such as gardening and pottery, and fitness activities including walking, jogging, and weightlifting. New methods of interaction that enable people to better use computing as a part of these activities can both make these activities more enjoyable as well as allow completely new computing experiences to emerge surrounding these activities. For example, when someone is working with a material that sticks to their hands, as in cooking or many crafts, not only are their hands busy, but they are coated in a material that could damage or dirty a mobile phone's touch screen. A physiological sensing based interface could allow people to look through recipes, browse sketches, answer a phone call, and control their music player while their hands are still dirty.

This is an essential area of future work for always-available interfaces that we refer to as microinteractions: enabling people to have a quick interaction with their mobile devices that is as fast or faster than pulling today's mobile phones out of one's pocket, unlocking them, launching an application, and interacting with the application. Ashbrook, et al. have conducted some initial work establishing this area of research (Ashbrook et al. 2008; Ashbrook, Lyons, and Starner 2008). They investigated the access times and error rates for watch-based interfaces enabling people to quickly touch the bezel of their watch to interact with their mobile computing environment. In future work, their ideas could be extended to leverage even more of the body's untapped bandwidth for microinteraction techniques in support of people's everyday activities.

7.3.3 Microbreaks: Utilizing Mental Idle Time for Learning (and other tasks)

Microlearning is an approach to learning where a large topic is broken down into a collection of smaller, more manageable, pieces that may be studied through lessons as short as a few seconds or minutes. Microlearning has been explored extensively in the context of distance learning on desktop computers. More recently, the microlearning approach has been extended to use the microbreaks in our day. For example, researchers have built a system in a residential research laboratory that detects opportune times to prompt the inhabitants with words and phrases in a foreign language (Beaudin et al. 2007). A potential area of future work is extending this application of microlearning to other microbreak scenarios that occur throughout people's day outside the home.

Spaced presentation of words and phrases has been shown to substantially increase vocabulary learning (Dempster 1987). Conventional vocabulary study methods spread learning through daily lessons over many weeks or months. However, finding time for daily lessons is often difficult for adult learners with busy lives. Throughout our day, there are many potential opportunities for learning where our mind is idle but our environment is too physically demanding to read a textbook or even interact with most mobile devices. For example, when we are walking to the bus in the cold wearing gloves, driving to work with our hands on the steering wheel, or holding a box while waiting in line at the post office, we can listen to language lessons through headphones, but our hands are not available to control these lessons. If instead, people had an always-available interface that allowed them to initiate and control a language lesson whenever they desired, they could learn vocabulary more easily and efficiently.

To explore how always-available interfaces can increase people's opportunity for language learning, in future work, researchers could build a vocabulary building application for mobile phones. There are many different possibilities for a mobile phone language learning application. An application could detect and make use of the user's context when choosing what words or phrases to present. Another approach would be to use object recognition such that an application could translate into a foreign language the names of objects that a user photographs. The future work in this area most relevant to this dissertation, would be simply making use of a muscle-computer interface to enable control of language lessons while on the go using a wireless headset and a mobile phone.

7.3.4 Rehabilitative Games

Two areas where physiological interfaces can enhance existing health technologies are in fitness and rehabilitation. In both these areas, physiological interfaces could be used to create more engaging games, exercise instructions, and record keeping. Enhancing existing technology for fitness and rehabilitation can help to better motivate people to stay with their fitness routine or comply with their physical therapy program. Currently, computing only plays a limited role in fitness and rehabilitation; always-available interfaces have the potential to greatly increase computational support for fitness and rehabilitation.

7.3.5 Continuous Medical Monitoring

As we create physiological sensing-based interfaces that have a value proposition strong enough to be worn throughout the day, new avenues will open up for continuous medical monitoring. Sensors worn on people's skin, embedded in contact lenses, or as part of dental retainers will be able to do long-term sensing that has never been done before on a large scale. This sensing has the potential to allow early detection of diseases or recognition of behaviors that lead to injury. For example, back tension or repetitive stress injuries in the shoulders, forearms, and hands might be detectable and manageable—perhaps even preventable. A potential area of future work would be for HCI researchers to partner with medical researchers and clinicians to explore the opportunities for continuous medical monitoring through always-worn interaction technology.

7.4 Summary

In this chapter, we have presented opportunities for future work in always-available interfaces in the areas of on-body input, output, and applications. The main summary point of this chapter is that this dissertation has both demonstrated a novel method of interacting with computers and has also has highlighted many opportunities for computing to be further embedded into our everyday lives supporting us in all of our activities.

Chapter 8
Conclusion

Today's mobile computing devices enable people to access the world's information at the mere touch of a screen. People can check if their bus is on time while waiting at a bus stop, respond to email while riding the bus, and read through their calendar while waiting in line for coffee—all before they arrive at their office to start their traditional work day. While this is a compelling use of mobile computing, I believe computing can and will support us better in the future by further weaving into the fabric of our everyday lives. Imagine also being able to access your mobile devices while jogging, carrying groceries, riding a bike, gardening, working with a patient, conducting research in a wet lab, or preparing dinner. In this dissertation, I propose, develop, and evaluate muscle-computer interfaces, an approach that can put mobile computing at just the twitch of a muscle in scenarios such as these. In this chapter, I revisit my thesis, summarize the contributions of this dissertation, discuss the limitations of my research, and reflect on some of the insights I gained through my dissertation research.

8.1 Thesis

Recall my thesis from Chapter 1:

> *Finger level gestures detected and classified through forearm electromyography can enable an always-available interaction paradigm for mobile computing*

I began demonstrating my thesis by showing in Chapter 3 that it is feasible to classify finger-level gestures through forearm electromyography (EMG). In Chapter 4, I built on this result to demonstrate the always-available nature of my approach by demonstrating that it is possible to detect and classify finger gestures in real-time and when hands are busy. I further showed that this approach could be possible in a mobile setting in Chapter 5 by building a wireless muscle-sensing armband that required no gel or adhesives and can classify gestures with no new training data when re-donned 48 hours after initially training the system. Lastly, I demonstrated the effectiveness of this approach for interactive applications by utilizing it in the Wireless Air Guitar Hero demo and integrating it with the Microsoft Surface in Chapter 6.

8.2 Contributions

In this section, I breakdown the contributions of my dissertation by the concepts and techniques I developed, the artifacts I created, and the results from my experiments.

