Serendipity: Finger Gesture Recognition using an Off-the-Shelf Smartwatch

Hongyi Wen[1] **Julian Ramos Rojas**[2] **Anind K. Dey**[2]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States
wenhy12@mails.tsinghua.edu.cn, {julian, anind}@cs.cmu.edu

ABSTRACT
Previous work on muscle activity sensing has leveraged specialized sensors such as electromyography and force sensitive resistors. While these sensors show great potential for detecting finger/hand gestures, they require additional hardware that adds to the cost and user discomfort. Past research has utilized sensors on commercial devices, focusing on recognizing *gross* hand gestures. In this work we present Serendipity, a new technique for recognizing unremarkable and *fine-motor* finger gestures using integrated motion sensors (accelerometer and gyroscope) in off-the-shelf smartwatches. Our system demonstrates the potential to distinguish 5 fine-motor gestures like pinching, tapping and rubbing fingers with an average f1-score of 87%. Our work is the first to explore the feasibility of using solely motion sensors on everyday wearable devices to detect fine-grained gestures. This promising technology can be deployed today on current smartwatches and has the potential to be applied to cross-device interactions, or as a tool for research in fields involving finger and hand motion.

Author Keywords
Wearable interfaces; Machine learning

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User Interfaces

INTRODUCTION
Wearable devices like smartwatches are becoming more popular but they have remained difficult to interact with. For instance, when we wear commercially available smartwatches or bands on our wrist we have to touch the relatively small screen or physical buttons to interact with the device, making simple tasks onerous and inefficient. Additionally, in some scenarios, when our other hand is occupied or it is inconvenient to perform a touch interaction, it is hard or even impossible for us to interact with the device in a natural way.

CHI'16, May 07-12, 2016, San Jose, CA, USA

DOI: http://dx.doi.org/10.1145/2858036.2858466

Expanding interaction space
There has been a lot of research focused on extending the interaction space outside the fixed screen. For example, SkinWatch [9] provides gesture input by sensing deformation of skin. Abracadabra [4] enables off-the-screen fine motor control by placing a magnet on the interacting finger. However these approaches require the use of the other hand (*i.e.*, hand not wearing the device) and specialized sensors to interact with the device.

Muscle-based Interfaces
Saponas *et al*. [11] used an electromyography (EMG) sensor on an armband to convert muscle movement to finger gestures. Dementyev *et al*. [3] developed a device with an array of force sensitive resistors (FSR) worn around the wrist to classify gestures from subtle tendon movement. This work used specialized sensors, which are not easy to integrate into current wearable devices.

Whole-hand gestures
Others have focused on using sensors that are already available on a smart phone or a Wii-controller to sense user gestures [1,7,10,12]. Bernaerts *et al.* [2] utilized the accelerometer sensor on a smartwatch to detect 3 forearm gestures. However, these approaches have been used to support only gross-motor or whole-hand gestures. These gestures are obtrusive and cannot be performed in an unremarkable way – without drawing attention to oneself.

In this work, we build on previous approaches, and the feasibility of leveraging mobile *sensors that are already available* on smartwatches to *sense fine-grained* user gestures that can be performed more unobtrusively in everyday life. We present Serendipity, a smartwatch-based system that can detect 5 subtle finger gestures: pinching, tapping, rubbing, squeezing, and waving. Our rationale for targeting these gestures is three-fold:

1. We want to support short interactions on wearable devices. Thus the gestures should be quick and easy to perform, with symbolic meanings for mapping to certain functions.
2. The current input methods for wearable devices are obtrusive. For example, on the Google Glass, a user has to scroll or tap on the frame. We chose our gestures to be less noticeable and more natural.
3. Performance of large-scale gestures (*e.g.* whole-hand moving) takes more physical effort than fine-grained gestures [15]. Frequent usage of large-scale gestures

may cause unnecessary arm fatigue, while subtler fine-grained gestures can reduce arm fatigue [13]. Similarly, fine-grained gestures require less attention to perform the gestures themselves and thus add little mental or physical effort to users [15].

The main contribution of our work is that we leverage motion sensors already available on mainstream mobile and wearable devices and demonstrate the feasibility of our approach for recognizing 5 fine-grained gestures with an average f1-score of 87%.

