CPSC 490: Assessing Tremor Severity using the Apple Watch

Derek Lo

November 11, 2016

Advisor: Professor Ruzika Piskac, Computer Science
Additional guidance: Dr. Elan Louis, Yale School of Medicine
Joseph Schlessinger, Yale School of Medicine

Abstract

An estimated 2.2% of the US population suffers from essential tremor and is known to be one of the most frequently reported movement disorders[1]. Essential tremor is a nerve disorder that causes rhythmic shaking in people's arms, hands and sometimes even vocal cords. Currently, doctors in the clinic assess the severity of essential tremor qualitatively by examining by eye the frequency and amplitude of the tremor. Using existing sensors, we seek to develop software that can assess tremor severity objectively and precisely. This has applications both in the clinic and also in the field. Doctors can use the software and sensor device to quantify more exactly the degree of severity in their patients' tremors. In the field, the system could ideally gather continuous data from the patient, which would allow patients to understand patterns in their tremors. Such continuous monitoring could be used to better determine the effects of anti-tremor medications such as the commonly administered propranolol and primidone.

Design Considerations

Our first consideration was in determining what data would be relevant to this problem. To do that, we consulted with Dr. Elan Louis, the Chief of Yale Medical's Division of Movement Disorders, to determine what information might be prevalent to our analysis. We proposed various data that we could acquire, including location data, accelerometer data, location data and heartbeat rate. He informed us that accelerometer data would be the most predictive of tremor severity and pointed us towards existing literature on the subject.

Now that we knew what data would be useful to collect, we began the process of finding a suitable sensor. After researching various devices that collect the relevant sensor information, such as the Fitbit, Apple Watch and smartphones, we chose to proceed using an Apple Watch. Although the Fitbit does collect accelerometer data, it was not accessible through their API. A smartphone originally seemed plausible, but we realized it could be impractical for a person to hold a smartphone during a tremor. Also, a smartphone would not be able to capture continuous accelerometer data if it was in a patient's pocket. Therefore, we chose the Apple Watch because of it's accessible API and practicality as a collection device.

---

[1] Louis, Elan D., and Ruth Ottman. "How Many People in the USA Have Essential Tremor? Deriving a Population Estimate Based on Epidemiological Data." Ed. Julian Benito-Leon. *Tremor and Other Hyperkinetic Movements* 4 (2014): 259. *PMC*. Web. 11 Nov. 2016.

As of November 11, 2016 all apps created for the Apple Watch must also have a corresponding app on a paired iPhone. Thus it was necessary for us to create two apps: one on the Apple Watch and one on an iPhone. The app on the Apple Watch would collect the accelerometer data from the watch, perform any preprocessing necessary and then send this information in real-time to the iPhone.