8.2.1 Concepts and Techniques

In this dissertation, I introduced several new concepts and techniques for classifying finger-level gestures through muscle-sensing. My dissertation research began with the development of a basic technique for sensing finger level gestures while a hand is resting on a horizontal surface in conjunction with four gesture sets for use on horizontal surfaces. Subsequently, I built on this work to develop concepts and techniques for a bimanual interaction technique robust enough to classify finger-level gestures with empty hands in free space as well as when holding objects. Finally, I developed several more concepts and techniques enabling the use of my muscle-computer interfaces with interactive surfaces. Below, I list the complete set of the concepts and techniques introduced in my dissertation:

Finger level gesture classification approach

- Upper forearm ring electrode placement
- A set of features extracted from 6 to 8 EMG channels
- Method for training an SVM using these features

Activation detection techniques

- Threshold approach with calibration
- Gradient detector with frequency band constraints

Bimanual interaction paradigm

- Select plus activate concept
- Smoothing via majority vote with streak-based short circuiting

Gesture sets

- Pressing, using pressure, lifting, and tapping on a horizontal surface
- Pinching with empty hands in free space
- Pulling and squeezing while holding objects
- Drawing, pinching, and throwing in conjunction with an interactive surface

8.2.2 Artifacts

In the course of my dissertation research, I have developed both software and physical artifacts. An important software artifact that I created in the course of this work is the Physiological Sensing Platform, which primarily consists of a Physiological Sensing Library and a tool that provides a graphical interface on top of that library. These two pieces of software provide the ability to capture streaming EMG data, compute features over EMG data, visualize EMG data and features, train and classify in real time with several different statistical machine learning algorithms (built on top of existing proprietary and open-source implementations of the SMO version of SVM), provide several types of stimuli for recording training data from people, and run offline tests including cross-validations with temporally adjacent holdout sets. My Physiological Sensing Platform provides the functionality to quickly carry out exploration of new gesture sets using EMG sensing.

Other software artifacts I developed include a mockup of a portable music player for testing interaction with our gesture sets, an interface to the Guitar Hero video game (on both PC and XBOX) to enable muscle-computer input to the game, embedded firmware for the microcontroller on our wireless armband to compress and transmit EMG data, and a system for combining muscle-sensing based gesture classification with the touch based gesture recognition on a Microsoft Surface.

I began this research using a wired commercial EMG/EEG acquisition device. To move from this tethered approach to a wireless always-available interface, I created several prototypes of our wireless EMG armband. In conjunction with this wireless armband, I also enabled a wireless air guitar experience by modifying an XBOX controller and a Guitar Hero controller so that they could be controlled by software over a USB to serial interface.

8.2.3 Experimental Results

I evaluated the concepts and techniques introduced in this dissertation by employing the artifacts I created in a series of experiments. From these experiments, this dissertation contributes several important results that together demonstrate the feasibility of muscle-computer interfaces as an approach to always-available interfaces for mobile computing. Below I outline these experimental results:

- Classifying finger-level gestures via forearm EMG is feasible (see Chapter 3)
- Ability to differentiate (see Chapter 3 and Chapter 4)
    - Tapping among five fingers
    - Lifting among five fingers
    - Pressing using two levels of pressure with index and middle fingers
    - Pressing in two different finger postures with index and middle fingers
    - Pinching with four fingers
    - Squeezing with four different fingers while holding an object
    - Pulling up on the handle of a weighted object with four different fingers
- Classify finger gestures regardless of forearm rotation (see Chapter 4)
- Classify finger gestures on a separate day without new training data (see Chapter 5)
- Ability to combine touch with muscle based gestures on MS Surface (see Chapter 6)

8.3 Limitations

Many of the limitations of this dissertation are implicitly described in the previous chapter through my discussion of future work. In this section, I briefly highlight the most important of these limitations.

8.3.1 Lack of Cross-Person Classification

Throughout this dissertation, all of the finger-gesture classification work I present only attempts to solve the problem of a system learning each user's gestures independently. This means that for any given new user of the system, that new user must spend some amount of time training the system to recognize their gestures. In the few attempts I made to generalize my techniques to building classification models that would generalize to other people, the recognition accuracies were very low. I do not think this is inherently a limitation of muscle-computer interfaces; there may be ways to build models that work across people. However, my dissertation does not explore this avenue of work.

8.3.2 Deploying Muscle-Computer Interfaces in the Wild

All of the experiments in this dissertation were conducted in a laboratory setting. The primary reason for this approach is the physical constraints of our data acquisition devices. The first device we used (the BioSemi Active Two system) required a physical connection to a computer. We experimented with building a belt pack to wear the device around the waist and connect it to a laptop in a backpack. However, we found this too cumbersome to employ in a study outside of the lab. We also developed a wireless armband platform that could transmit approximately 30 feet with good signal quality. Unfortunately, this prototype was not physically robust enough for a field study and broke easily when it came in contact with other objects (for example, wire leads could easily be pulled off of the PCBs and this requires delicate re-soldering). These frailties are definitely not inherent to muscle-computer interfaces. Future iterations of our wireless armband system will be physically robust enough to be worn throughout everyday activities.

8.3.3 Combining Muscle-Computer Input with Non-Visual Computer Output

While our system does provide visual output in the form of classification output incorporated in our mockup of a portable music player and the Guitar Hero video game, visual output is not always the best method of computer output. For example, in many of the real-world scenarios I use to motivate this dissertation, people are eyes-busy with their primary non-computing task. In these situations, muscle-computer interfaces will require non-visual forms of output. A limitation of my dissertation is that I have not yet explored methods for combining non-visual computer output with my muscle-computer input techniques.

8.3.4 Richer Input

Although the gestures sets I presented in this dissertation are sufficient for many application scenarios, richer input vocabularies will enable always-available interfaces to support even more of the activities in our everyday lives. Two logical extensions to my work are 1) expanding the available gestures by creating compound gesture set such as *which finger* is in use and with *how much pressure* is that finger pinching (or squeezing, pulling, pressing) and 2) combining muscle-based gestures with other modes of on-body sensing such as accelerometers. My dissertation work does present some results in both of these areas in that I demonstrate the ability to differentiate two levels of pressure in combination with the index and middle finger as well as the ability to fuse muscle-sensing data with touch data from an interactive surface. However, this is an important area wide open for future work.

8.4 Reflections and Insights

In addition to the contributions and limitations of my dissertation that I have outlined above, in this section I reflect on two insights I find particularly relevant to future research in the area of human-computer interfaces for always-available mobile computing.

8.4.1 Limitations and Opportunities of Physiological-Sensing-Based Interfaces

A key to creating great experiences with new user interface technology is fitting the interaction design to the strength of that new technology. Perhaps the worst use of the muscle-computer interfaces I have presented in this dissertation would be trying to map the types of gestures I can classify to mouse and keyboard input for use in a traditional windowed desktop setup. To do so would be to ignore what is unique and great about muscle sensing as a technology for computer input and instead try to shoehorn it into an existing interaction paradigm.

What is advantageous about muscle-computer input is that it can give people an always-available method for interacting with their computing environment. In mobile computing, muscle-computer interfaces can provide not just a faster way to, say, skip to the next song on a phone's music player application, but also change when and how we use computing. For example, presently people would not even attempt to make use of their mobile phone or laptop when working in a pottery studio because having your hands covered in clay and water is inherently incompatible with mice, keyboards, and especially touchscreens. However, muscle computer interfaces create a new opportunity to perhaps review design sketches or images that someone has captured for inspiration and even to perhaps take photos or notes of their work while in progress. Similarly, muscle-computer input has the potential to enable someone to interact with a language learning application while on the go or silently send a short "I'm running late" text messages to a colleague. The main takeaway here is that the most compelling opportunities for muscle-computer input (by itself or in combination with other sensing modalities) is in creating entirely new computing experiences.