RELATED WORK

Saponas *et al*. [11] used a surface EMG sensor to detect subtle gestures like pinching and pressing a finger on a surface (*e.g.*, a table or the arm of a chair) since it directly senses the muscle activity by measuring electrical signals between electrode pairs. Lu *et al*. [8] combined an accelerometer sensor with an SEMG to improve recognition accuracy, but used the accelerometer for detecting larger-scale gestures and the SEMG for small-scale gestures. Commercial products based on EMG sensors such as the Myo [16] armband are available, however they have tended to focus on coarse-motor whole hand gestures, and may have difficulty distinguishing finer finger-based gestures. In addition, these devices were not intended to be worn all day. Dementyev *et al*. [3] developed a system based on FSR to detect tendon movement. An advantage of their system over the EMG approach is low energy consumption however, the cost and prototyping time is high when integrating with current wearable devices [3]. Mounted LEDs and cameras underneath the wrist [5,6] are also used to detect finger movement. In general, all of these previous approaches require extra hardware, which increases the cost and decreases the wearability of using them for detecting finger-based gestures. The disadvantages of these former approaches motivate us to build a system that can be easily installed and accessible to current devices.

One approach that would address these disadvantages would be to use motion sensors that are already built into commercial off-the-shelf smartwatches. Motion sensors, such as an accelerometer, measures accelerations from vibrations or movements. They have been used to capture noticeable and gross hand motions [1] and have achieved a high accuracy for a dictionary of 18 large-scale hand gestures. Xu *et al.* [14] used a motion sensor attached on the wristband to detect three types of gestures: arm, hand, finger. However, they used data collected from a single person and only explored when the user's wrist and arm were affixed to the chair, limiting their results.

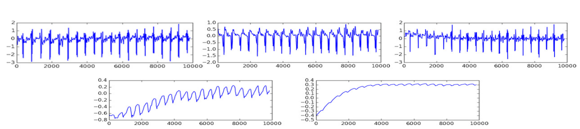

Figure 1. Time-series sensor data for pinching gesture. First row: accelerometer, gyroscope, linear accelerometer sensor; Second row: gravity, rotation sensor

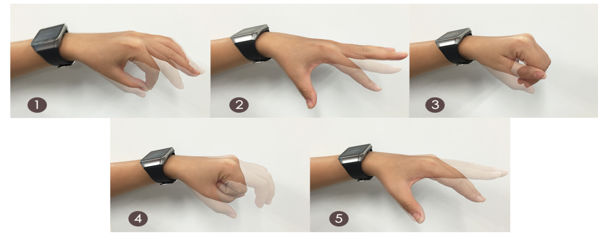

Figure 2. Gesture set: 1-Pinch; 2-Tap; 3-Rub fingers; 4-Squeeze; 5-Wave

We believe that the motion sensors on a watch are appropriate for sensing these fine-motor finger gestures. We posit that muscle activity differs when performing different finger gestures, causing different types of vibrations and motions that are captured by the motion sensors on the watch. In this paper, we show that this hypothesis is true.

SERENDIPITY SYSTEM

We used an off-the-shelf Samsung Galaxy Gear smartwatch to develop our gesture recognition system. In a pilot study, we recorded data from all the motion sensors available through the Android API: accelerometer, gyroscope, rotation and gravity sensors. We continuously collected data for each gesture for a length of 10 seconds. We used a sampling rate of 50Hz, which can detect gestures of up to 25 Hz according to the Nyquist Theorem. We used a 1 second sliding window for performing statistical feature extraction. We plotted the raw data and discrete-time Fourier transformation spectrograms. We did not observe any distinct patterns from the rotation and gravity sensors, but did observe them from the accelerometer, gyroscope and linear accelerometer sensors (Figure 1).

Gesture Set

As we mentioned in the introduction, our gestures need to be quick and unobtrusive. We designed 5 gestures: Pinch (to select), Tap (click), Rub fingers (scroll), Squeeze (confirm), and Wave (decline) (Figure 2). Pinch and Tap involve the slightest finger movement we captured. Rub involves high-frequency, low-amplitude movement. Squeeze and Wave are of larger scale, while still subtler than most hand gestures investigated in previous work.

