

Leveraging Dual-Observable Input for Fine-Grained Thumb Interaction Using Forearm EMG

Donny Huang¹, Xiaoyi Zhang¹, T. Scott Saponas², James Fogarty¹, Shyamnath Gollakota¹
Computer Science & Engineering, DUB Group, University of Washington¹ Microsoft Research²
{donnyhuang, xiaoyiz, jfogarty, gshyam}@cs.washington.edu ssaponas@microsoft.com

ABSTRACT

We introduce the first forearm-based EMG input system that can recognize fine-grained thumb gestures, including left swipes, right swipes, taps, long presses, and more complex thumb motions. EMG signals for thumb motions sensed from the forearm are quite weak and require significant training data to classify. We therefore also introduce a novel approach for minimally-intrusive collection of labeled training data for always-available input devices. Our *dual-observable input* approach is based on the insight that interaction observed by multiple devices allows recognition by a primary device (e.g., phone recognition of a left swipe gesture) to create labeled training examples for another (e.g., forearm-based EMG data labeled as a left swipe). We implement a wearable prototype with dry EMG electrodes, train with labeled demonstrations from participants using their own phones, and show that our prototype can recognize common fine-grained thumb gestures and user-defined complex gestures.

Author Keywords

Always-Available Interaction; Dual-Observable Input; EMG.

INTRODUCTION

Thumbs have evolved to be strong and flexible, and we have become skilled at using their dexterity in manipulating tools. Input interfaces designed to take advantage of thumbs, such as touchscreen phones or video game controllers, are common and natural ways for people to interact with devices. If we can create wearable input devices that detect fine-grained thumb gestures, we can bring the advantages of thumb-based interaction to systems without physical transducers. For example, we imagine people using mid-air swipes of their thumb to navigate wearable displays.

ThumbsUp is a forearm-based electromyography (EMG) input system that can recognize fine-grained thumb gestures. In particular, the gestures that ThumbsUp recognizes are the same motions that people use in daily interaction with their touchscreen phones. Because it is worn on the wrist, ThumbsUp can be used in situations where a person cannot physically touch a device (e.g., wearing gloves, working with materials that coat the hands) or where having a device in the hand is simply inconvenient (e.g., cooking, using other tools).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST 2015, November 8–11, 2015, Charlotte, NC, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-3779-3/15/11 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2807442.2807506>

Although prior work demonstrates forearm-based recognition of finger gestures using EMG [15, 16, 17] or pressure [2], it is limited to coarse gestures requiring sustained pressure and the use of multiple fingers (e.g., pinches, flicks), which result in relatively strong EMG signals. Thumb gestures produce fairly weak corresponding EMG signals. The weakness of these signals means sensors are sensitive to placement, which can change each time a device is worn. Further, thumb gestures correspond to complex combinations of muscle motions, so EMG signals vary between people. Our experimentation has found classification of fine-grained thumb gestures using forearm EMG requires multiple demonstrations per gesture per sensor placement per person.

We address this training data challenge via *dual-observable input*, a novel approach for minimally-intrusive collection of labeled training data for always-available input devices. The key insight is that people frequently perform thumb gestures on their phone touchscreen, and the phone can already recognize these gestures. An always-available input system like ThumbsUp can therefore obtain labeled demonstrations by observing this same gesture (i.e., using recognition by the phone to label EMG data captured from the same gesture). In addition, training data will be gathered over time, with a person performing the gesture across different sensor placements and environments. These demonstrations can then be used to train the always-available input system to recognize the same gestures performed without the phone.

We prototype ThumbsUp as a custom Bluetooth-enabled EMG board with dry EMG electrodes and a set of Android apps. We train this prototype using dual-observable input in a study with participants providing labeled demonstrations through interaction with their phones during normal daily activities. We show we can use forearm-based EMG to distinguish between left swipe, right swipe, tap, and long press thumb gestures with an accuracy of 82.9%. We further show we can recognize complex user-defined gestures, and we briefly discuss potential applications of ThumbsUp and the dual-observable input approach to labeled training data.