8.4.2 Conducting Experiments with "Brand New" Interaction Techniques

One of the challenges I encountered in conducting my dissertation research is that my experiments asked participants to engage in a "brand new" interaction technique. This is a challenge because participants have little prior experience to draw on when interacting with my system. Certainly, finger gestures like pinching and pressing are advantageous because they draw on humans natural and learned abilities to use their fingers to grab and manipulate objects; but, the concept of pinching "in a repeatable" manner and timing those pinches with a contralateral hand squeeze is very different from most people's prior experience.

This challenge should, perhaps, not be regarded as surprising. When people learn new interfaces such as the controls of a motorized vehicle or playing a first person shooter game with thumb sticks, their performance typically begins at a low level and takes time to increase to an acceptable level such that someone is able to be licensed to drive or competitive in a game. Surprising or not, I still found this to be an important challenge to consider and address in designing and conducting my experiments. Finding the right set of instructions to explain the new gestures being used by participants was vital to successfully evaluating the system. In particular, I think the best insight I had when explaining the pinching gesture was to tell people to "press hard enough to dent a tomato but not hard enough to break the skin" so as to elicit a repeatable and not-tiring gesture from my participants. Going hand in hand with that explanation was giving people a practice period at the beginning of each segment of our experiment to practice a gesture set.

This challenge also complicates the interpretation of the results from my experiments. One might argue that because our experiments were conducted in a laboratory setting, the system's classification accuracies established a performance ceiling (when taken out of the lab, one might assume performance will degrade). However, I argue that it is the case that these accuracies do not represent a ceiling and may, in some cases, represent a performance floor. In all of our experiments, we saw the system perform nearly perfectly for some of our participants. It is possible that this could be attributed to either some amount of chance or a natural distribution among the population. Instead, I believe it demonstrates that people for whom our gesture sets came more naturally were able to make very repeatable gestures and simultaneously concentrate on the task in the study and not make mistakes. I can imagine useful previous experiences coming from a number of sources (e.g., gaming or playing a musical instrument), but the important take-away is that for users who were able to perform gestures in a repeatable way, the interaction techniques worked quite well. This suggests that for users who were not initially able to perform replicable gestures that they might develop this ability in the future with more practice.

8.5 Final Remarks

Computing has permeated and improved many aspects of our life, yet there are still circumstances in which it is impractical or impossible to use computing devices. My work expands the settings and circumstances in which computing is accessible, thereby making the benefits of computing available to more people in a broader range of situations. The goal of my dissertation work was to create and demonstrate user interface concepts and techniques to extend the abilities of computing in general and mobile computing in particular. I have demonstrated that muscle-computer interfaces are feasible, practical, and hold great promise to augment our current computing experiences and better support our everyday lives. From the surgeon wearing gloves in a sterile surgical theater and trying to review MRI images to the gardener knee deep in manure and trying to answer her cell phone, the new interaction paradigm I have developed opens a new world of possibilities and can create entirely new computing experiences.

My hope is that in 20 years, we will look back and have trouble imagining the time when we advanced slides in a presentation with a handheld clicker, would stop running to mess with the controls of our portable music player, or could not answer phone calls because we were wearing gloves. I believe that novel on-body sensors and actuators, including muscle-computer interfaces, can vastly increase the role computing plays in people's everyday lives.

REFERENCES

Agrawala, Maneesh, Andrew C Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. 1997. The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In *SIGGRAPH 97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 327-332. ACM Press/Addison-Wesley Publishing Co.

Ambient Devices. Ambient Orb. http://www.ambientdevices.com/cat/orb/orborder.html.

Arduino. Arduino. http://www.arduino.cc/.

Ashbrook, Daniel, James Clawson, Kent Lyons, Nirmal Patel, Thad Starner, and Santa Clara. 2008. Quickdraw: the impact of mobility and on-body placement on device access time. In *CHI 08 Proceeding of the twentysixth annual SIGCHI conference on Human factors in computing systems*, 219-222. Florence, Italy, April 05 - 10, 2008: ACM Press.

Ashbrook, Daniel, Kent Lyons, and Thad Starner. 2008. An investigation into round touchscreen wristwatch interaction. In *MobileHCI 08 Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, 311–314. Amsterdam, The Netherlands, September 02 - 05, 2008: ACM Press.

Baber, Chris, James Knight, D Haniff, and L Cooper. 1999. Ergonomics of wearable computers. *Mobile Networks and Applications* 4, no. 1: 15-21.

Bach-Y-Rita, Paul, Carter C Collins, Frank A Saunders, Benjamin White, and Lawrence Scadden. 1969. Vision Substitution by Tactile Image Projection. *Nature* 221, no. 5184: 963-964.

Bach-Y-Rita, Paul, Mitchell E Tyler, and Kurt A Kaczmarek. 2003. Seeing with the Brain. *International Journal of Human-Computer Interaction* 15, no. 2: 285 - 295.

Beaudin, Jennifer S., Stephen S. Intille, Emmanuel M. Tapia, Randy Rockinson, and Margaret E. Morris. 2007. Context-sensitive microlearning of foreign language vocabulary on a mobile device. *Ambient Intelligence* 4794: 55–72.

Benko, Hrvoje, T. Scott Saponas, Dan Morris, and Desney Tan. 2009. Enhancing input on and above the interactive surface with muscle sensing. In *ITS'09 International Conference On Interactive Tabletops And Surfaces*, 93-100. Banff, Alberta, Canada, November 23-25, 2009: ACM Press.

Benko, Hrvoje, and Andrew D Wilson. 2009. DepthTouch: Using Depth-Sensing Camera to Enable Freehand Interactions On and Above the Interactive Surface. *Microsoft Research Technical Report*, no. MSR-TR-2009-23.

Benko, Hrvoje, and Steven Feiner. 2007. Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection. In *2007 IEEE Symposium on 3D User Interfaces*, 7:79–86. IEEE.

Bernstein, Lynne E, Marilyn E Demorest, David C Coulter, and Michael P O'Connell. 1991. Lipreading sentences with vibrotactile vocoders: performance of normal-hearing and hearing-impaired subjects. *The Journal of the Acoustical Society of America* 90, no. 6: 2971-2984.

Betts, Jonathan. 2006. *Time restored: The Harrison Timekeepers and R.T. Gould, the man who knew (almost) everything*. Oxford University Press.

Bilmes, Jeff, Jonathan Malkin, Xiao Li, Susumu Harada, Kelley Kilanski, Katrin Kirchhoff, Richard Wright, et al. 2006. The Vocal Joystick. In *ICASSP 2006 Proceedings. IEEE International Conference on Acoustics, Speech and Signal Processing.*, 1:625-628.

Biosemi. Biosemi EEG ECG EMG BSPM NEURO amplifier electrodes. http://www.biosemi.com.

Bohus, Dan, and Eric Horvitz. 2009. Dialog in the Open World: Platform and Applications. In *Proceedings of the 2009 international conference on Multimodal interfaces*, 31-38. Cambridge, MA, USA, November 2-4, 2009.

Bolt, Richard A. 1980. 'Put-that-there': Voice and Gestures at the Graphics Interface. *SIGGRAPH Computer Graphics* 14, no. 3: 262-270.

