Enabling Always-Available Input with Muscle-Computer Interfaces

T. Scott Saponas[1], Desney S. Tan[2], Dan Morris[2], Ravin Balakrishnan[4], Jim Turner[3], James A. Landay[1]

[1]Compuer Science and Engineering
DUB Group
University of Washington
{ssaponas, landay}@cs.washington.edu

[2]Microsoft Research
{desney, dan}@microsoft.com

[3]Microsoft Corporation
jturner@microsoft.com

[4]Department of
Computer Science
University of Toronto
ravin@dgp.toronto.edu

ABSTRACT
Previous work has demonstrated the viability of applying offline analysis to interpret forearm electromyography (EMG) and classify finger gestures on a physical surface. We extend those results to bring us closer to using muscle-computer interfaces for always-available input in real-world applications. We leverage existing taxonomies of natural human grips to develop a gesture set covering interaction in free space even when hands are busy with other objects. We present a system that classifies these gestures in real-time and we introduce a bi-manual paradigm that enables use in interactive systems. We report experimental results demonstrating four-finger classification accuracies averaging 79% for pinching, 85% while holding a travel mug, and 88% when carrying a weighted bag. We further show generalizability across different arm postures and explore the tradeoffs of providing real-time visual feedback.

ACM Classification: H.1.2 [User/Machine Systems]; H.5.2 [User Interfaces]: Input devices and strategies; B.4.2 [Input/Output Devices]: Channels and controllers

General terms: Design, Human Factors

Keywords: Electromyography (EMG), Muscle-Computer Interface, input, interaction.

INTRODUCTION
Our hands and our ability to control them have evolved over thousands of years, yielding an amazing ability to precisely manipulate tools. As such, we have often crafted our environments and technologies to take advantage of this ability. For example, many current computer interfaces require manipulating physical devices such as keyboards, mice, and joysticks. Even future looking research systems often focus on physical input devices [5]. However, as computing environments become more diverse, we often find ourselves in scenarios where we either cannot, or prefer not to, explicitly interact with a physical device in hand.

UIST'09, October 4–7, 2009, Victoria, British Columbia, Canada.

Previous work has explored hands-free and implement-free input techniques based on a variety of sensing modalities. For example, computer vision enables machines to recognize faces, track movement and gestures, and reconstruct 3D scenes [24]. Similarly, speech recognition allows for hands-free interaction, enabling a variety of speech-based desktop and mobile applications [8, 11]. However, these technologies have several inherent limitations. First, they require observable interactions that can be inconvenient or socially awkward. Second, they are relatively sensitive to environmental factors such as light and noise. Third, in the case of computer vision, sensors that visually sense the environment are often susceptible to occlusion.

We assert that computer input systems can leverage the full bandwidth of finger and hand gestures without requiring the user to manipulate a physical transducer. In this paper, we show how forearm electromyography (EMG) can be used to detect and decode human muscular movement in real time, thus enabling interactive finger gesture interaction. We envision that such sensing can eventually be achieved with an unobtrusive wireless forearm EMG band (see Figure 1).

Previous work exploring muscle-sensing for input has primarily focused either on using a single large muscle (rather than the fingers) [2, 3, 4, 22, 25], 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 [3, 4, 15, 21, 23, 25], which is not a realistic usage scenario for always-available input devices. Saponas et al. [18] demonstrated the feasibility of using offline machine learning techniques to interpret forearm muscle-sensing and classify finger gestures on a surface. We extend their offline classification results to achieve online classifi-

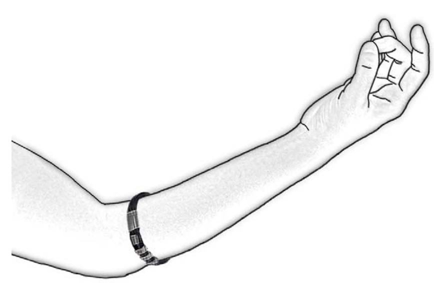

Figure 1. Artist rendering of a forearm muscle-sensing band.

cation that enables using muscle-sensing for always-available input in real-world applications that are not constrained to a surface. Note that our contribution is not in the realm of trying to better understand or measure the physiology of human musculature, but rather in simply sensing muscle activity to enable interaction. Specifically:

1. 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.
2. 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 bimanual "select and activate" paradigm that enables use in interactive systems.
3. 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 further suggest generalizability across different arm postures. Furthermore, we show preliminary evidence of use within a more ecologically valid example application: controlling a simulated portable music player.