Specifically, we make the following contributions:

1. We introduce a forearm-based EMG input system that can recognize fine-grained thumb gestures, including swipes and taps as well as complex user-defined gestures.
2. We introduce dual-observable input as an approach to collecting training data for always-available input devices.
3. We build a prototype forearm-based EMG device and demonstrate the feasibility of training gesture recognition through dual-observable interactions with a phone.

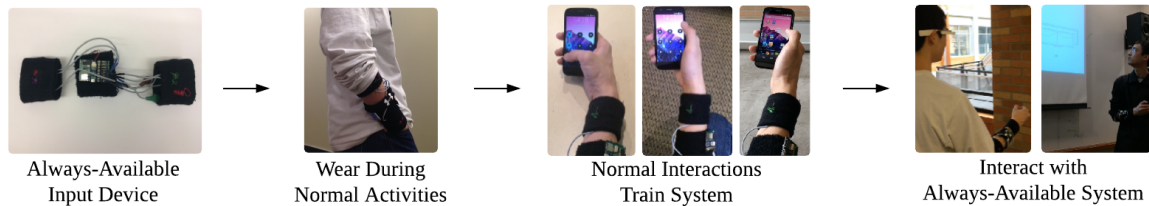


Figure 1: Interaction with a dual-observable input system: (1) An always-available input system is initially untrained, requiring per person per gesture training demonstrations. (2) A person wears the always-available device during normal activities. (3) Normal interactions with other devices collect labeled training demonstrations for the always-available input system. (4) The always-available system is available for interaction, but also continues to learn.

RELATED WORK

EMG has previously been used for detecting finger and hand gestures with a forearm device [18]. Saponas et al. propose a wearable EMG armband enabling interaction in scenarios where a display is not available (e.g., eyes-free control of a music player), a person’s hands are busy (e.g., carrying something, steering a vehicle), touching a display is undesirable (e.g., cooking, gardening, sculpting, changing engine oil), or in playful new contexts (e.g., using air guitar gestures to control a video game) [15, 16, 17]. They recognize lifting, pressing, and pinching, even while gripping an object [15, 16, 17]. Ju et al. explore use of data from multiple people to perform the same pinch gesture [8]. These gestures involve multiple fingers and significant pressure, resulting in a signal at the forearm that is much more prominent than those we recognize here. Additional work has recognized sign language gestures [10, 11, 21], which are again much more prominent in the signal domain.

A number of additional efforts examine always-available input [13]. Kim et al. use a wrist-worn 3D infrared camera to recognize finger gestures [9]. However, cameras can suffer from line-of-sight occlusions and raise privacy issues that might limit them from being always available. Pressure and bio-acoustic sensors have also been used for designing forearm-based devices. Deyle et al. use piezoelectric sensors to distinguish among gestures (e.g., flick, multi-finger snap, rubbing thumb over the index finger, moving the foot in different directions) [3]. Dementyev et al. use an array of force-sensitive resistors to detect finger pinch gestures [2]. Because their signals change each time the device is removed and put back on, they note a need for new approaches to collecting training data. Rekimoto proposes GestureWrist, a watch-like device that recognizes hand gestures using accelerometers and capacitance changes associated with different hand shapes [14]. Commercial systems such as the Myo [19] use a combination of accelerometers and EMG, but it is unclear if they can detect fine-grained finger gestures.

