Gesture Recognition: Final Project Report (CS698F, Group 1)

Ashwin Shenai (180156, ashwins@iitk.ac.in) Ishanh Misra (180313, imisra@iitk.ac.in)

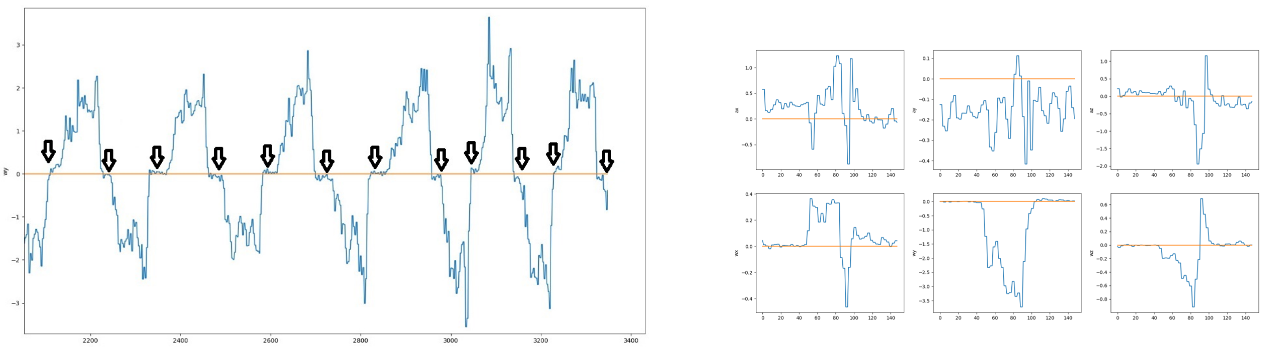
1 Introduction

We worked on the problem of recognition of handmade gestures using mobile phone in this project. Existing technologies like PhonePoint Pen described in [1] used accelerometer to write simple short messages/diagrams in air, but lacked the use of gyroscope. In particular, [2] inspired us to do this project since it tackles coping with background noise using a moving average. For displacement computation, it uses double integrator model, but requires setting small velocities to 0 periodically.

Our project aims to work on only a set of given recognizable gestures (palm going forward/backward, rotation palm, etc.). This is done to focus on remote operation of mega-machines (cranes, firetrucks, etc.) from outside their cabin. Our project aims incorporation of both accelerometer and gyroscope. In our project, we initially focused on collection of data for a given set of hand gestures. These are manually split into multiple signatures of a given gesture (called "template"). We explore multiple versions of Dynamic Time Warping (DTW) algorithm to match a test signal with templates, and compare their runtime performance. Multiple versions of distance metric used in DTW is also explored. Later in the project we did improvements on collected dataset so that our results could be much more refined. We then experiment with SoftDTW (DTW's continuous version), first proposed in [3], and use its existing implementation in [4, 5] to speed up our computation.

2 System Design

The first part of the project was data collection. For this we used our mobile phone and the application "Physics Toolbox Sensor Suite" available at https://play.google.com/store/apps/details?id=com.chrystianvieyra.physicstoolboxsuite Note that throughout our project, we use mobile phone for both data collection and inference give the online nature of this semester. For each of the gestures[1] { forward+backward, wave, cross, flip+flop, bow_up+bow_down }, we recorded 15-20 iterations continuously on the app. Note that initially our focus was on first three gestures only, so we collected only accelerometer data (total 3 dimnesions) for them, while for the rest, we collected both accelerometer and gyroscope data (total 6 dimensions). The data for each gesturewas plotted, and points where splits were required were manually identified. The splittings were put in a folder named "`raw_templates`". You can see this process as follows:



(a) Pattern observed only in gyroscope data w_y

(b) One particular signature after splitting at black arrows

This way we formed folders for { forward, backward, wave, cross, flip, flop, bow_up, bow_down } each of which contained 10-20 CSV signal samples. For experimenting with SoftDTW, we used this data as it is (the latter four sets used for this). However, for mainly applying DTW (the former four sets used for this), we needed to improve this data further. We used four techniques for the same[2]: Downsampling, Polynomial Interpolation, Wavelet Denoising, Low Pass Filtering. These refined our data to much extent. After data got refined, we compared several variants of DTW: basic DTW, Sakoe Chiba DTW, Itakura DTW, Fast DTW (SoftDTW not included since that was only experimental, see [6]). Basic DTW is the vanilla version taught in class. Sakoe Chiba DTW considers only a band about the diagonal of the DTW graph while Itakura DTW considers a parallelogram in which the DTW must be performed. Both of them restrict DTW to a smaller area, even if the theoretical asymptotic time complexity remains same. Also, parameters for both need to be tuned. Fast DTW approximates DTW in linear time and space by recursively applying DTW at different granularities (divide-and-conquer approach). For more details, please refer to our slides for better theoretical understanding of DTW variants. Note that in each of these

[1] Here "+" means that movements are opposite to each other. Also note that, what each gesture looks like has been shown in demonstration.