We conclude the paper with discussion of our results, the limitations of our techniques, implications for design, and proposals for future work.

BACKGROUND AND RELATED WORK

Sensing Muscles with EMG

Humans employ a complex set of skeletal muscles and adjoining tendons and bones to create body movement. The brain initiates movement process by transmitting an electrical signal through the nervous system. This signal stimulates the fibers that make up our muscles, which contract in response to create forces or body movement.

EMG senses this muscular activity by measuring the electrical potential between pairs of electrodes. This can either be done invasively (with needles in the muscle) or from the surface of the skin. While invasive EMG can be very accurate, our work focuses on surface EMG because it is more practical for HCI applications. For more detailed information on electromyography, see Merletti et al. [13].

For either modality (surface or invasive), the EMG signal is an oscillating electrical wave. When a muscle is contracted, the amplitude of this wave increases, with most of the power in the frequency range of 5 to 250 Hz [13].

Applications of EMG Sensing

EMG is frequently used in clinical settings for *muscle function assessment* during rehabilitation and for *measuring muscle activation* to assess gait [9]. In clinical applications, a typical statistic computed over the EMG signal is the root mean squared (RMS) amplitude of the measured potential. This provides a rough metric for how active a muscle is at a given point in time. For a review of processing techniques used in previous work, see [14].

EMG is also used in both research and clinical settings for *controlling prosthetics*. This typically involves sensing the activity in large individual muscles and using it as input to control the movement of physical devices. For example, the shoulder muscle might be used to control one of the degrees of freedom in a lower-arm prosthetic. Other work has explored similar techniques for sensing activity in large muscles such as the biceps or pectoralis for *computer input* by healthy individuals (e.g. [2]). However, learning to perform fine tasks with muscles that are not normally used for dexterous manipulation can be difficult.

Recent work has used surface EMG to sense and decipher muscle activity that drives fine motor function in our fingers, wrists, and hands. Wheeler et al. [23] explore EMG-based input systems, but assume that the hands are in a constrained, static posture, and do not address calibration issues associated with real-world use. Ju et al. [6] explored several machine learning approaches to classifying a finger-pinch gesture using electrodes placed near participants' wrists, and achieved classification accuracies as high as 78% when differentiating among four gestures. Their work, however, was focused on machine learning techniques, and does not address the human-computer interaction issues that impact the feasibility of real-world EMG applications. In particular, their work does not address posture-independence (e.g., arm rotation), hands-busy scenarios, scenarios in which hands are not constrained to a surface, the "Midas Touch" problem (differentiating intended gestures from rest), or real-time classification. Our work builds on the work of Ju et al. by addressing each of these issues.

Saponas et al. [18] used 10 EMG sensors worn in a narrow band around the upper forearm to differentiate position, pressure, tapping, and lifting gestures across five fingers placed on a surface. They showed the effectiveness of using not only RMS amplitude but also frequency energy and phase coherence features in a linear classifier to attain compelling proof-of-concept results. However, their work was limited in that participants were constrained to fixed arm postures while sitting in a chair and working on a physical surface. Furthermore their data was processed using offline analysis, which did not allow exploration of real-time interactions or the potential effects of feedback to the user.

We seek to extend previous muscle-sensing work to explore real-time classification of finger-level movement for more naturalistic settings including when people are holding objects. We also investigate practical concerns including arm posture independence, "Midas touch," and visual feedback.

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 [12]. Schlesinger [20] put forth most well-known of these taxonomies (see Figure 2), characterizing six different manual grasps:

- **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.

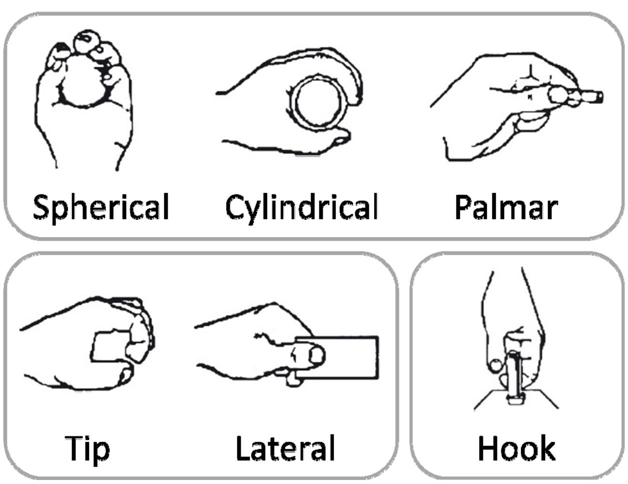

Figure 2. Schlesinger's natural grip taxonomies [20] as depicted in MacKenzie and Iberall [12]. Groupings indicate the three similarity classes that guide our gesture set.