We are not aware of prior work using dual-observable input to train always-available recognition. Harada et al. propose using speech to provide training labels for activity recognition, as the act of providing a label can interfere with the activity itself [6]. We also propose obtaining labels via a separate channel, but further identify existing natural interactions as a minimally-intrusive approach to training. Lyons et al. propose dual-purpose speech, in which speech that is appropriate for human conversation also provides input to a system [12]. Our focus is similarly on preserving

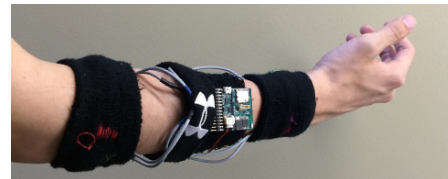


Figure 2: Our prototype hardware, a custom EMG board with dry EMG electrodes and Bluetooth communication. The setup is fully mobile and can be worn relatively comfortably for a significant period of time.

and leveraging existing natural interactions, but further considering how they might help train always-available recognition systems. Finally, Chen et al. [1] coordinate motion and touch input between a phone and a watch for multi-device gestures and sensing. We similarly leverage a combination of devices for new interactions.

We propose that always-available devices can be trained by gestures performed on everyday devices such as phones. A mirror of our motivation is that everyday gestures normally performed on devices such as phones can be performed without that phone. Gustafson et al. pursue this in their work on imaginary interfaces [4], including their proposal of an imaginary phone [5]. Their prototype imaginary phone uses a shoulder-mounted camera to detect taps and slides on an empty non-dominant hand, mapping that input to an actual phone. A key challenge of imaginary interfaces is a person’s ability to manipulate the interface without feedback, a point we revisit in our analysis of dual-observable interaction.

SYSTEM DESIGN AND IMPLEMENTATION

ThumbsUp is a forearm-based wearable EMG input device that can detect and classify thumb gestures (e.g., left swipes, right swipes, taps, long presses, user-defined gestures). We envision these gestures can in turn be used to interact with other devices (e.g., music players, ubiquitous computing environments, wearable computers). This section details our prototype of a forearm-based wearable EMG input device, our collection of training data via dual-observable interaction in a set of Android apps, and our algorithm for classifying thumb gestures based on collected training data.

Hardware and Placement

Our prototype is implemented with a custom EMG board that includes: a Texas Instruments ADS1298 analog front-end device, pins for connecting EMG electrodes, an on-board accelerometer, a microSD memory card for storing data, and a Bluetooth module for communicating with other devices during recognition and during training via dual-observable input. For dry EMG electrodes, we use silver chloride

electrode pellets connected to the board using EMG snap leads. For input, we use the on-board accelerometer and seven EMG electrodes, with two additional electrodes as an active ground. The accelerometer is sampled at 100 Hz, and the EMG signals are sampled at 250 Hz.

The board and EMG electrodes are sewn onto three armbands, with labels indicating how they should be worn (Figure 2). When a person puts on the bands, the electrodes will be located such that they roughly correspond to the locations of the extrinsic hand muscles that control the thumb. The placement does not need to be exact, due to there being multiple sensors per muscle and because our dual-observable input approach can collect labels for varying placements.

The setup is fully mobile, takes less than a minute to put on, and can be worn relatively comfortably for significant periods. The form and packaging could be improved in terms of appearance and physical robustness. Our current components also require Microsoft Windows drivers for the communication channel. Communication with an Android phone is therefore implemented using a tablet as a relay, thus requiring the device remain within 10 meters of the tablet. This impacts our study protocol presented in the next section, but could be readily addressed in a more refined prototype.

Training via Dual-Observable Input

A traditional approach might collect training data in specific training sessions. We have noted this is limited here due to a need for labeled demonstrations per sensor placement per person. We instead aim to collect labeled demonstrations through a person’s everyday interaction with their phone. Our dual-observable input approach has three advantages: (1) it enables collecting a large number of training demonstrations, (2) demonstrations are person-specific and therefore robust to individual variation, and (3) demonstrations will be gathered under a relatively wide variety of circumstances, including being robust to different specific sensor placements.

