Gesture Recognition with Flex Sensors for Virtual Reality

From Neural Networks to Dynamic Time Warping

James Whitney
Cal Poly College of Engineering
San Luis Obispo, California
Contact@JasWhitney.com

ABSTRACT

Flex sensors are cheap sensors capable of detecting small movements through changes in its resistance. Here I present the results of my research to determine if when adhered to the forearm, flex sensors can be used for real time gesture recognition. The motivation for real time operations comes from the desire to use these sensors for mixed reality applications. The finding of my research is that flex sensors can be used for this application through the utilization of dynamic time warping and signal processing, the design described in this paper is capable of recognizing all of the gestures supported by the comparable Myo Armband[3] and can easily be adapted to additional custom gestures.

CCS CONCEPTS

• **Human-centered computing → Interaction devices**;

KEYWORDS

Flex Sensors, Gesture Recognition, Virtual Reality, Dynamic Time-Warping, Signal Processing

ACM Reference Format:

James Whitney. 2018. Gesture Recognition with Flex Sensors for Virtual Reality: From Neural Networks to Dynamic Time Warping. In *Proceedings of Example conference (Conference'18).* ACM, New York, NY, USA, 3 pages. https://doi.org/example

1 INTRODUCTION

Existing technologies exist for gesture recognition, most notably the Myo Armband a $200 premium product that relies on reading the EMF waves produced by your hand in order to determine gestures[2]. The theory with this research is that by detecting the motion of the arm with relatively cheap $8 flex sensors[1] and produce the same or better results through signal processing. These flex sensors are not a new technology and have been used in a variety of applications, most notably in old Nintendo power gloves. However the intended application for flex sensors in this applications is on the forearm instead of the hand, the reason being that the hand itself does not contain the muscles that apply it to move but actually only the tendons and ligaments, therefore it should be theoretically possible to infer the motion of the hand based on the

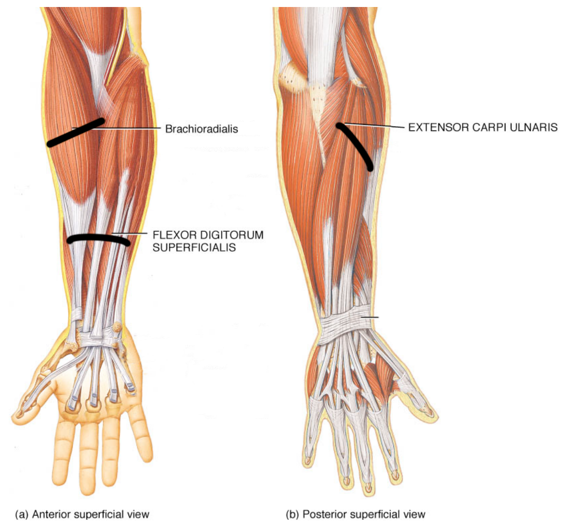

Figure 1: Forearm Anatomy diagram[4] edited to show the locations of the flex sensor placements.

movement of muscles in the forearm. However due to the fact that you cannot place flex sensors on the muscles themselves, only the skin on top there is little expectation that in practice, only but the most significant motion of the hands will be detectable.

2 IMPLEMENTATION

2.1 Sensor Placement

In this design I chose to use three flex sensors, placed at different locations on the arm in order to capture different types of arm motion as well as the motion of muscles/tendons under the arm. The three muscles selected are shown in Figure 1, the Flexor digitorum superficialis, Brachioradialis, and the Extensor carpi ulnaris. These three locations in were chosen through trial and error in attempting to find three places on the arm which had the most independent readings when gestures were performed, as well as having the least interference from movement in the upper arm. It should be noted that the sensors are placed on the surface of the skin, which does not necessarily mean that during use they will stay over the targeted muscle.

2.2 Embedded System

The first stage of implementation for the project was to collect readings from the flex sensors and send them to a computer. Flex sensors change their resistance when bent, so readings were taken

Conference'18, November 2018, Tokyo, Japan

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/example

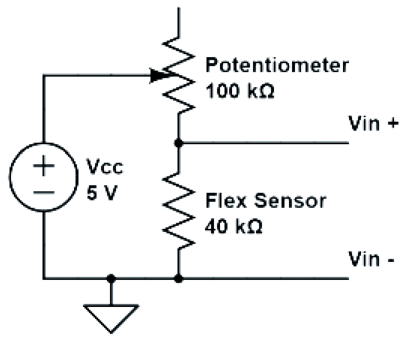

Figure 2: Circuit diagram of flex sensor voltage divider, Arduino analog reads from Vin+ and supplies the 5V voltage.

by measuring the voltage across the sensors with an Arduino Uno once every millisecond. In order to combat noise in the readings the Arduino then sent the readings forty times a second over a UART connection to the PC, taking the average of the last twenty-five readings each time. As seen in Figure 2, a potentiometer was used in order to shift neutral resting position of the sensors so they read approximately 2.5V, this was to allow for shift in either the positive or negative directions during use.