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.

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 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.

Equipment and Setup

We used a *BioSemi Active Two* system as our forearm EMG sensing device (www.biosemi.com). 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. The preparation, gel and adhesive are artifacts of our EMG setup and could be eliminated if dry electrodes such as the Dri-Stik (NeuroDyne Medical, Corp.) are used. This would clearly be more appropriate for real-world use.

To get the best possible signal, EMG sensing is traditionally conducted with two sensors spread an inch apart on a muscle belly. However, Saponas et al. [18] showed that they were able 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 1. Setup took about 15 minutes.

Design and Procedure

We divided the experiment into three parts. Part A examined gestures when the participant's hand was free of objects and explored the sensitivity of our techniques to arm posture. Part B examined gestures while the hands were busy holding objects, a travel mug and a weighted bag that created constant muscular load. In Part C, participants used the muscle-computer interface (while holding an object) to control a simulated portable music player.

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.

Parts A and B of the experiment each contained a training phase, in which the system prompted the participant to perform finger gestures while it collected training data. This data was immediately used to train our gesture recognizer and build a predictive model. The training phase was followed by a testing phase in which the system attempted to classify the participant's gestures in real-time. Part C used the training data collected in Part B for real-time control.

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 contra-lateral hand, in this case the non-dominant hand. Squeezing is a large multi-muscle action that can be robustly detected with consistent timing, but in itself is not sufficiently complex for most applications. By combining rich gestures performed with one hand and robust but simple gestures performed with the other hand, we allow 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 our experiment.

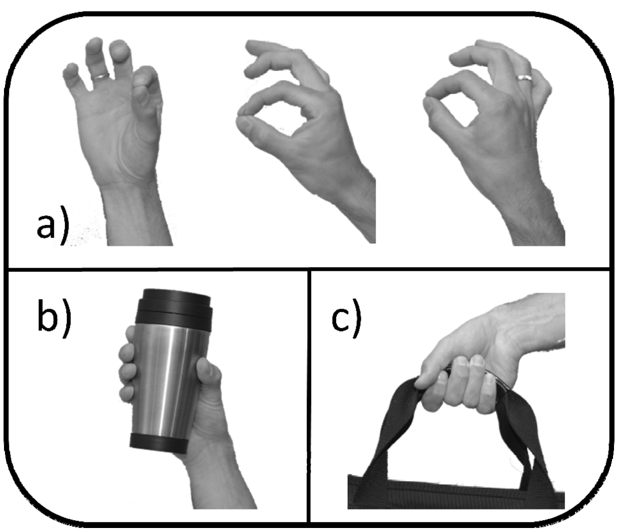

Figure 3. Our finger gesture sets. a) pinch gestures performed in three different arm postures b) fingers squeezing a travel mug c) fingers pulling up against the handle of a carried bag

Part A: Hands-Free Finger Gestures

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 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 1 and Figure 3a).

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 and 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.

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 3a).

Hands-Free 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 4a). We asked 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.

Hands-Free Testing Phase

In the testing phase, participants performed 25 blocks of gestures in each of the three arm postures. As in the training

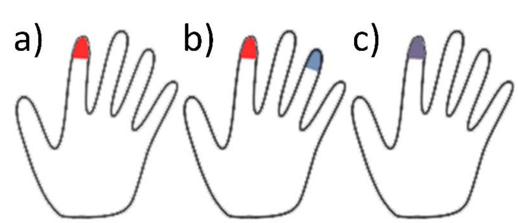

Figure 4. (a) A red highlight indicates that a gesture should be performed with the given finger; (b) a blue highlight indicates the currently recognized gesture; (c) a purple highlight indicates that the correct gesture is being performed.

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 gesture with their right hand and "lock it in" by clenching 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 Part B of the experiment. 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.

Part B: Hands-Busy Finger Gestures

The second part of our experiment explored performing finger gestures when the hands are already busy holding an object. We looked at two different classes of objects. 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 3b). Second, we tested larger and heavier objects being carried. Participants stood in front of the desk and carried a laptop bag in each hand (see Figure 3c). Each bag held a book weighing approximately one kilogram.