One might maximize training data volume by capturing all phone touch events, but this presents two major barriers: (1) observing touch events across all applications requires root access, and (2) within a stream of raw touch events, it is difficult to know which correspond to purposeful gestures, thus compromising training data quality. We therefore pursue an approach of modifying the default Android *launcher* and *lock screen*. These can be replaced without root access, and our modifications preserve their appearance and functionality (e.g., lock screen security). The launcher allows observation of purposeful left swipes, right swipes, taps, and long presses. The lock screen further allows observation of user-defined gestures performed on a 3-by-3 grid of dots. Prior studies have found that people perform unlock gestures 10 to 200 times per day [20], and we present a study in the next section which further examines the frequency of these gestures.

Each time a person uses a gesture to manipulate the widgets in our modified apps, ThumbsUp uses the Bluetooth connection to inform the EMG board that a gesture has been performed. This allows the gesture to be associated with the previous few seconds of signals. We additionally filter any demonstrations

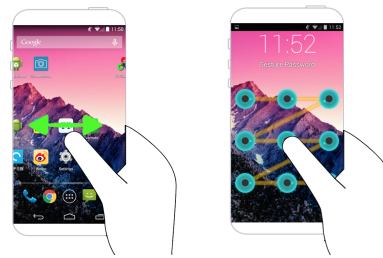


Figure 3: We collect left swipe, right swipe, tap, and long press demonstrations from the Android launcher (left). We collect complex user-defined gestures from the Android lock screen (right).

that the phone’s accelerometer detects were performed under acceleration greater than 0.2g. These are noisy because the forearm-based EMG signals are impacted by arm motions.

We also note the launcher imposes a minimum threshold on the width of left swipe and right swipe gestures, as part of its own need to ensure the purposefulness of those gestures. We have informally observed that very small swipes use so little strength that they result in very weak EMG signals that are unlike signals obtained when people perform gestures without a phone. The launcher’s own minimum threshold was sufficient in this work, but this does surface a design consideration for future dual-observable input systems.

Thumb Gesture Classification Algorithm

Thumb gestures are classified by filtering and smoothing the signals, then matching them against training data using a k-nearest neighbor (KNN) algorithm with a custom similarity metric. Specifically, we perform the following:

Filtering: For each EMG channel, we use the past 2 seconds of signal ($2s \times 250 \text{ Hz} = 500$ samples). We apply a 80 Hz high-pass Hanning window filter and take the absolute value. This helps reduce 60 Hz noise caused by power lines and helps remove drift caused by changing arm stance, changing environment, and movement of wires in our prototype.

Smoothing: For each EMG channel, we divide the filtered signal by the standard deviation of that channel for the past 2 minutes. This normalizes the signal, allowing us to better compare signals from different environments. We then smooth the signal by summing windows of 20 samples, with 10 samples overlap (i.e., channels now contain 49 samples).

Classification: We then classify a gesture using KNN to compare with our filtered and smoothed training gestures. We currently set $k=5$ and use a similarity metric of the maximum value of the normalized cross correlation between gestures. For a pair of gestures, we calculate the normalized cross correlation of each channel, sum the resulting sequences, and find its maximum value (i.e., larger scores are more similar).

For left swipe, right swipe, tap, and long press, we classify according to the majority label. Our user-defined gestures are unlock gestures, so we additionally want to know that the *best* match is a *good* match. We know the user-defined gesture is not one of our simple gestures, so we compare against other user-defined gestures and against some random noise samples, requiring the average similarity for the $k=5$ best matches be above a threshold.