Preprocessing

We collect raw sensor data for the 3 axes (x, y and z axis) of 3 sensors (accelerometer, gyroscope and linear accelerometer). To compensate for different watch orientations, we calculate the magnitude of the combined axes ($\sqrt{x^2+y^2+z^2}, \sqrt{x^2+y^2}, \sqrt{y^2+z^2}, \sqrt{x^2+z^2}$) (and refer to them as m, xy, yz, xz) . Thus we have 7 types of time-series data for each sensor. We calculate 7 statistical features from a 1-second sliding window: mean, standard deviation, max, min, 3 quantiles. Then, a Fast Fourier Transform (FFT) of the window produces 25 power bands, and we keep the lower 10 bands, representing the frequency spectrum from 0 ~ 10Hz. Since a single gesture takes around 0.4s to 0.8s to be performed, which yields a peak between 1~2Hz, this band range is sufficient to capture the frequency features of the gestures. We perform the same calculation for all 7 axes: x, y, z, m, xy, yz, xz, yielding 119 features for each sensor, for a total of 357. In general, gyroscope features are important for recognizing high-frequency gestures. Accelerometer features are useful in capturing amplitude differences. Since we test different orientations, the combined-axis features are also important.

Gesture Classification

We use a supervised machine learning approach to recognize our gestures. We test performance across a few basic classifiers such as a support vector machine (SVM), a Naive Bayes classifier, Logistic Regression and k-Nearest Neighbors (k-NN). Overall, the SVM achieved the best results in our pilot test. During real-world use, our system can try all of the classifiers and can automatically select the classifier and features that best fit a particular user.

Avoiding False Positives

In order to address false positive errors in recognizing any valid gesture, most past work employs an activation gesture. This gesture “wakes up” the gesture recognition system and indicates that the user is about to perform a meaningful gesture (not unlike saying “Ok Google” to turn on voice recognition on an Android device). We first try to reduce the number of false positive without requiring an activation gesture. We implemented an algorithm based on Dynamic Time Warping (DTW) [1] and k-NN. To distinguish noise (non-gestures) from the gestures, we calculate the DTW distance for the input sensor data from each labeled sample and select the nearest sample as the candidate gesture. If the distance is within a certain threshold (empirically determined), we infer that a gesture is being performed, and our system uses the classifiers to recognize the gesture; however, if the distance is beyond the threshold, we define it as ‘noise’ and the detection system will not react.

EXPERIMENT

We evaluated the effectiveness of our classification system through a multi-session experiment similar to [11]. 10 participants (4 females and 6 males) from our institutions volunteered to participate in the experiment. They ranged in age from 20 to 37 (Mean=28). All of our participants were right-handed, and all wore the device on their non-dominant hand based on their own preferences. We did not require the position of the device to be similar for all participants. The band tightness was also adjusted according to personal preferences, which mimics real world use of watches.

Orientation

In this study, the participants performed the assigned gestures in different orientations to capture as much variation within gestures as possible. By doing this, we increased the number of variations of the same gesture (due to rotation) that the system can recognize. We chose 3 orientations that exemplify 3 common use scenarios: 1) keep the arm flat as if the participant is looking at the watch; 2) lift the hand a little while the forearm leans on a table, as if the participant is interrupted while working on the laptop; 3) hold arm while the surface of the watch is perpendicular to the ground, as if the participant is or talking to others and does not have to look at the watch screen (see Figure 3).

Figure 3. Three common use scenarios with different orientations of the smartwatch

Training and Validation

Each participant took part in two data collection sessions occurring on two separate days. In the first session, participants provided data for training by performing each of the 5 gestures repeatedly, in 20 segments for each of the 3 orientations. The order of the gestures and orientations was counterbalanced across our participants. During data collection, participants were asked to perform the gestures (except for the rubbing finger gesture) at their own speed. We observed that most participants perform the gestures twice in each segment, for a total of 40 examples for each gesture in each orientation. For the rubbing finger gesture, the instruction given to participants was to ‘keep performing the gesture repeatedly for 1 second, since this gesture is designed to support continuous interaction.

On a separate day, participants came back for a second data collection session that follows the same scheme as the first session. We used the data from the first session for training the 4 classifiers (presented earlier) for each user and the second session for validation. Using training and testing data from different days simulates real-world usage, where the watch may not be worn and gestures may not be performed in exactly the same way on different days.

Noise Detection

In both sessions, we also recorded two minutes worth of data during which participants wore the smartwatch while typing on their laptops or talking to other people. This data represents instances where the user is not performing any gesture. We applied our DTW-k-NN algorithm to test its ability to reduce the false positive error rate. In future work, we will record whole-day data as non-gesture dataset.