As in Part A, 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.

Hands-Busy 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.

Hands-Busy 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 Part A. However, unlike in Part A, 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 Part A's 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.

The "with visual feedback" condition 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 4b-c). We explained to participants that this was the system's best guess at their current gesture. Users were asked 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.

Part C: Controlling a Portable Music Player Application

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

Our simulated portable music player (see Figure 5) 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 from Part B 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 experiments, the left-hand grasping gesture was used to activate the gesture be-

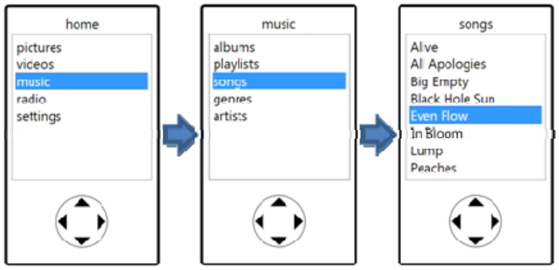

Figure 5. Software mockup of a portable music player

ing 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 he 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 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. Above the music player participants were given task instructions such as "Select Even Flow." 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.

Data Processing Technique

To classify gestures from an EMG signal, we used a similar approach to Saponas et al. [18], performing basic signal processing, computing a set of features, using those features to train a support vector machine (SVM) [1], and then using that SVM to classify finger gestures. While Saponas, et al. did not test this, we show here that this can be used in a real-time system. We outline the procedure here, but more details on the approach can be found in their paper [18].

Basic Signal Processing

Our first step is to convert the raw EMG data into a form suitable for our machine learning algorithm. 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.

Feature Generation

For each 31ms sample, we generated 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 channel, which correlates with magnitude of muscle activity. From the six base RMS features generated by sensors on the right 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 right arm. This yields 78 frequency energy features per 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 right arm.

These calculations result in 114 features per sample for right-hand gesture classification. The only feature we use for left-hand "squeeze" recognition is a single RMS feature computed over the subtractions of the two channels available on the left hand.

Classification of Right-Hand Finger Gestures

Support vector machines (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 [16].

In supervised machine learning, training data inherently needs to be labeled with a 'ground truth'. In our case, this is the gesture being performed by a participant at a given time when the muscle-sensing data segment was gathered. Because people respond to a stimulus with varying delay, there is some amount of mislabeled information early within each stimulus presentation. 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. We chose this approach to reduce sensitivity to momentary fluctuations in classification. Throughout this paper, our classifiers were trained and tested independently on data from each participant during a single participant session.

Classification of Left-Hand Squeeze

Detecting the squeezing gesture performed by the left hand is much simpler. We take the RMS features from the difference of the two channels on the left arm. This process removes noise such as a person's cardiac electrical activity, giving a good estimate of the total muscle activity in the upper forearm. The system took any value above 40% of the maximum value seen during calibration to mean that the left 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 left-hand squeeze. We enforced this "silent" period to prevent unintentional rapid sequences of selections.

Results

In both parts of our experiment, 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 Parts A and B 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.

Part A: Hands-Free Finger Gesture Recognition

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.

As shown in Table 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 (in amount of arm rotation) between training and testing postures. 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 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%.

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

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

Part B: 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 7). 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.

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 7). 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

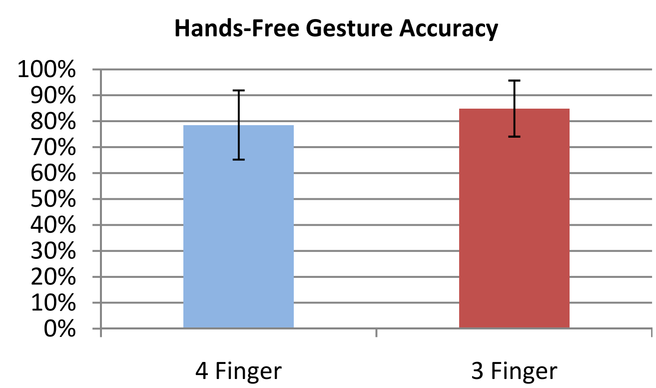

Figure 6. Mean classification accuracies for pinch gesture. Error bars represent standard deviation in all graphs.