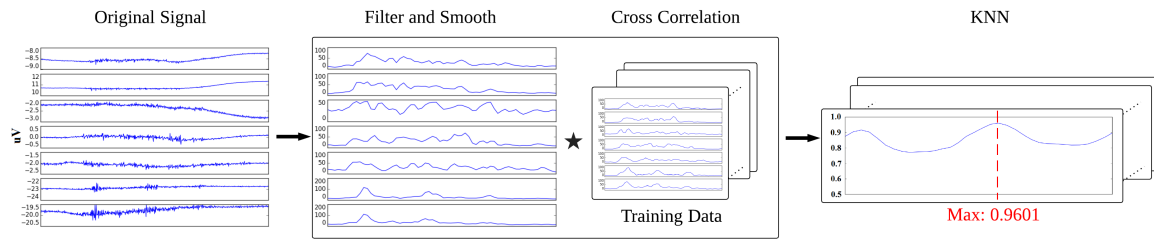


Figure 4: Gestures are classified by: (1) Considering the past 2 seconds of signal from each EMG channel. (2) Filtering and smoothing the signals. (3) Comparing against all alignments to training demonstrations using normalized cross correlation. (4) Using KNN to choose a class.

This classification strategy provides two advantages. First, KNN accuracy will improve as observations are collected across varying placements and environments without requiring an explicit model of such variation, a good synergy with our dual-observable input approach to collecting a large number of labeled gestures. Second, cross correlation considers all offsets and thus allows gestures to be compared even when misaligned, so we do not need to precisely segment each gesture from its two second window.

EVALUATION

We evaluated the accuracy of ThumbsUp and its training via dual-observable input in a study with 7 people from our organization (2 female, 5 male, ages 21 to 27). None had extensive experience with EMG or other biomedical signals. All had an Android phone on which they were willing to allow us to install our apps, and all were willing to allow us to monitor their usage patterns with those apps. Each participant was compensated with a \$30 Amazon gift card.

We ultimately envision dual-observable input systems that are continuously worn and trained throughout our daily lives, but the previously discussed hardware limitations of our current prototype made that infeasible for our evaluation. We therefore devised a two-part study. Participants first used their phones normally for an entire day so that we could monitor how often they performed each gesture in our instrumented apps. We then arranged three EMG data collection sessions with each participant, wherein they worked normally in their normal work environment, interacted with their phone as normal, but were also prompted to interact with their phone by notifications that required unlocking the phone, navigating via the launcher to the alerting app, and pressing to dismiss the prompt. This section details our studies and our results.

Data Feasibility Study

Prior to participation in any EMG data collection, participants installed our apps to monitor their gesture usage patterns with regard to the feasibility of training via dual-observable input. We collected data from each participant for an entire day, with no intervention except the installation of our instrumented apps. Figure 5 shows how often each participant performed each gesture. Participants performed an average of 34 left swipe, 31 right swipe, 62 tap, 5 long press, and 54 unlock gestures in our instrumented apps. This suggests a large amount of potential training data that could be gathered through our approach to dual-observable input.

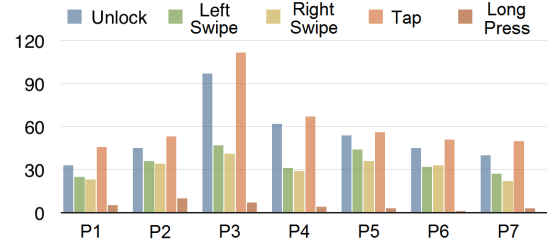


Figure 5: Participant gesture usage in our instrumented apps for a day.

Data Collection Study

We collected data in three sessions for each participant. Sessions were held in their normal work environments (except where electrical noise interfered with the EMG sensor itself, in which case data collection occurred in a common work space). Participants were free to move as desired, including walking around and performing gestures in any pose. We have noted hardware limitations required participants stay within 10 meters of a data collection relay, but we positioned this to allow each participant to work normally. Sessions were an hour long, distributed across multiple days whenever possible, with our EMG device completely removed and put back on between sessions. Participants also defined their own complex unlock gestures (e.g., Figure 10 shows several participant-defined gestures).

Training data was collected with participants gesturing with their phone during the first 40 minutes of each session. Demonstrations were captured whenever the participant interacted with the launcher or lock screen. To ensure adequate data collection during this abbreviated schedule, we also prompted participants, with the goal of obtaining a total of 8 to 10 training demonstrations per gesture per session.

