The Feasibility of Recognizing Pinch Gestures with Commodity Smartwatch Hardware and Machine Learning

Chris Kinzel, Dr. Anthony Tang
University of Calgary

ABSTRACT

Lacking the large screen size of mobile phones, smartwatch interaction faces challenges with regards to finger occlusion and dense hard to target controls that arise from their extremely tiny screens and input surfaces. Attempts to reconcile this problem often burden the user with bulky, expensive, extra hardware and lack simplicity and accuracy. Our approach uses a set of pinching gestures between the thumb and individual fingers on the smartwatch hand to provide a simple, socially acceptable, and comfortable input method. Our implementation makes use of common onboard sensors and machine learning to process the noisy, complex sensor data. Our method can be used to augment and extend traditional touch screen input on smartwatch devices. We evaluate our approach by measuring classification accuracy in a small user study with two different scenarios one where the user is stationary and one involving movement. Our results indicate that the approach is a feasible extension to touchscreen interaction but further work is needed to increase classification during movement.

Keywords: smartwatch interaction; input technology; gestures; machine learning; inertial sensing; acoustic sensing

Index Terms: blank

1 INTRODUCTION

Smartwatch devices are much smaller compared to traditional personal computing devices such as laptops and smartphones. Consequently, the available input surface for interaction with the device is also much smaller. Despite this decrease in input area, the dominate interaction method for consumer smartwatches is still touchscreen based, like that of a smartphone. Replicating more traditional input methods such as keyboard and mice is impractical and inconvenient due to the size of smartwatch devices and their form factor.

Touchscreens provide a powerful and rich way to interact with smartphone devices, however the use of touchscreens on smartwatch devices has yet to reach the same level of utility due to the small screen size approaching the size of the human finger. This results in many problems, for example multitouch gestures now become difficult or impossible to perform since a limited number of fingers can co-exist on the screen at the same time, also fingers can occlude the UI on the screen, a problem further hampered by UI components that are typically smaller than those on traditional personal computing devices. This occlusion coupled with tiny UI elements can result in accidental triggering of buttons, sliders, and other onscreen input that inconveniences users and can cause frustration (fat-finger problem). Thus, providing an input technique that is convenient and simple to use that does not consume the already scarce screen space, while at the same time requiring minimal changes to existing smartwatch technology is of interest to the HCI community.

A wide variety of smartwatch interaction techniques not involving the touchscreen have previously been explored. Many of these techniques require additional hardware and/or require movement that may not be socially acceptable or comfortable in all environments. We approach this problem by recognizing a set of pinching gestures between the fingers and thumb. There are four distinct pinching gestures i.e. a finger and thumb pinch gesture for each finger on the hand. Due to the lack of sensing hardware near the fingers and the desire to avoid additional hardware, our approach makes use of onboard sensor data that is present in almost all commodity smartwatch devices. Specifically, the sensors we wish to investigate for this purpose are the accelerometer, gyrometer, and microphone. Due to the relatively noisy nature of these sensors, we use machine learning, specifically a support vector machine, that is trained to classify the described gestures in real-time. Our approach does not require any additional hardware, is relatively simple and fast, and provides a socially non-intrusive way of extending the interaction with the device that does not consume screen real estate allowing the screen to serve primarily as an output unit. Pinching gestures do not require the use of the other hand and can be performed with much smaller motions than flicking or rotating the wrist or arm. The use of pinching gestures for interaction with wearable computing devices has been previously explored [1, 2, 3, 4], those approaches either required complementary hardware to function or did not make use of the full suite of sensors available on consumer smartwatches.

We evaluate our approach by assessing classification accuracy of the system using 10-fold cross validation on users performing the four different pinching gestures using data collected from a Sony SmartWatch3. Data collection is performed in a lab environment with two scenarios, one where the user is stationary sitting in a chair and the other where the user is in motion walking in a straight line and performing the gestures at the same time. User-dependent, independent, and user adaptive are explored.