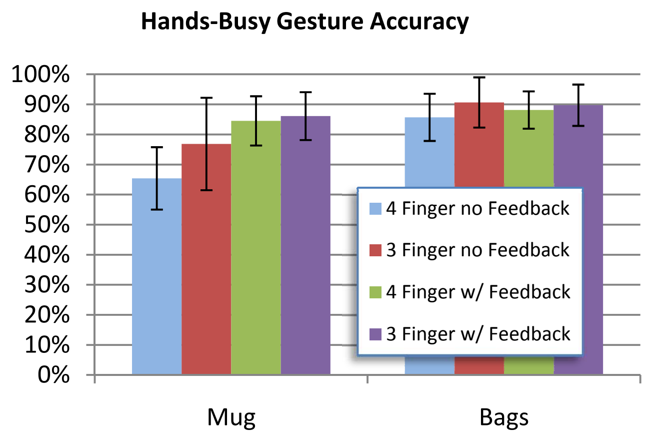

Figure 7. Mean classification accuracies of hands-busy gestures. Error bars represent the standard deviation.

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.

Part C: 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) through 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.

DISCUSSION

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.

Posture Independence

The results from Part A 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.

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 Part B 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.

Quantity of Training Data and Classification Accuracy

Figure 8 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.

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 [10, 19]. 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.

Interaction Design Issues

Even if a system can recognize individual gestures with reasonable accuracy, deployment in real-world scenarios

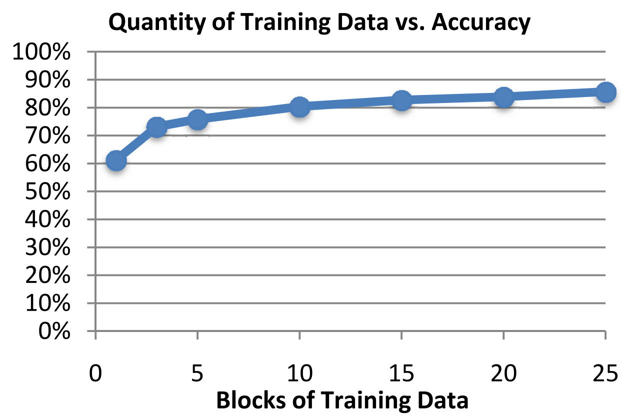

Figure 9. Classification accuracy versus blocks of training data for four finger gestures with bags in hand. Each training block takes seven seconds for a four finger classifier.

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 Part B 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.

Ongoing and Future Directions

Air-Guitar Hero

Encouraged by the results, we developed an application that allows a user to use our muscle-computer interface to play the Guitar Hero game. In Guitar Hero, users hold a guitar-like controller and press buttons using both hands as the system presents stimuli timed to popular music. Using our muscle-computer interface, users can now play with an "air-guitar". A user controls four buttons with our pinching gestures and moves the opposite wrist in a strumming motion. Informal tests of the system show that users are able to complete the easy mode of the game. We demonstrate this system in our video figure.

Wireless Electromyography

Although we extended previous work by not tethering people's arms and hands to specific orientations or surfaces, our experiment was conducted in a lab using a wired electromyography device, and we have yet to validate our classification approaches in scenarios with more variable gesture execution. To this end, we have recently created a small, low-power wireless prototype muscle-sensing unit (see Figure 9). Each of these units is equipped with four electrodes (two differential electromyography channels) sampling at 128 Hz, and multiple units can be used simultaneously. We are currently working to put this wireless unit into an armband form factor with dry electrodes.

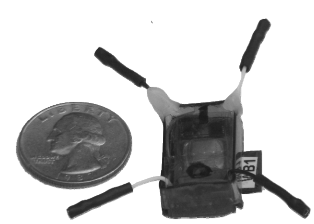

Figure 8. Our wireless EMG device prototype, weighing five grams and measuring 26x18x8mm.

CONCLUSION

Our work 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, in contrast to previous work that relied on off-line analysis. 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.

ACKNOWLEDGEMENTS

We thank Jonathan Lester and Carl Ringler for assistance with the wireless device, Meredith Skeels for fabricating some armbands, as well as John Platt and Jay Stokes for machine learning advice and tools.