Test data was collected with participants gesturing without their phone during the final 20 minutes of each session. Gesturing without a phone feels different, but people can adapt to this difference. We advised not to make other large motions while performing a gesture and to use some strength in performing a gesture (e.g., an index finger can be treated like the screen when performing left swipe, right swipe, tap, and long press thumb gestures, an opposing hand can be treated like a surface when performing a complex unlock gesture). Participants were able to rehearse, with live recognition feedback, until they reported they were comfortable performing the gestures without the phone. We then collected 5 test demonstrations per gesture per session.

Actual Gesture Performed	Left Swipe	89 (84.8%)	5 (4.8%)	9 (8.6%)	2 (1.9%)
	Right Swipe	11 (10.5%)	79 (75.2%)	6 (5.7%)	9 (8.6%)
	Tap	13 (12.4%)	9 (8.6%)	83 (79.0%)	1 (0.9%)
	Long Press	2 (1.9%)	3 (2.9%)	3 (2.9%)	97 (92.4%)
		Left Swipe	Right Swipe	Tap	Long Press
		Gesture Classified			

Figure 6: ThumbsUp has an overall 82.9% accuracy at classifying left swipe, right swipe, tap, and long press thumb gestures.

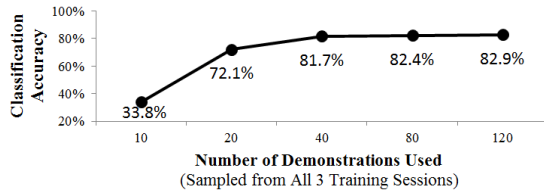


Figure 7: Randomly sampling from all 3 training sessions, recognition accuracy begins to plateau after training with 40 demonstrations.

RESULTS

Figure 6 shows our overall 82.9% accuracy classifying left swipe, right swipe, tap, and long press thumb gestures. Results were obtained by training a per-participant model with data from all three training sessions, then testing on all three test sessions. This is consistent with continuously collecting per-person labels via dual-observable input (i.e., more realistic than testing between-person or between-session).

Figure 7 next examines how accuracy improves with the number of available training demonstrations. Results were obtained by randomly sampling from the training pool used above, thus drawing upon all three training sessions. Accuracy grows from 33.8% with 10 demonstrations, to 81.7% with 40 demonstrations, then more slowly to 82.9% with the full 120 demonstrations. Figure 8 asks a slightly different question, examining how accuracy improves when the demonstrations from randomly chosen training sessions are made available. More sessions notably improve the model, from 67.6% with 40 demonstrations in 1 session, to 76.0% with 80 demonstrations in 2 sessions, to 82.9% with the full 120 demonstrations in 3 sessions. These analyses suggest ThumbsUp benefits from training data collected under different conditions more than it does from a larger number of training demonstrations under similar conditions. Dual-observable input is a good fit for continuously collecting such diverse demonstrations (e.g., different environments, different times, minor differences in sensor placement).

Imaginary interfaces face the fundamental challenge that people perform input without the visual or tactile feedback found in traditional interfaces [4, 5]. We observed some participants initially found it difficult to perform thumb gestures without the phone. Even after participants reported being comfortable, we believe some of the classification errors in Figure 6 are due to differences in performing the

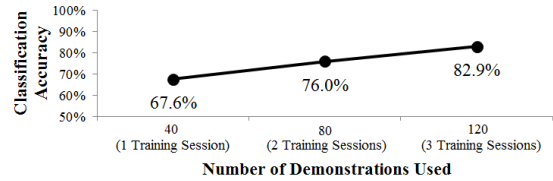


Figure 8: Recognition accuracy continues to improve with data from additional sessions. Collecting data in different conditions is more valuable than collecting larger volumes of data in similar conditions.