Bolt, Richard A. 1985. Conversing with computers. *Technology Review* 88, no. 2: 35-43.

Bretzner, Lars, Ivan Laptev, and Tony Lindeberg. 2002. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*: 423-428.

Brewster, Stephen A, Peter C Wright, and Alistair D N Edwards. 1992. A detailed investigation into the effectiveness of earcons. In *ICAD'92 Proceedings of the First International Conference on Auditory Display*, 471-498. Santa Fe, NM, October 28-30, 1992: Addison-Wesley Publishing Company.

Brown, Lorna M., and Topi Kaaresoja. 2006. Feel who's talking: using tactons for mobile phone alerts. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, 604-609. Montréal, Québec, Canada, April 22 - 27, 2006: ACM.

Burges, Christopher J C. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2, no. 2: 121-167.

Bush, Vannevar. 1945. As We May Think. *The Atlantic Monthly* 176, no. 1: 101-108.

Chang, Angela, Sile O'Modhrain, Rob Jacob, Eric Gunther, and Hiroshi Ishii. 2002. ComTouch: design of a vibrotactile communication device. In *DIS '02 Proceedings of the 4th conference on Designing interactive systems processes practices methods and techniques*, 312-320. London, England, June 25 - 28, 2002: ACM.

Cholewiak, Roger W, and Amy A Collins. 2000. The generation of vibrotactile patterns on a linear array: influences of body site, time, and presentation mode. *Perception psychophysics* 62, no. 6: 1220-1235.

Cholewiak, Roger W, and Amy A Collins. 2003. Vibrotactile localization on the arm: effects of place, space, and age. *Perception psychophysics* 65, no. 7: 1058-1077.

Consolvo, Sunny, David W McDonald, Tammy Toscos, Mike Y Chen, Jon Froehlich, Beverly Harrison, Pedrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. 2008. Activity sensing in the wild: a field trial of UbiFit garden. In *CHI '08 Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, 1797-1806. Florence, Italy, April 05 - 10, 2008: ACM.

Consolvo, Sunny, Peter Roessler, and Brett E Shelton. 2004. The CareNet display: Lessons learned from an in home evaluation of an ambient display. In *UbiComp 2004: 6th International Conference on Ubiquitous Computing*, 3205:1-17. Nottingham, UK, September 7-10, 2004: Springer.

Costanza, Enrico, Samuel A. Inverso, Rebecca Allen, and Pattie Maes. 2007. Intimate interfaces in action: assessing the usability and subtlety of EMG-based motionless gestures. In *CHI '07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 819 - 828. San Jose, California, USA, April 28 - May 03, 2007.

Costanza, Enrico, Samuel A. Inverso, and Rebecca Allen. 2005. Toward subtle intimate interfaces for mobile devices using an EMG controller. In *CHI '05 Proceedings of the SIGCHI conference on Human factors in computing systems*, 481 - 489. Portland, Oregon, USA, April 02 - 07, 2005: ACM Press.

Cutler, Lawrence D, Bernd Frölich, and Pat Hanrahan. 1997. Two-Handed Direct Manipulation on the Responsive Workbench. In *SI3D '97 Proceedings of the 1997* *symposium on Interactive 3D graphics*, 107 - 114. Providence, Rhode Island, United States, April 27 - 30, 1997: ACM Press.

Czernuszenko, Marek, Dave Pape, Daniel Sandin, Tom DeFanti, Gregory L Dawe, and Maxine D Brown. 1997. The ImmersaDesk and Infinity Wall projection-based virtual reality displays. *Computers & Graphics* 31, no. 2: 46-49.

Dempster, Frank N. 1987. Effects of variable encoding and spaced presentations on vocabulary learning. *Journal of Educational Psychology* 79, no. 2: 162-170.

Dietz, Paul, and Darren Leigh. 2001. DiamondTouch: a multi-user touch technology. In *UIST '01 Proceedings of the 14th annual ACM symposium on User interface software and technology*, 219-226. Orlando, Florida, November 11 - 14, 2001: ACM.

Dohrn-Van Rossum, Gerhard. 1996. *History of the hour: Clocks and Modern Temporal Orders*. Chicago: The University of Chicago Press.

Easton, Randolph D. 1992. Inherent problems of attempts to apply sonar and vibrotactile sensory aid technology to the perceptual needs of the blind. *Optometry and vision science official publication of the American Academy of Optometry* 69, no. 1: 3-14.

Englehart, Kevin, and Bernard Hudgins. 2003. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering* 50, no. 7: 848 - 854.

Evans, Rand. 1991. The Just Noticeable Difference: Psychophysical Instrumentation and the Determination of Sensory Thresholds. In *Proceedings of the Eleventh International Scientific Instrument Symposium*, 137-141. Bologna, Italy, September 9-14, 1991.

Farry, Kristin A, and Ian D Walker. 1993. Myoelectric teleoperation of a complex robotic hand. In *1993 IEEE International Conference on Robotics & Automation*, 502-509. Atlanta, GA, May 2-6, 1993.

FeelSpace Belt Project. FeelSpace Belt Project. http://feelspace.cogsci.uos.de/en/index.html.

Forsyth, David A, and Jean Ponce. 2002. *Computer Vision: A Modern Approach*. *Notes*. USA: Prentice Hall.

Froehlich, Jon, Tawanna Dillahunt, Prefrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A Landay. 2009. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. *CHI '09 Proceedings of the 27th international conference on Human factors in computing systems*: 1043–1052.

Gallace, Alberto, Hong Z Tan, and Charles Spence. 2007. The Body Surface as a Communication System: The State of the Art after 50 Years. *Presence: Teleoperators and Virtual Environments* 16, no. 6: 655- 676.

Galán, Ferran, Marnix Nuttin, Eileen Lew, Pierre W Ferrez, G Vanacker, Johan Philips, and Jose Del R Millán. 2008. A brain-actuated wheelchair: asynchronous and non-invasive Brain-computer interfaces for continuous control of robots. *Clinical Neurophysiology* 119, no. 9: 2159-2169.

Gemperle, Francine, Nathan Ota, and Dan Siewiorek. 2001. Design of a wearable tactile display. In *ISWC '01 Proceedings of the Fifth International Symposium on Wearable Computers*, 5-12. Zurich, Switzerland, October 8-9, 2001: IEEE Computer Society.

Girouard, Audrey, Erin Treacy Solovey, Leanne M Hirshfield, Krysta Chauncey, Angelo Sassaroli, Sergio Fantini, and Robert J Jacob. 2009. Distinguishing Difficulty Levels with Non-invasive Brain Activity Measurements. In *Interact 2009 Twelfth IFIP TC13 Conference on Human-Computer Interaction*, 440-452. Uppsala, Sweden, August 24-28, 2009: Springer-Verlag.

Gray, Henry, and Warren H. Lewis. 1918. *Anatomy of the Human Body (20th Edition)*. Lea & Febiger.