2.3 Signal Processing

A helper program on the PC side is responsible for reading in the values over the serial connection, it is this program that does the processing in order to differentiate gestures. As the data is read in it is stored in a array used as an first-in-first-out queue long enough to contain any recent motions from the user. In order to recognize a gesture the program first has to be able to recognize when any gesture occurs, this is done by looking at the standard deviation over a recent window; when the standard deviation exceeds a threshold, all values in the window are flagged as well as any new reading until the standard deviation falls below the threshold. Once the motion is over and the flagged block of reads leaves the recent window, the block becomes a sample and the system attempts to classify it.

With a sample waveform acquired I entered the crux of the problem, how to classify a waveform. Due to personal interests in Deep Learning, I initially decided to implement a multi-input neural network to take the readings from each sensor channel and have the network classify the waveform. For this implementation I used tensorflow through the Keras API, as setup the network as follows, three inputs layers, each flowing into a convolutional layer, then a recurrent layer, before concatenating the layers and passing it through several more recurrent and dense layers before a six output dense layer, one for each of the five Myo Gestures and one for a neutral pose. The model was trained using RMSprop with categorical cross entropy as the loss function over 600 samples 100 for each of the classification categories. Unfortunately despite several changes to the network architecture and hyperparameters, the network was never able to achieve an accuracy of greater than 25.1% which was unsatisfactory. I believe the main reason for the failure of the network is its inability with just convolutional and recurrent layers to appropriately categorize gestures performed at different speeds or occurred at different position in the sample window or due to natural noise in the sample. The network also became a problem as it relied on pre-sampled data that would not be representative of every user.

In order to combat these issues several things where implemented, the first being a low pass filter, the low pass filter implemented that was a Butterworth filter applied to the entirety of the queue. The Butterworth filter was selected for its relatively fast speed and its presence in the sklearn python library. Once the waveforms were de-noised, the second thing implemented was to convolve each of the sample waveforms with [-1, 1] in order to capture the relative changes in the waveforms rather than their absolute changes making them positionally invariant, this has the same intent as the convolutional layer from the neural network. The last thing implemented was a dynamic time warping comparison against several reference waveforms of gestures taken in various positions. This was not originally intended to replace the neural network, however due to its surprising effectiveness at recognizing the sample waveforms I ended up having the system classify the sample to the gesture will this closest euclidian distance to the sample. While effective dynamic time warping is an expensive operation, luckily for the purpose of my research a reasonable accuracy was reached with only two reference comparisons per gesture.

3 RESULTS AND DISCUSSION

The accuracy results shown in Table 1 show that by utilizing only two example waveforms for each gesture, the system is able to classify with average of 90.0% accuracy across the first five gestures. These gestures were selected because they are the same as the Myo Armband and there these results can be used for a direct comparison against the capabilities of that premium device[3]. With the system as designed, it is capable of altering its current classifications simply by swapping out the reference sets for a new gesture. For example with the locations of the sensors they are they can classify the twisting of forearm to the left and right with an accuracy of 98 and 97 respectively.

3.1 Future Work

The single most expensive operations in the program are the dynamic time warp comparisons, with three signals from each sensor, two reference samples, and five classifications, the program is performing 90 dynamic time warp operations for each classification. While the program can still run in real time, there is a noticeable hiccup in the run time of the program when a classification occurs. In my implementation of the helper program these are run sequentially due to python poor concurrency support, however in future implementation these operations should be performed concurrently.

The current state of the project relies on an embedded system passing data over UART to to a helper program running on the PC side, as the intent of this system is to be used in real time mixed reality applications, isolating the performance cost to an device outside the system would be essential. An ideal system for a

Table 1: Results of accuracy testing for each gesture, classified from two example gestures each.

Gesture	Wave Left	Wave Right	Double-Tap	Fist	Fingers-Spread	Twist Left	Twist Right
Correct	87	86	95	100	82	98	97
Total	100	100	100	100	100	100	100

future implementations should use a device akin to a Raspberry Pi utilizing an attached chip to perform the analog reads. This would allow the Pi to perform all of the calculations separate from the PC. Pis also have on-board GPUs which would allow for concurrent implementations of many operations currently used.

4 CONCLUSION

The research show here demonstrates that performing gesture recognition in real time using flex sensors is not only possible, but relatively simple through the use of dynamic time warping to classify the input waveforms. Through the additional implementation to allow for handling concurrent evaluations of incoming signals, significantly more robust results as well as larger gesture sets are certainly possible. Additionally the material requirements for this device is remarkably cheap, with the combined material cost using off the shelf components of less than $50, which is $150 cheaper than the comparable Myo armband.

REFERENCES

[1] [n. d.]. Flex Sensor 2.2". https://www.sparkfun.com/products/10264
[2] [n. d.]. Myo Gesture Control Armband. https://www.myo.com/
[3] [n. d.]. What gestures does the Myo armband recognize? https://support.getmyo.com/hc/en-us/articles/202647853-What-gestures-does-the-Myo-armband-recognize-
[4] 2018. Forearm Anatomy Bones The Muscles Of The Arm And Hand - Anatomy-Medicine. https://humanbodyanatomy.co/forearm-anatomy-bones/forearm-anatomy-bones-the-muscles-of-the-arm-and-hand-anatomy-medicine/