REFERENCES

1. Burges, C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121-167.
2. Costanza, E., Inverso, S.A., Allen, R., & Maes, P. 2007. Intimate interfaces in action: Assessing the usability and subtlety of EMG-based motionless gestures. *CHI '07*, 819-828.
3. Englehart, K. & Hudgins, B. 2003. A robust, real time control scheme for multifunction myoelectric control. *IEEE Trans Biomedical Engineering*, 50(7), 848-854.
4. Farry K., Walker I. & Baraniuk R. G. 1996. Myoelectric teleoperation of a complex robotic hand. *Proc IEEE Int Conf Robot Autom*, 775-788.
5. Greenberg, S. & Fitchett, C. 2001. Phidgets: easy development of physical interfaces through physical widgets. *UIST '01*, 209-218.
6. Ju, P., Kaelbling, L. P. & Singer, Y. 2000. State-based Classification of Finger Gestures from Electromyographic Signals. *ICML '08,* 439-446.
7. Kiguchi, K., Tanaka, T. & Fukuda, T. 2004. Neuro-fuzzy control of a robotic exoskeleton with EMG signals. *IEEE Trans. on Fuzzy Systems, 12(4),* 481-490.
8. Lakshmipathy, V., Schmandt, C., and Marmasse, N. 2003. TalkBack: a conversational answering machine. *UIST '03*.
9. Lanyi, X. & Adler, A. 2004. An improved method for muscle activation detection during gait. *Canadian Conf. on Electrical and Computer Engineering*, 357-360.
10. Lee, J.C. & Tan, D.S. 2006. Using a low-cost encephalograph for task classification in HCI research. *UIST '06,* 81-90.
11. Lyons, K., Skeels, C., Starner, T., Snoeck, C. M., Wong, B. A. & Ashbrook, D. 2004. Augmenting conversations using dual-purpose speech. *UIST '04*, 237-246.
12. MacKenzie, C. L. and Iberall, T. 1994. The Grasping Hand. Amsterdam: North-Holland, Elsevier Science.
13. Merletti, R., & Parker, P.A. 2004. *Electromyography: Physiology, engineering, and noninvasive applications.* John Wiley & Sons: Hoboken, New Jersey.
14. Naik, G.R., Kumar, D.K., Singh, V.P. & Palaniswami, M. 2006. Hand gestures for HCI using ICA of EMG. *HCSNet Workshop on the Use of Vision in HCI,* 67-72.
15. Peleg, D., Braiman, E., Yom-Tov, E. & Inbar G.F. 2002. Classification of Finger Activation for Use in a Robotic Prosthesis Arm. *Trans Neural Syst Rehabil Eng*, 10(4).
16. Platt, J. 1998. Sequential Minimal Optimization: A fast algorithm for training support vector machines. *Microsoft Research Technical Report MSR-TR-98-14*.
17. Raez, M.B.I., Hussain, M.S. & Mohd-Yasin, F. 2006. Techniques of EMG signal analysis: detection, processing, classification, and applications. *Biological Procedures Online, 8,* 11-35.
18. Saponas, T. S., Tan, D. S., Morris, D. & Balakrishnan, R. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. *CHI '08,* 515-524.
19. Sassaroli, A., Zheng, F., Hirshfield, L.M., Girouard, A., Solovey, E.T., Jacob, R.J.K. & Fantini, S. 2008. Discrimination of Mental Workload Levels in Human Subjects with Functional Near-Infrared Spectroscopy. *J Innovative Optical Health Sciences*, 1(2), 227-237.
20. Schlesinger, G. Der Adechanische Auflau der kunstlichen Glieder. 1919. Ersatzglieder und Arbeitshilfen, 1. Springer Verlag, Berlin.
21. Tenore, F., Ramos, A., Fahmy, A., Acharya, S., Etienne-Cummings, R. & Thakor, N. 2007. Towards the Control of Individual Fingers of a Prosthetic Hand Using Surface EMG Signals. *IEEE EMBS*.
22. Wang, G., Wang, Z., Chen, W. & Zhuang, J. 2006. Classification of Surface EMG signals using optimal wavelet packet method based on Davies-Bouldin criterion. *Med Biol Eng Comput* 44, 865-872.
23. Wheeler, K.R, Chang M.H. & Knuth K.H. 2006. Gesture-Based Control and EMG Decomposition. *IEEE Trans. on Systems, Man, and Cybernetics*, 36(4).
24. Wilson, A. 2005. PlayAnywhere: a compact interactive tabletop projection-vision system. *UIST '05*, 83-92.
25. Yatsenko, D., McDonnall D. & Guillory, S. 2007. Simultaneous, Proportional, Multi-axis Prosthesis Control using Multichannel Surface EMG. *IEEE EMBS*.