Grimes, David, Desney S Tan, Scott E Hudson, Pradeep Shenoy, and Rajesh P N Rao. 2008. Feasibility and pragmatics of classifying working memory load with an electroencephalograph. In *CHI '08 Proceeding of the twentysixth annual CHI conference on Human factors in computing systems*, 835-844. Florence, Italy, April 5-10, 2008: ACM Press.

Haluck, Randy S, and Thomas M Krummel. 2000. Computers and virtual reality for surgical education in the 21st century. *Archives of surgery* 135, no. 7: 786-792.

Han, Jefferson Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05 Proceedings of the 18th annual ACM symposium on User interface software and technology*, 115-118. Seattle, WA, October 23-26, 2005: ACM Press.

Harada, Susumu, Jacob O Wobbrock, and James A Landay. 2007. Voicedraw: a hands-free voice-driven drawing application for people with motor impairments. In *Assets '07 Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, 27-34. Tempe, Arizona, USA, October 15-17, 2007: ACM.

Harada, Susumu, James A Landay, Jonathan Malkin, Xiao Li, and Jeff A Bilmes. 2008. The Vocal Joystick: evaluation of voice-based cursor control techniques for assistive technology. *Disability and rehabilitation Assistive technology* 3, no. 1: 22-34.

Harada, Susumu, T. Scott Saponas, and James A. Landay. 2007. VoicePen: augmenting pen input with simultaneous non-linguisitic vocalization. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, 178-185. Nagoya, Aichi, Japan, November 12 - 15, 2007.

Harrison, Chris, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the Body as an Input Surface. In *CHI '10 Proceedings of the 30th international conference on Human factors in computing systems*, 453-462. Atlanta, GA, USA, April 10-15, 2010.

Harrison, Chris, and Scott E. Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *UIST '09 Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, 121-124. Victoria, BC, Canada, October 4-7, 2009.

Hilliges, Otmar, Shahram Izadi, Andrew D Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. 2009. Interactions in the air: adding further depth to interactive tabletops. In *UIST '09 Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, 139-148. Victoria, BC, Canada, October 4-7, 2009: ACM.

Hirshfield, Leanne M, Erin Treacy Solovey, Audrey Girouard, James Kebinger, Robert J K Jacob, Angelo Sassaroli, and Sergio Fantini. 2009. Brain Measurement for Usability Testing and Adaptive Interfaces: An Example of Uncovering Syntactic Workload with Functional Near Infrared Spectroscopy. In *CHI '2009 Proceedings of the 27th International Conference on Human Factors in Computing Systems*, 2185-2194. Boston, MA, USA, April 4-9, 2009: ACM.

Hoggan, Eve, Sohail Anwar, and Stephen A Brewster. 2007. Mobile multi-actuator tactile displays. In *HAID 2007 Haptic and Audio Interaction Design, Second International Workshop*, 22-33. Seoul, South Korea, November 29-30, 2007: Springer.

Huber, Daniel, Anuj Kapuria, Raghavendra Donamukkala, and Martial Herbert. 2004. Parts-Based 3D Object Classification. In *CVPR'04 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 82-89. Washington DC, USA, June 27-July 2, 2004: IEEE Computer Society.

Hudson, Scott E, Chris Harrison, Beverly L Harrison, and Anthony LaMarca. 2010. Whack gestures. In *TEI '10 Proceedings of the Fourth International conference on Tangible Embedded and Embodied Interaction*, 109-112. Cambridge, Massachusetts, USA, Janurary 24-27, 2010: ACM Press.

Huo, Xueliang, Jia Wang, and Maysam Ghovanloo. 2008. A magneto-inductive sensor based wireless tongue-computer interface. *IEEE Transactions on Neural and Rehabilitation Systems Engineering* 16, no. 5: 497-504.

Iddan, Gavriel J, and Giora Yahav. 2001. 3D imaging in the studio (and elsewhere...). In *SPIE 2001 International Symposium on Microelectronics and Micro-Electro-Mechanical Systems*, 4298:48–55. Adelaide, Australia, December 17-19, 2001.

Izadi, Shahram, Steve Hodges, Stuart Taylor, Dan Rosenfeld, Nicolas Villar, Alex Butler, and Jonathan Westhues. 2008. Going beyond the display: a surface technology with an electronically switchable diffuser. In *UIST '08 Proceedings of the 21st annual ACM symposium on User interface software and technology*, 19:269–278. Monterey, CA, USA, October 19-22, 2008: ACM New York, NY, USA.

Jacobsen, Sephen C, and Robert B Jerard. 1974. Computational requirements for control of the Utah arm. In *ACM '74 Proceedings of the 1974 Annual Conference/Annual Meeting*, 149-155.

Jiang, Mingwen, Rencheng Wang, Jingzhang Wang, and Dewen Jin. 2005. A Method of Recognizing Finger Motion Using Wavelet Transform of Surface EMG Signal. In *27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2672-2674. Shanghai, China, September 1-4, 2005.

Ju, Peter, Leslie Pack Kaelbling, and Yoram Singer. 2000. State-based Classification of Finger Gestures from Electromyographic Signals. In *ICML-2000 Proceedings of the Seventeenth International Conference on Machine Learning*, 439-446. Palo Alto, CA, June 29-July 2, 2000: Morgan Kaufmann.

Kaczmarek, Kurt A, J G Webster, P Bach-Y-Rita, and W J Tompkins. 1991. Electrotactile and vibrotactile displays for sensory substitution systems. *IEEE Transactions on Biomedical Engineering* 38, no. 1: 1-16.

Kamel, Hesham M., and James A. Landay. 1999. The integrated communication 2 draw (IC2D): a drawing program for the visually impaired. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems*, 222-223. Pittsburgh, Pennsylvania, May 15 - 20, 1999: ACM.

Kapoor, Ashish, Pradeep Shenoy, and Desney S. Tan. 2008. Combining brain computer interfaces with vision for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1-8. Anchorage, AK, June 23-28, 2008: IEEE.

Kattinakere, Raghavendra S, Tovi Grossman, and Sriram Subramanian. 2007. Modeling steering within above-the-surface interaction layers. In *CHI '07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 317-326. San Jose, California, USA, April 28-May 3, 2007: ACM Press.

Kiguchi, Kazuo, Takakazu Tanaka, and Toshio Fukuda. 2004. Neuro-Fuzzy Control of a Robotic Exoskeleton With EMG Signals. *IEEE Transactions on Fuzzy Systems* 12, no. 4: 481-490.

Kleissen, R.F.M., J.H. Buurke, J. Harlaar, and G. Zilvold. 1998. Electromyography in the biomechanical analysis of human movement and its clinical application. *Gait posture* 8, no. 2: 143-158.

Lanyi, Xu, and Andy Adler. 2004. An improved method for muscle activation detection during gait. In *CCECE '04 Seventeenth Canadian Conference on Electrical and Computer Engineering*, 357-360. Niagara Falls, Ontario, Canada, May 2-5, 2004.

Lee, Johnny Chung, and Desney S Tan. 2006. Using a low-cost electroencephalograph for task classification in HCI research. In *UIST '06 Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, 81-90. Montreux, Switzerland, October 15-18, 2006: ACM Press.