RESULTS

The measure we used for gesture classification is the average f1-score for all five gestures (Figure 4). This graph shows the classification performance for our 10 participants. The mean f1-score across all participants is 87% (SD=7.9%). The best classification model differed amongst participants: SVM worked best for 5, and Logistic Regression for the other 5. Note that the performance for P5, P9 and P10 is lower than the others. These participants performed certain gestures (wave, tap, pinch, respectively)

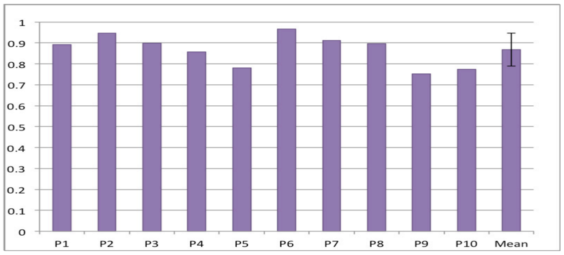

Figure 4. Average f1-score for classification

at different speeds in the two sessions, resulting in lower performance. This indicates that the participants may need more practice to perform gestures consistently.

The confusion matrix for the gestures is shown in Figure 5. We find that the rubbing fingers gesture outperformed the other gestures. Pinching and tapping were confused with each most often, as the hand-arm movement involved in these two gestures are similar.

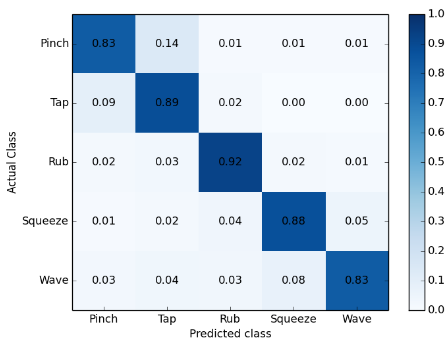

Figure 5. Confusion matrix for 5 finger gestures classification

While the classification of our gestures is quite good (all over 83%), we found that an activation gesture was necessary. With our DTW-kNN algorithm, the rate of false positives is 25% (*i.e.*, non-gesture detected as gesture), and 57% during typing. The error rate for false negatives. (*i.e.* gestures wrongly detected as noise) is 16%. A lower DTW-kNN threshold lowers the false positive rate to 9%, but at the cost of a much higher false negative rate (53%).

Based on these poor results, we chose to add an activation gesture. We selected the gesture with the highest classification rate as our activation gesture: rubbing fingers. The resulting false positive rate dropped to only 8% (11% during typing) and the false negative rate (*i.e.* gestures wrongly detected as noise) to 10%. As both rates seem high, we performed a follow-up small experiment with a single participant. We modified the activation gesture to be rubbing fingers *three times*. This reduced the false positive rate to 0.38% (0.84% during typing), and the false negative rate to 5%, and did not impact the classification accuracy.

User Feedback

After the data collection session, several of the participants offered informal feedback on the gestures. Some participants indicated that they may prefer the larger-scale gestures like squeezing rather than pinching because they do not require significantly more effort but provide a greater sense of control. Additionally, some of the participants also noted that the gesture set does not have to be too large as long as they can pick some of their favorite ones and those gestures function robustly.

DISCUSSION AND FUTURE WORK

Beyond improving the classification accuracy and expanding the range of gestures we can recognize, we now discuss ideas for further exploration with our fine-motor finger gesture recognition work.

Real-time detection System: Our demo system runs the gesture classification algorithm offline on a separate server. The smartwatch sends sensor data every 1s to the server through the watch's integrated Wi-Fi module and receives the classification result. In the future, we expect to migrate our algorithm to run on the watch itself or on the smartphone it has been paired with.

Going mobile: In this work, we explored the feasibility of using motion sensors to detect fine-motor gestures in 3 common use-case driven orientations, when the user is not moving. We will explore, in future work, whether we can detect these gestures while moving.

Applications: We see a number of future applications of this work. By building on the community's initial work with Apple's HealthKit, we see a large application area for our work in monitoring health and wellness. In particular, we would like to explore whether our techniques can be applied to detecting hand and finger motions in different orientation for diseases such as Parkinsons.