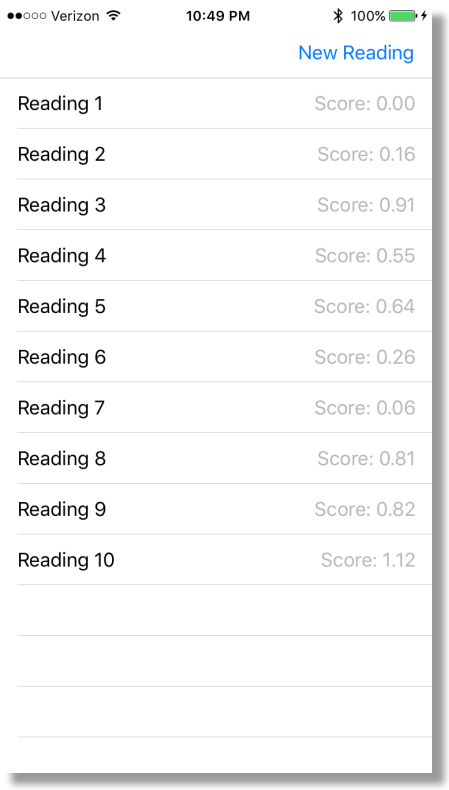
Deliverables

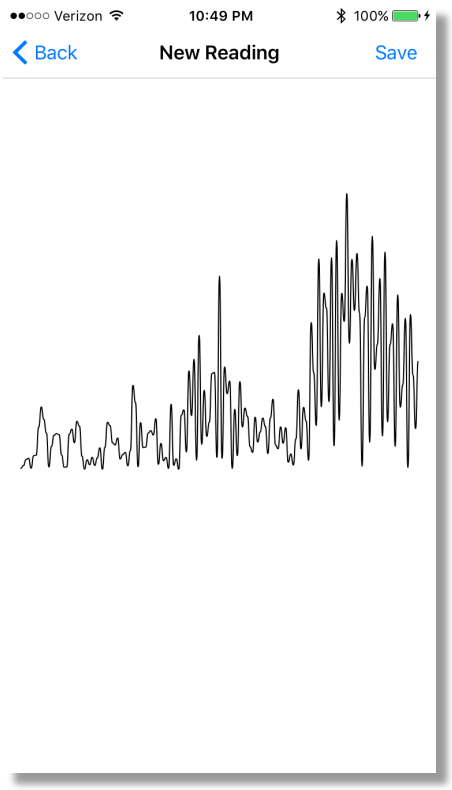
Over the course of the semester I developed the two apps mentioned in **"Design Considerations"**. We name the Apple Watch app the Tremor Reader and the iPhone app the Tremor Keeper. There is also acceleration data from four trials using the software.

Interface

Tremor Reader

Tremor Keeper





Product Usage Instructions

First the patient puts on the Apple Watch and foregrounds the Tremor Reader. The patient also foregrounds the Tremor Keeper, since the Tremor Reader will not run unless its companion app has been foregrounded.

The Tremor Reader, running on the Apple Watch, first presents the patient with a screen that displays a large “Start” button in pink. When the patient decides he or she is ready to start a reading, the patient holds out his or her left or right arm and presses “Start”. The Tremor Reader then transitions to a screen that displays a countdown timer. The timer currently starts at 20 seconds and decreases continually until the timer reaches 0. During this time, the patient is holding out his or her arm while the Tremor Reader collects information from the Apple Watch’s accelerometer every .1 seconds.

Technical Details

Programming Language: both apps are written in Objective-C.

Tremor Reader

HomeInterfaceController
A straightforward, plain-vanilla Controller with a single start button that initiates a push segue to the TimerInerfaceController.

TimerInterfaceController
The main body of the software is within this Controller. It interfaces with Apple's API to read data from the watch's accelerometer and sends that data in real-time to the phone.

```
82  - (void)startTrackingAccelerometerData {
83      if ([WCSession isSupported]) {
84          WCSession *session = [WCSession defaultSession];
85          session.delegate = self;
86          [session activateSession];
87      }
88
89      [self.motionMgr setAccelerometerUpdateInterval:0.1];
90      [self.motionMgr startAccelerometerUpdatesToQueue:self.motionQueue withHandler:^(CMAccelerometerData *
            accel, NSError *error){
91          self.acceleration = accel.acceleration;
92          if(!error) {
93              NSArray *dataPoint = @[
94                                     [NSNumber numberWithDouble:self.acceleration.x],
95                                     [NSNumber numberWithDouble:self.acceleration.y],
96                                     [NSNumber numberWithDouble:self.acceleration.z]
97                                     ];
98              [self sendDataPoint:dataPoint];
99              [self.data addObject:dataPoint];
100         } else {
101             NSLog(@"Error: %@",error.localizedDescription);
102         }
103     }];
104 }
```

Tremor Keeper

DashboardViewController
After each session the data is store in Tremor Collector using NSUserDefaults. The score computed is simply the average of all of the acceleration points.

GraphViewController

To graph the accelerometer data, we use an iOS-based charting library called JBChartView.

Results

After developing the system, I tested it on myself. I present four different trials below. The top left is a still arm with no tremor. In the top right I simulate a very weak tremor. In the bottom left I simulate a stronger tremor, and lastly in the bottom right I try to create a very strong tremor.