Leeb, Robert, Doron Friedman, Gernot R Müller-Putz, Reinhold Scherer, Mel Slater, and Gert Pfurtscheller. 2007. Self-Paced (Asynchronous) BCI Control of a

Wheelchair in Virtual Environments: A Case Study with a Tetraplegic. *Computational intelligence and neuroscience* 2007, no. Article ID 79642.

Lenman, Sören, Lars Bretzner, and Björn Thuresson. 2002. Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction. *Royal Institute of Technology, Sweden*, no. TRITA-NA-D0209, CID-172.

Lenman, Sören, Lars Bretzner, and Björn Thuresson. 2002. Using marking menus to develop command sets for computer vision based hand gesture interfaces. In *NordiCHI '02 Proceedings of the Second Nordic Conference on Human-Computer Interaction*, 239-242. Aarhus, Denmark, October 19-23, 2002: ACM Press.

Li, Kevin A., Patrick Baudisch, William G. Griswold, and James D. Hollan. 2008. Tapping and rubbing: exploring new dimensions of tactile feedback with voice coil motors. In *UIST '08 Proceedings of the 21st annual ACM Symposium on User Interface Software and Technology*, 181-190. Monterey, CA, USA, October 19-22, 2008: ACM New York, NY, USA.

Lieberman, Jeff, and Cynthia Breazeal. 2007. TIKL: Development of a Wearable Vibrotactile Feedback Suit for Improved Human Motor Learning. *IEEE Transactions on Robotics* 23, no. 5: 919-926.

Lindeman, Robert W, Yasuyuki Yanagida, Haruo Noma, and Kenichi Hosaka. 2005. Wearable vibrotactile systems for virtual contact and information display. *Virtual Reality* 9, no. 2-3: 203-213.

Lindeman, Robert W., Robert Page, Yasuyuki Yanagida, and John L. Sibert. 2004. Towards full-body haptic feedback: the design and deployment of a spatialized vibrotactile feedback system. In *VRST '04 Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 146-149. Hong Kong, November 10 - 12, 2004.

Lyons, Kent, Daniel Plaisted, and Thad Starner. 2004. Expert Chording Text Entry on the Twiddler One-Handed Keyboard. In *ISWC '04 Proceedings of the 8th*

International Symposium on Wearable Computers, 94-101. Arlington, VA, October 31-November 3, 2004: IEEE.

MacKenzie, Christine L, and Thea Iberall. 1994. *The Grasping Hand*. Amsterdam: North-Holland, Elsevier Science.

Magielse, Remco, and Panos Markopoulos. 2009. HeartBeat: An Outdoor Pervasive Game for Children. In *CHI '09 Proceedings of the 27th international Conference on Human Factors in Computing Systems*, 2181-2184. Boston, MA, USA, April 04 - 09, 2009: ACM Press.

Malik, Shahzad, Abhishek Ranjan, and Ravin Balakrishnan. 2005. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *UIST '05 Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, 43-52. Seattle, WA, October 23-26, 2005: ACM.

Manabe, Hiroyuki. 2004. Evaluations of the ring shaped EMG measurement system. In *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2188-2191. San Francisco, CA, September 1-5, 2004.

Mandryk, Regan, Kori Inkpen, and Thomas Calvert. 2006. Using psychophysiological techniques to measure user experience with entertainment technologies. *Behaviour & Information Technology* 25, no. 2: 141-158.

Mann, Steve. 1998. Wearable Computing as Means for Personal Empowerment. In *ICWC-98 Proceedings of the First International Conference on Wearable Computing*, Keynote Address. Fairfax, VA, May 12-13, 1998. http://wearcam.org/icwckeynote.html.

Merletti, Roberto, and Philip A Parker. 2004. *Electromyography: Physiology, engineering, and noninvasive applications*. New Jersey: Wiley-IEEE Press.

Microsoft. Introducing Kinect for Xbox 360. http://www.xbox.com/kinect.

Microsoft. Microsoft Surface. http://www.microsoft.com/surface.

Microsoft. Windows Speech Recognition. http://www.microsoft.com/enable/products/windowsvista/speech.aspx.

Mistry, Pranav, and Pattie Maes. 2009. SixthSense: a wearable gestural interface. In *SIGGRAPH ASIA '09 Proceedings of International Conference on Computer Graphics and Interactive Techniques*, Article No. 11. Yokohama, Japan, December 16 - 19, 2009: ACM.

Mynatt, Elizabeth D, Jim Rowan, Annie Jacobs, and Sarah Craighill. 2001. Digital family portraits: supporting peace of mind for extended family members. In *CHI '01 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 333-340. Seattle, WA, USA, March 31-April 5, 2001: ACM Press.

Naik, Ganesh R, Dinesh Kant Kumar, Vijay Pal Singh, and Marimuthu Palaniswami. 2006. Hand gestures for HCI using ICA of EMG. In *VisHCI 2006 Proceedings of the Hcsnet Workshop on Use of Vision in Human-Computer interaction*, 67-72. Canberra, Australia, November 01 - 01, 2006.

Nenonen, Ville, Aleksi Lindblad, Ville Häkkinen, Toni Laitinen, Mikko Jouhtio, and Perttu Hämäläinen. 2007. Using heart rate to control an interactive game. In *CHI '07 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 853-856. San Jose, California, USA, April 28 - May 03, 2007: ACM Press.

New Abilities Systems. New Abilities Systems. http://www.newabilities.com/.

Nijholt, Anton, Desney Tan, Gert Pfurtscheller, Clemens Brunner, Jos Mill, Brendan Allison, Bernhard Graimann, Florin Popescu, Benjamin Blankertz, and Klaus-R M. 2008. Brain-Computer Interfacing for Intelligent Systems. *IEEE Intelligent Systems* 23, no. 3: 72-79.

Nuance Communication. Dragon NaturallySpeaking Solutions. http://www.nuance.com/naturallyspeaking/.

Obermaier, Bernhard, Gernot R Müller, and Gert Pfurtscheller. 2003. "Virtual keyboard" controlled by spontaneous EEG activity. *IEEE Transactions on Neural and Rehabilitation Systems Engineering* 11, no. 4: 422-426.

Oliver, Nuria, and Fernando Flores-Mangas. 2006. MPTrain: a mobile, music and physiology-based personal trainer. In *MobileHCI '06 Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services*, 21-28. Helsinki, Finland, September 12 - 15, 2006.

Oviatt, Sharon, Phil Cohen, Lizhong Wu, J Vergo, L Duncan, B Suhm, J Bers, et al. 2000. Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions. *Human-Computer Interaction* 15, no. 4: 263-322.

Patek Philippe Museum. The 1st Patek Philippe Wristwatch. http://www.patekmuseum.com/pop/popWatch.asp?ID=41&themeId=7&floorId=First.

Peleg, Dori, Eyal Braiman, Elad Yom-Tov, and Gideon F Inbar. 2002. Classification of finger activation for use in a robotic prosthesis arm. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 10, no. 4: 290-293.

Peng, Qiyu, and Thomas F Budinger. 2007. ZigBee-based wireless intra-oral control system for quadriplegic patients. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1647-1650. Lyon, France, August 23-26, 2007: IEEE.