[2] Please refer to slides where this is explained, avoided here due to space constraints

variants, we have used Cosine Metric for distance calculation between two signal values (see eq 1). Note that cosine is preferred over Euclidean distance since the former better accounts for angle between the vectors too. We have chosen the usual value of $scale = 0.5$.

$$d_{ij} := \left(1 - scale \times \frac{< x_i, y_j >}{||x_i||_2 \cdot ||y_j||_2 + 10^{-6}}\right) \times ||x_i - y_j||_2 \tag{1}$$

The realtime gesture recognition was demonstrated successfully. Most of our time went in improving data and exploring faster DTW variants. To solve for gyroscope problem, we used SoftDTW, and ran tests for last four templates. We also tried building a server for realtime tests, but did not achieve much there.

3 Evaluation

The various pre-processing techniques to improve data yielded the following plot. Each pre-processing technique has its own comparison plot, which can be found in presentation due to space constraints here. Also, variants of DTW gave the following tabular results (clearly, FastDTW and Sakoe Chiba perform better, and downsampled data takes much less time) in table 1.

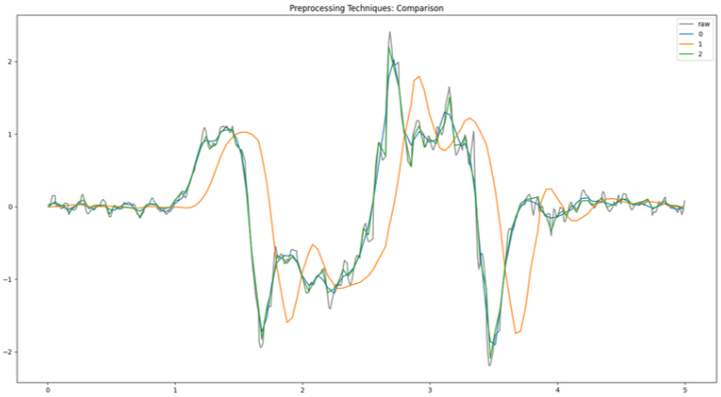


Figure 1: Progressive Pre-Processing (0: Polynomial Interpolation, 1: Low Pass Filter, 2: Wavelet Denoising)

DTW Variant	Length of Signal X	Length of Signal Y	Average Runtime (s)
Vanilla	753	1154	7.99
Vanilla (DS)	188	289	1.45
Sakoe Chiba	753	1154	3.43
Sakoe Chiba (DS)	188	289	0.285
Itakura	753	1154	5.61
Itakura (DS)	188	289	0.387
FastDTW	753	1154	0.266
FastDTW (DS)	188	289	0.069

Table 1: Comparison Results ("DS" means downsampled data)

In case of SoftDTW, we were able to perform 24800 DTW operations in 102.4 seconds, averaging to 4.13 ms each, which is much faster than all above variants.

4 Individual Contribution

- Ashwin Shenai (50 %): Recording translation data, Sakoe Chiba & Itakura DTW, Downsampling, Polynomial Interpolation
- Ishanh Misra (50 %): Recording of rotation data, basic DTW and SoftDTW, Wavelet Denoising, Low Pass Filtering

All work was done in close collaboration. Code also available at: https://github.com/Ishanh2000/CS698F (private repository). Please contact us if there is problem in running code.

References

[1] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, (New York, NY, USA), p. 15–28, Association for Computing Machinery, 2011.

[2] M. Liu, "A study of mobile sensing using smartphones," *International Journal of Distributed Sensor Networks*, vol. 9, no. 3, p. 272916, 2013.

[3] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 894–903, PMLR, 06–11 Aug 2017.

[4] M. Maghoumi, E. M. T. II, and J. J. L. Jr, "DeepNAG: Deep Non-Adversarial Gesture Generation," 2020.

[5] M. Maghoumi, *Deep Recurrent Networks for Gesture Recognition and Synthesis.* PhD thesis, University of Central Florida Orlando, Florida, 2020.

[6] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.