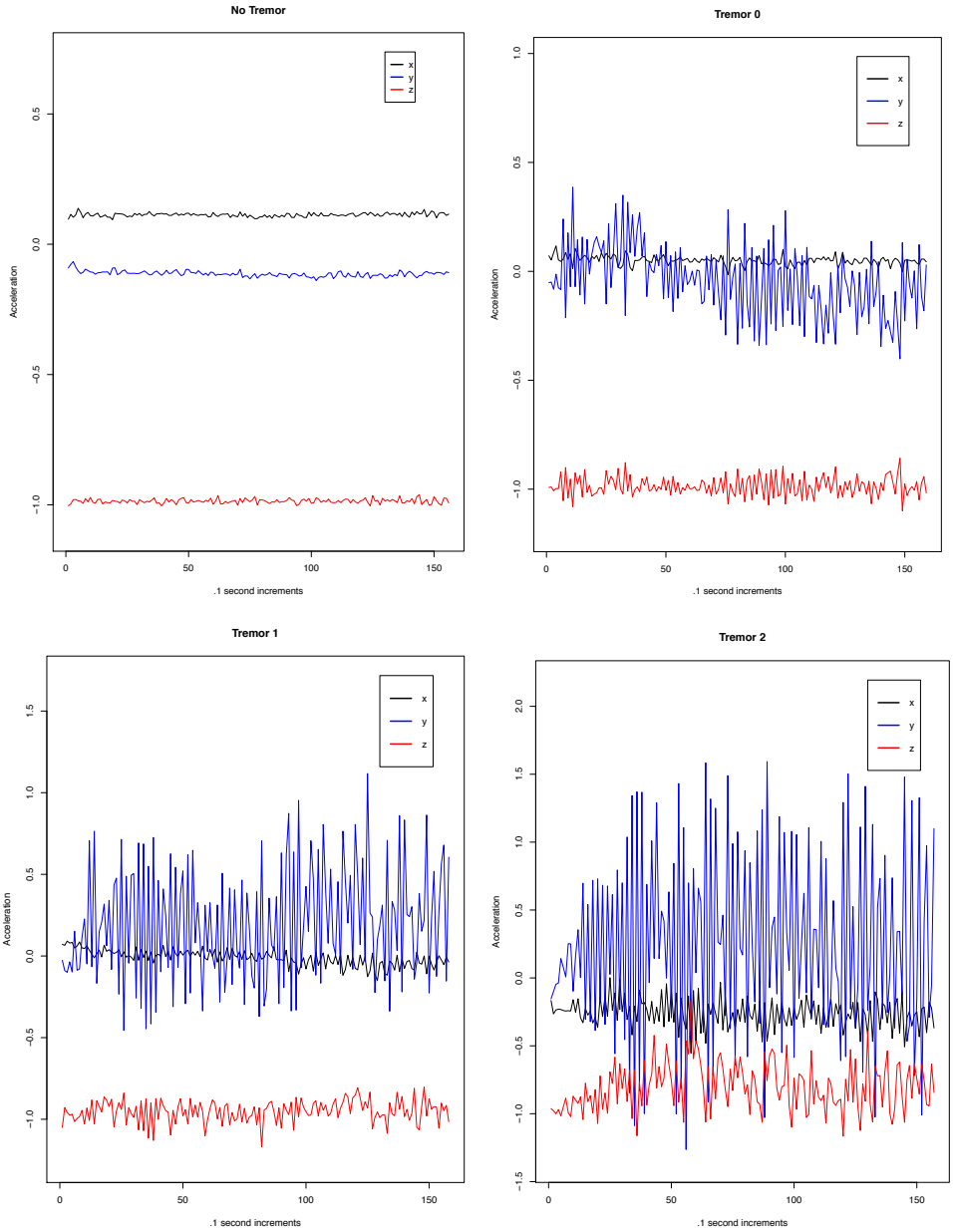


Figure 1

Further Work

We would like to create a system that can continuously take in acceleration data throughout an entire day. And also a feature that allows patients to manually enter a tremor occurrence. This would allow us to develop more robust statistical methods to detect tremors. We could use the manually entered data to train a machine learning model, for example. Finally, it would be ideal to test this system on real patients.

Acknowledgements

Ruzica Piskac: thank you for advising me throughout the semester, providing encouragement and useful feedback from a software engineering point of view.

Elan Louis: thank you for meeting and explaining the

Joseph Schlessinger: thank you for connecting me with Dr. Louis!

Sahand Negahban: thank you for agreeing to advise me in an independent study to analyze the data collected from this system.