We provide the following contributions:

- Design and implementation of a machine learning model for pinch gesture classification using IMU and microphone sensors
- Evaluation of pinching gesture accuracy in a laboratory environment
- Evaluation of user-dependent, independent, and user-adaptive models for personalization of the pinching gesture model.

2 THEORY

When pinching the fingers and thumb impact from the collision creates micro vibrations that propagate through the skin and other tissues at high frequencies from the contact point towards the microphone in the smartwatch. These waves will travel through different types of tissues (bone, skin, muscle) each of which affects the propagation of the wave differently, resulting in a unique pattern for every gesture. In addition, the wrist and arm slightly rotate and wobble in 3D space, the direction and magnitude of such motion is dependent on the pinching gesture being performed. For example, when pinching the pinky with the thumb the wrist slightly rotates away from the user. These minute physical changes are detectable from sensors present on the smartwatch. Specifically, the wobbling movement can be detected by a 3-axis accelerometer, the rotational movement by a 3-axis gyrometer and the vibrations that propagate through the skin by a regular acoustic microphone. Our system samples these three sensors and computes a feature vector with 413 features from a 1-second window of captured data. This feature vector is fed to a support vector machine that has been trained from labelled feature vectors to recognize the input gesture.

3 DISCUSSION

Our laboratory setup involved accumulating captured user data from the watch using a simple on-board program which was then transferred to a desktop machine to perform training and evaluation offline. Two SVM's were trained, one to distinguish between a gesture action and noise (lack of a gesture). This "gesture detector" would in theory always be on capturing data using a 1-second sliding window with some overlap. If a gesture is detected the feature vector is passed to the second SVM (really a collection of SVM's to support multiclass classification) that determines which of the four gestures was performed. To train the SVM's we use three different approaches, user-dependent training, user-independent training, and adaptive training. Each of these approaches is outlined and discussed below.

User-dependent

In the user-dependent approach the SVM used to classify between the different gestures is trained solely on data captured from the user themselves. This approach has the advantage that the system is tuned for that particular user and is able to take advantage of any subtleties in the way they perform the gestures, but has the disadvantage that the user may perform the training process poorly (i.e. user does not pay attention during training) and thus the performance of the system will reflect this. Also, in a real-world scenario the user-dependent training model requires the most data collection since all data must come from the user which can be time consuming and slow.

User-independent

In the user-independent approach the SVM used to classify between the different gestures is trained on data aggregated from multiple different users and all users use the same classification model. This approach has the advantage that the system is quick and easy to train and that training data containing errors can be removed by an expert to avoid tainting the system. The disadvantage to this approach is that users may perform gestures differently from one another resulting in confusion for the classifier and some of these differences could even be exploited to make classification easier for some users but is left untapped.

User-adaptive

In the user-adaptive approach the best of both worlds is used for training. The user-dependent and user-independent approaches are combined to produce a system that is both quick and easy to train but is also tailored to each particular user. Essentially a large aggregated of data is combined with a small number of training examples for the particular user to train the SVM which boosts performance by taking advantage of a robust set of training data and tuning it towards the subtleties of the user's method of performing the gestures.

4 CONCLUSION

In order to improve the convenience of smartwatch interaction without introducing any further on-body instrumentation or hardware, we showed how on-board sensing of commodity smartwatch devices can be used to classify a set of simple pinching gestures. Our classification accuracy is good enough for everyday use in a variety of non-mobile application scenarios. We can identify 4 unique pinching gestures each of which is easy to perform and non-invasive. While our system performs poorer in environments involving movement, we believe that future refinements to the extracted features and classification model could improve the system enough to make it a feasible approach to smartwatch interaction even in environments involving movement.

REFERENCES

[1] C. Harrison, D. Tan and D. Morris, "Skinput", Proceedings of the 28th international conference on Human factors in computing systems - CHI '10, 2010.

[2] A. Dementyev and J. Paradiso, "WristFlex", Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14, 2014.

[3] G. Laput, R. Xiao and C. Harrison, "ViBand", Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16, 2016.

[4] H. Wen, J. R. Rojas, and A. K. Dey, "Serendipity," Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16, 2016.