	With Phone	Without Phone
Classification Accuracy	89.2%	82.9%

Figure 9: Comparing recognition accuracy with and without the phone, we see that participants perform gestures differently without the phone.

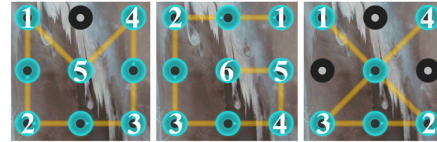


Figure 10: Examples of user-defined complex unlock gestures. Gestures ranged from simple geometric shapes, to more intricate paths.

gesture without the phone (e.g., pressing too hard against the index finger during a tap, thus activating the same muscles normally used in a tap). Figure 9 examines this by comparing the classification accuracy of gestures performed with and without the phone. We evaluate accuracy of gestures performed with the phone using a leave-one-out cross-validation with a participant's entire training data. The results show that differences in how people perform a gesture without the phone does impact accuracy, and that dual-observable input systems therefore need to consider how consistently people can perform gestures without the feedback provided by a data collection device.

Evaluation of user-defined gesture recognition is highly dependent upon the specific gesture chosen by each participant and the acceptable balance between precision and recall. For the gestures chosen by our participants, a recall of 70% results in a precision of 91%. Most errors appear to be due to two participants. Restricting definition to more distinguishable gestures could likely improve recognition.

DISCUSSION AND CONCLUSION

We introduce the first forearm-based EMG input system that can recognize fine-grained thumb gestures, including left swipes, right swipes, taps, long presses, and more complex thumb motions. We achieve this using dual-observable input as an approach to collecting labeled training data for always-available input devices. We demonstrate and explore our approach using a prototype implementation of a forearm-based wearable EMG input device, a set of Android apps for collecting training data via dual-observable interaction, and algorithms for classifying thumb gestures based on collected training data.

We also implemented two simple applications demonstrating potential usage of ThumbsUp, presented in Figure 11 and our associated video. The first is a presentation remote control, in which a right swipe advances to the next slide and a left



Figure 11: As demonstrations, we implement a ThumbsUp presentation remote control (left) and a Google Glass music player (right).

swipe returns to the previous. The second is a music player on Google Glass. It supports a right swipe to go to the next song, a left swipe to the previous, a tap to pause/play, and a long press to dim the screen for power saving.

These applications require a method for online detection of when a gesture is performed and should be classified. We implemented this by using the wearable's accelerometer to ensure the arm is steady (i.e., arm-related EMG signal is low), then applying an energy threshold to the EMG signals.

In addition to our contributions to forearm-based EMG recognition of fine-grained thumb gestures, we imagine a variety of opportunities for the dual-observable input approach. One opportunity for forearm-based EMG is suggested by Harrison et al.'s TapSense, in which a phone responds differently to taps with a finger, nail, knuckle, or stylus [7]. Their implementation cannot preview these different behaviors because it is unaware of which type of tap a person intends to perform. We imagine using dual-observable interaction with forearm-based EMG to learn what arm poses precede different types of taps, thus enabling previews before the different types of taps are performed. We can also imagine dual-observable approaches with very different sensors (e.g., Kinect limb-tracking within a game used to train a wearable sensor used away from the camera).

Dual-observable input is not a panacea. We have noted that people can perform gestures differently absent feedback, impacting recognition. Our informal experimentation has also found limitations of forearm-based EMG for recognizing fine-grained thumb gestures. Even with training examples, differentiating down swipe and up swipe thumb gestures from left swipe and right swipe gestures remains challenging. A minimum swipe size was imposed by the Android launcher, and we informally noted that smaller gestures would have been difficult to recognize. Designers of dual-observable input will need to account for such constraints in creating effective systems. But when it can be applied, our results demonstrate the potential for dual-observable input to enable interactions for which it would be otherwise infeasible to gather necessary per-person training data.