Piateski, Erin, and Lynette Jones. 2005. Vibrotactile Pattern Recognition on the Arm and Torso. In *WHC '05 First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 90-95. Pisa, Italy, March 18-20, 2005: IEEE.

Platt, John. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. *Microsoft Research Technical Report* TR-98-14.

PrimeSense. The PrimeSensor Technology. http://www.primesense.com/?p=486.

Raez, M B I, M S Hussain, and F Mohd-Yasin. 2006. Techniques of EMG signal analysis: detection, processing, classification and applications. *Biological Procedures Online* 8, no. 1: 11-35.

Rehg, James M, and Takeo Kanade. 1994. DigitEyes: vision-based hand tracking for human-computer interaction. In *Proceedings of 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 16-22. Austin, TX, USA, November 11-12, 1994: IEEE.

Ren, Xiaofeng, and Matthai Philipose. 2009. Egocentric recognition of handled objects: Benchmark and analysis. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1-8. Miami, FL, June 20-25, 2009: IEEE.

Rowley, Henry A, Shumeet Baluja, and Takeo Kanade. 1998. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, no. 1: 23-38.

Saeedi, Ehsan, Samuel Kim, and Babak A. Parviz. 2008. Self-assembled crystalline semiconductor optoelectronics on glass and plastic. *Journal of Micromechanics and Microengineering* 18, no. 7: 1-7.

Salem, Chris, and Shumin Zhai. 1997. An isometric tongue pointing device. In *CHI '97 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 538-539. Atlanta, Georgia, United States, March 22 - 27, 1997: ACM Press. doi:10.1145/258549.259021.

Saponas, T Scott, Desney S. Tan, Dan Morris, and Ravin Balakrishnan. 2008. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. In *CHI '08 Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, 515-524. Florence, Italy, April 5-10, 2008.

Saponas, T. Scott, Daniel Kelly, Babak A. Parviz, and Desney S. Tan. 2009. Optically sensing tongue gestures for computer input. In *UIST '09 Proceedings of the 22nd Annual ACM Symposium on User interface Software and Technology*, 177-180. Victoria, BC, Canada, October 04 - 07, 2009.

Saponas, T. Scott, Desney S. Tan, Dan Morris, Jim Turner, and James A. Landay. 2010. Making muscle-computer interfaces more practical. In *CHI '10 Proceedings of the 28th international Conference on Human Factors in Computing Systems*, 851-854. Atlanta, Georgia, USA, April 10-15, 2010.

Saponas, T. Scott, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling always-available input with muscle-computer interfaces. In *UIST '09 Proceedings of the 22nd Annual ACM Symposium on User interface Software and Technology*, 167-176. Victoria, BC, Canada, October 4-7, 2009.

Sassaroli, Angelo, Feng Zheng, Leanne M Hirshfield, Audrey Girouard, Erin Treacy Solovey, Robert J K Jacob, and Sergio Fantini. 2008. Discrimination of mental workload levels in human subjects with functional near-infrared spectroscopy. *Journal of Innovative Optical Health Sciences* 1, no. 2: 227-237.

Scherer, Reinhold, Gernot R Müller, Christa Neuper, Bernhard Graimann, and Gert Pfurtscheller. 2004. An asynchronously controlled EEG-based virtual keyboard: improvement of the spelling rate. *IEEE Transactions on Biomedical Engineering* 51, no. 6: 979-984.

Schieber, Marc H. 1995. Muscular production of individuated finger movements: the roles of extrinsic finger muscles. *The Journal of Neuroscience: the Official Journal of the Society for Neuroscience* 15, no. 1 Pt 1: 284-297.

Schlesinger, G. 1919. *Der Mechanische Aufbau der Kunstlichen Glieder. Ersatzglieder und Arbeitshilfen, part II*. Berlin: Springer Verlag.

Schneiderman, Henry, and Takeo Kanade. 2000. A statistical method for 3D object detection applied to faces and cars. In *CVPR '00 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 746-751. Hilton Head, South Carolina, June 13-15, 2000: IEEE.

Schuon, Sebastian, Christian Theobalt, James Davis, and Sebastian Thrun. 2008. High-quality scanning using time-offlight depth superresolution. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1-7. Anchorage, Alaska, June 23, 28, 2008: IEEE.

Smith, Raymond Carl. 2004. *Electroencephalograph based Brain Computer Interfaces*. Masters Thesis, Electrical and Electronic Engineering, University College Dublin.

Solovey, Erin Treacy, Audrey Girouard, Krysta Chauncey, Leanne M Hirshfield, Angelo Sassaroli, Feng Zheng, Sergio Fantini, and Robert J K Jacob. 2009. Using fNIRS brain sensing in realistic HCI settings: experiments and guidelines. In *UIST '09 Proceedings of the 22nd Annual ACM Symposium on User interface Software and Technology*, 157-166. Victoria, BC, Canada, October 04 - 07, 2009: ACM.

Spitzer, Mark Bradley, Noa Rensing, Robert William Mcclelland, and Paul Daniel Aquilino. 1997. Eyeglass-Based Systems For Wearable Computing. In *ISWC '97 Proceedings of the First IEEE International Symposium on Wearable Computers*, 48-51. Cambridge, MA, USA, October 13-14, 1997: IEEE.

Sporka, Adam J., Susumu Harada, and Sri H. Kurniawan. 2007. Striking a c[h]ord: vocal interaction in assistive technologies, games, and more. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, 2869 - 2872. San Jose, CA, USA, April 28 - May 03, 2007.

Starner, Thad, Jake Auxier, Daniel Ashbrook, and Maribeth Gandy. 2000. The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring. In *ISWC '00 Proceedings of the 4th*

IEEE International Symposium on Wearable Computers, 87-94. Atlanta, GA, USA, October 16-17, 2000: IEEE.

Struijk, Lotte N S Andreasen. 2006. An inductive tongue computer interface for control of computers and assistive devices. *IEEE Transactions on Biomedical Engineering* 53, no. 12 Pt 2: 2594-2597.

Sugiura, Atsushi, and Yoshiyuki Koseki. 1998. A user interface using fingerprint recognition. In *UIST '98 Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, 71-79. San Francisco, CA, USA, November 1-4, 1998: ACM Press.

Tan, Desney S., and Anton Nijholt. 2010. *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction*. London: Springer.

Tang, Hui, and David J. Beebe. 1998. A microfabricated electrostatic haptic display for persons with visual impairments. *IEEE Transactions on Rehabilitation Engineering* 6, no. 3: 241-248.

Tenore, Francesco, Ander Ramos, Amir Fahmy, Soumyadipta Acharya, Ralph Etienne-Cummings, and Nitish V Thakor. 2007. Towards the control of individual fingers of a prosthetic hand using surface EMG signals. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 6146-6149. Lyon, France, August 23-26, 2007.

Toney, Aaron, Lucy Dunne, Bruce H Thomas, and Susan P Ashdown. 2003. A shoulder pad insert vibrotactile display. In *ISWC '03 Proceedings of the 7th International Symposium on Wearable Computers*, 35–44. White Plains, NY, USA, October 21-23, 2003: IEEE.