In addition, while we mapped our gestures to certain operational functions, for use on a smartwatch, we believe these gestures will also be valuable when interacting in a multi-device environment. For example, we can apply these gestures to improve interaction with mobile phones or laptops when interacting with them directly is inconvenient (*e.g.*, switch between applications using the rubbing finger gesture) or deal with interruptions in a short time (*e.g.*, decline a phone call by simply waving). Moreover, we envision our finger-based gesture interaction technique working well to augment head-mounted devices such as Google Glass, thus provide a less obtrusive approach for interaction with those devices. Integrated communication modules on the smartwatches (*e.g.,* Bluetooth, Wi-Fi) will make cross-device interactions easy to realize.

CONCLUSIONS

In this note, we present Serendipity, a novel fine-grained finger-based gesture recognition system that leverages integrated motion sensors on an off-the-shelf smartwatch. We demonstrate that our system can detect our 5 gestures with an average f1-score of 87%, across two different sessions where the smartwatch may be worn slightly differently. Our smartwatch-based system is accessible today and can be equipped on users for daily use.

REFERENCES

1. Ahmad Akl, Chen Feng, and Shahrokh Valaee. 2011. A Novel Accelerometer-Based Gesture Recognition System. *IEEE Transactions on Signal Processing* 59, 12: 6197-6205.
2. Yannick Bernaerts, Matthias Druwé, Sebastiaan Steensels, Jo Vermeulen, and Johannes Schöning. 2014. The office smartwatch: development and design of a smartwatch app to digitally augment interactions in an office environment. In *Proceedings of the 2014 companion publication on Designing interactive systems* (DIS Companion '14), 41-44. http://doi.acm.org/10.1145/2598784.2602777
3. Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14), 161-166. http://doi.acm.org/10.1145/2642918.2647396
4. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09), 121-124. http://doi.acm.org/10.1145/1622176.1622199
5. Bruce Howard and Susie Howard. 2001. Lightglove: Wrist-Worn Virtual Typing and Pointing. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers* (ISWC '01), 172-173.
6. David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12), 167-176. http://dx.doi.org/10.1145/2380116.2380139
7. Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive Mob Comput* 5, 6: 657-675.
8. Zhiyuan Lu, Xiang Chen, Qiang Li, Xu Zhang, and Ping Zhou. 2014. A Hand Gesture Recognition Framework and Wearable Gesture-Based Interaction Prototype for Mobile Devices. *IEEE Transactions on Human-Machine Systems* 44, 2: 293-299.
9. Masa Ogata and Michita Imai. 2015. SkinWatch: skin gesture interaction for smart watch. In *Proceedings of the 6th Augmented Human International Conference* (AH '15), 21-24. http://doi.acm.org/10.1145/2735711.2735830
10. Lorenzo Porzi, Stefano Messelodi, Carla Mara Modena, and Elisa Ricci. 2013. A smart watch-based gesture recognition system for assisting people with visual impairments. In *Proceedings of the 3rd ACM international workshop on Interactive multimedia on mobile & portable devices* (IMMPD '13), 19-24. http://dx.doi.org/10.1145/2505483.2505487
11. T. Scott Saponas, Desney S. Tan, Dan Morris, Jim Turner, and James A. Landay. 2010. Making muscle-computer interfaces more practical. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 851-854. http://doi.acm.org/10.1145/1753326.1753451
12. Thomas Schlömer, Benjamin Poppinga, Niels Henze, and Susanne Boll. 2008. Gesture recognition with a Wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction* (TEI '08), 11-14. http://doi.acm.org/10.1145/1347390.1347395
13. Jaime Ruiz and Daniel Vogel. 2015. Soft-Constraints to Reduce Legacy and Performance Bias to Elicit Whole-body Gestures with Low Arm Fatigue. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 3347-3350. http://doi.acm.org/10.1145/2702123.2702583
14. Chao Xu, Parth H. Pathak, and Prasant Mohapatra. 2015. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition using Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (HotMobile '15), 9-14. http://doi.acm.org/10.1145/2699343.2699350
15. Wendy Yee. 2009. Potential Limitations of Multi-Touch Gesture Vocabulary: Differentiation, Adption Fatigue. In *Proceedings of the 13th Annual Conference on* Human-Computer Interaction International: Novel Interaction Methods and Techniques (HCI International 2009). 291-300.
16. Thalmic Labs. Myo Gesture Control Arm Band: https://www.myo.com/