ACKNOWLEDGEMENTS

We thank Liusha Huang and Yuchen Jin. This work was funded in part by the Intel Science and Technology Center for Pervasive Computing, a Google Faculty Award, Nokia, and National Science Foundation awards CNS-1420654 and OAI-1028195.

REFERENCES

1. Chen, X., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. *UIST 2014*, 159-168.

2. Dementyev, A., and Paradiso, J. A. WristFlex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors. *UIST 2014*, 161-166.
3. Deyle, T., Palinko, S., Poole, E. S., and Starner, T. Hambone: A Bio-Acoustic Gesture Interface. *ISWC 2007*, 3-10.
4. Gustafson, S., Bierwirth, D., and Baudisch, P. Imaginary Interfaces: Spatial Interaction with Empty Hands and Without Visual Feedback. *UIST 2010*, 3-12.
5. Gustafson, S., Holz, C., and Baudisch, P. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. *UIST 2011*, 283-292.
6. Harada, S., Lester, J., Patel, K., Saponas, T. S., Fogarty, J., Landay, J. A., and Wobbrock, J. O. VoiceLabel: Using Speech to Label Mobile Sensor Data. *ICMI 2008*, 69-76.
7. Harrison, C., Schwarz, J., and E., H. S. TapSense: Enhancing Finger Interaction on Touch Surfaces. *UIST 2011*, 627-636.
8. Ju, P., Kaelbling, L. P., and Singer, Y. State-Based Classification of Finger Gestures from Electromyographic Signals. *ICML 2000*, 439-446.
9. Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., and Olivier, P. Digits: Freehand 3D Interactions Anywhere using a Wrist-Worn Gloveless Sensor. *UIST 2012*, 167-176.
10. Kosmidou, V. E., Hadjileontiadis, L. J., and Panas, S. M. Evaluation of Surface EMG Features for the Recognition of American Sign Language Gestures. *EMBC 2006*, 6197-6200.
11. Li, Y., Chen, X., Tian, J., Zhang, X., Wang, K., and Yang, J. Automatic Recognition of Sign Language Subwords Based on Portable Accelerometer and EMG Sensors. *ICMI-MLMI 2010*.
12. Lyons, K., Skeels, C., Starner, T., Snoeck, C. M., Wong, B. A., and Ashbrook, D. Augmenting Conversations Using Dual-Purpose Speech. *UIST 2004*, 237-246.
13. Morris, D., Saponas, T. S., and Tan, D. Emerging Input Technologies for Always-Available Mobile Interaction. *Foundations and Trends in Human-Computer Interaction*, 4, 4 (2011), 245-316.
14. Rekimoto, J. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. *ISWC 2001*, 21-27.
15. Saponas, S. T., Tan, D. S., Morris, D., and Balakrishnan, R. Demonstrating the Feasibility of Using Forearm Electromyography for Muscle-Computer Interfaces. *CHI 2008*, 515-524.
16. Saponas, S. T., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J. A. Enabling Always-Available Input with Muscle-Computer Interfaces. *UIST 2009*, 167-176.
17. Saponas, T. S., Tan, D. S., Morris, D., Turner, J., and Landay, J. A. Making Muscle-Computer Interfaces More Practical. *CHI 2010*, 851-854.
18. Tenore, F., Ramos, A., Fahmy, A., Acharya, S., Etienne-Cummings, R., and Thakor, N. V. Towards the Control of Individual Fingers of a Prosthetic Hand Using EMG Signals. *EMBC 2007*, 6145-6148.
19. Thalmic Labs. Myo Gesture Control Armband. <https://www.thalmic.com/myo/>.
20. Truong, K. N., Shihpar, T., and Wigdor, D. J. Slide to X: Unlocking the Potential of Smartphone Unlocking. *CHI 2014*, 3635-3644.
21. Zhang, X., Chen, X., Li, Y., Lantz, V., Wang, K., and Yang, J. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Transactions on Systems, Man and Cybernetics*, 41, 6 (2011), 1064-1076.