Torrey, Lisa, Trevor Walker, W. Shavlik, Jude, and Richard Maclin. 2006. Skill Acquisition via Transfer Learning and Advice Taking. In *ECML '06 Proceedings of the*

Seventeenth European Conference on Machine Learning, 425-436. Berlin, Germany, September 18-22, 2006: Springer Berlin.

Tsukada, Koji, and Michiaki Yasumura. 2004. Activebelt: Belt-type wearable tactile display for directional navigation. In *UbiComp 2004: 6th International Conference on Ubiquitous Computing*, 384–399. Tokyo, Japan, September 11-14, 2005: Springer.

Van Erp, Jan B F, Herdrik A H C Van Veen, Chris Jansen, and Trevor Dobbins. 2005. Waypoint navigation with a vibrotactile waist belt. *ACM Transactions on Applied Perception (TAP)* 2, no. 2: 106-117.

Van Erp, Jan B F, and Hendrik A H C Van Veen. 2004. Vibrotactile in-vehicle navigation system. *Transportation Research Part F: Traffic Psychology and Behaviour* 7, no. 4-5: 247-256.

Viola, Paul, and Michael J Jones. 2002. Robust real-time object detection. *International Journal of Computer Vision* 57, no. 2: 137–154.

Wall, Steven A., and William Harwin. 2001. A high bandwidth interface for haptic human computer interaction. *Mechatronics* 11, no. 4: 371-387.

Wang, Gang, Zhizhong Wang, Weiting Chen, and Jun Zhuang. 2006. Classification of surface EMG signals using optimal wavelet packet method based on Davies-Bouldin criterion. *Medical & Biological Engineering & Computing* 44, no. 10: 865-72.

Weiser, Mark. 1991. The Computer for the 21st Century. *Scientific American* 265, no. 3: 94-104.

Wheeler, Kevin R, Mindy H Chang, and Kevin H Knuth. 2006. Gesture-based control and EMG decomposition. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews* 36, no. 4: 503-514.

Wheeler, Kevin R, and Charles C Jorgensen. 2003. Gestures as input: neuroelectric joysticks and keyboards. *IEEE Pervasive Computing* 2, no. 2: 56-61.

Wilson, Andrew D. 2005. PlayAnywhere: a compact interactive tabletop projection-vision system. In *UIST '05 Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology*, 83 - 92. Seattle, WA, USA, October 23 - 26, 2005.

Wilson, Andrew D. 2004. TouchLight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04 Proceedings of the 6th international Conference on Multimodal interfaces*, 69–76. State College, PA, USA, October 13-15, 2004: ACM.

Wilson, Andrew D. 2006. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *UIST '06 Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, 255-258. Montreux, Switzerland, October 15 - 18, 2006: ACM.

Wilson, Andrew D. 2007. Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction. In *TABLETOP '07 Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, 201-204. Newport, RI, USA, October 10-12, 2007: IEEE.

Witten, Ian H, and E Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd. San Francisco: Morgan Kaufmann.

Yatsenko, Dimitri, Daniel McDonnall, and K Shane Guillory. 2007. Simultaneous, proportional, multi-axis prosthesis control using multichannel surface EMG. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 6134-6137. Lyon, France, August 23-26, 2007.

Yeow, John T.W., and Bob Cheung. 2005. An integrated tactile and visual display motion simulator for air and land application. In *ICMA 2005 IEEE International*

Conference on Mechatronics and Automation, 1:298 - 302. Niagara Falls, Ontario, Canada, July 29-August 1, 2005: IEEE.

Zhang, Cha, Yong Rui, and One Microsoft Way. 2006. Robust Visual Tracking via Pixel Classification and Integration. In *ICPR 2006 18th International Conference on Pattern Recognition*, 37-42. Hong Kong, August 20-24, 2006: IEEE.

Zhao, QiBin, LiQing Zhang, and Andrzej Cichocki. 2009. EEG-based asynchronous BCI control of a car in 3D virtual reality environments. *Chinese Science Bulletin* 54, no. 1: 78-87.

VITA

T. Scott Saponas earned his Bachelor of Science degree in Computer Science from the Georgia Institute of Technology in 2004 where he was advised by Professor Gregory D. Abowd. From 2004, he attended the University of Washington, where he was advised by Professor James A. Landay and Dr. Desney Tan. In 2006, he earned his Masters of Science in Computer Science and Engineering. In 2010, he graduated with a Doctor of Philosophy in Computer Science and Engineering.

Appendix A.
Questionnaire from Muscle-Computer Interfaces Feasibility Experiment

See Section 3.3 for the experiment details. The following page contains the demographic questionnaire and summary data from our participants. We had thirteen participants in this experiment.

Gender:

□ Male	□ Female	□ Other
5	**8**	**0**

Age:

□ 18-25	□ 16-35	□ 36-45	□ 46-55	□ 56+
2	**2**	**0**	**5**	**4**

Approximate Height:

____ feet ____ inches

5’9”, 5’3”, 5’7”, 5’6”, 6’0”, 5’1”, 6’1”, 5’8”, 5’11”, 5’7”, 5’11”, 5’3”, 6’8”

Approximate Weight:

□ <100 lbs	□ 101-125 lbs	□ 126-150 lbs	□ 151-175 lbs	□ 176-200 lbs	□ 201-225 lbs	□ 226+ lbs
0	**0**	**2**	**4**	**2**	**3**	**2**

How often do you use a computer?

□ Never	□ A few times per year	□ A few times per month	□ A few times per week	□ Daily
0	**0**	**0**	**0**	**13**

How often do you play video games?

□ Never	□ A few times per year	□ A few times per month	□ A few times per week	□ Daily
0	**7**	**2**	**3**	**1**

Appendix B.
Questionnaire from Real-Time, In-Air Gesture Classification Experiment

See Section 4.4 for the experiment details. The following page contains the demographic questionnaire and summary data from our participants. We had twelve participants in this experiment.

Gender:

□ Male □ Female □ Other

7 **5** **0**

Age: ________

All 12 were in the range of 18 to 55 with an average age of 36.

How often do you use a computer?

□ Never	□ A few times per year	□ A few times per month	□ A few times per week
0	**0**	**0**	**0**

□ Daily

12

Appendix C.
Questionnaire from Cross-Session Classification Experiment

See Section 5.2 for the experiment details. The following page contains the demographic questionnaire and summary data from our participants. We had eight participants in this experiment.

Gender: M F

4 male, 4 female

Age: _____

23, 24, 25, 26, 26, 27, 28, 31

I am: right-handed / left-handed

7 right-handed, 1 left-handed

Do you have any known neuromuscular disease: No / Yes ________________________________

8 No, 0 Yes

Appendix D.
Questionnaire from Surfaces with Muscle-Computer Input Experiment

See Section 6.2.3 for the experiment details. The following page contains the demographic questionnaire and summary data from our participants. We had six participants in this experiment.

Gender: M F

3 male, 3 female

Age: _____

All participants were within the range of 20 to 35 years of age.

I am: right-handed / left-handed

All 6 were right handed.