UNI FREIBURG

Tennis Gets Smart:

Machine Learning Based Tennis Personal Coach

SS 2017

Wearables Computing Laboratory

Embedded Systems chair, University of Freiburg

Amir Kallel | Alexander Kozhinov | Suresh Dharani Parasuraman

09.08.2017

Table of Contents

09.08.2017

1 Introduction

Tennis is a fun game to play, however quite difficult to get better at: a Tennis club membership is usually the only option for ambitious players who want to level up their game. This comes with a big cost and time commitment.

Fortunately, a smart racket may replace the effort of a personal trainer at the club as well as the pricey bill that follows. This project serves as proof of concept as an alternative to a personal coach while training, where it guides the player via live feedback of his tennis moves in order to develop a proper technique.

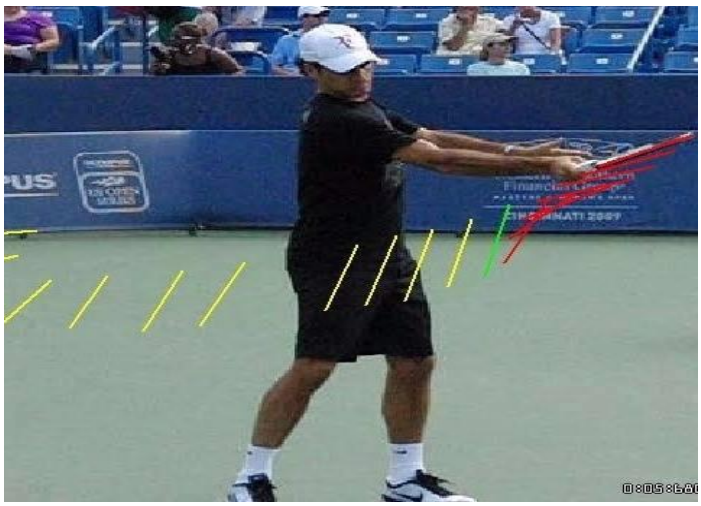


Figure 1 : Professional tennis player swing motion illustrated [1]

2 Theory and Methods

2.1 Tennis swing modeling

A parametric model has to be defined to model the correct swing motion. 93 data set was recorded while performing swing motions. This data is then plotted and analyzed using MATLAB and modeled using the curve fitting toolbox plugin with different fit types: Polynomial, gaussian etc. The most appropriate fit type determined is the gaussian fit (see equation 2.1) due to the close resemblance to the data recorded as seen in figure 5. The distribution of fitted Gaussian parameters (σ, A, μ, b) assumed to be Gaussian distributed over different data records with zero mean noise - which satisfies LDA assumptions [2].

$$y = Ae^{-\frac{(x-\mu)^2}{2\sigma^2}} + b \text{ [3]} \tag{2.1}$$

Where:

A : Amplitude

μ : Mean representing the amplitude of the data recorded.

σ : Standard deviation representing the duration of the swing recorded.

b : Bias

5 out of 9 sensor data channels were modeled using 1st and 2nd gaussian distributions as shown in the following table:

Table 1: Fit types and features of sensor channels models

Sensor channel	Fit type	Features	Number of features
X Acceleration axis	2nd Gaussian	$A_1, \mu_1, \sigma_1, A_2, \mu_2, \sigma_2, b$	7
Y Acceleration axis	1st Gaussian	A_1, μ_1, σ_1, b	4
Y Rotation speed axis	1st Gaussian	A_1, μ_1, σ_1, b	4
Z Rotation speed axis	2nd Gaussian	$A_1, \mu_1, \sigma_1, A_2, \mu_2, \sigma_2, b$	7
β Euler angle	1st Gaussian	A_1, μ_1, σ_1, b	4
Total features			26

In total, 26 parameters define the model obtained. For dimensionality reduction purposes, Linear Discriminant Analysis (LDA) algorithm is used on the 93 recorded data set to obtain a linear classifier that is later used to predict goodness of swing motion.

The peak of the acceleration X-axis of all 93 recordings were aligned on the time domain by centering the peaks at the middle of the window length of 141 points corresponding to the swing motion speed and slicing the rest of the data points. Finally, a label matrix $y_{1 \times 93}$ is created and populated with the correctness of the recorded swing (±1).

2.2 Training and simulating prediction

Once data is preprocessed, the training part takes place. The Linear Discriminant Analysis (LDA) is adopted to generate a linear classifier that is simulated for accuracy determination.

2.2.1 Creating Prediction Framework

The prediction framework loops through all 93 datasets and fit them with their respective fitting method as shown in table 1. Once the model parameters vector of a certain data are determined $X\_i_{1 \times 26}$ where 26 is the model features

number, it is populated in the LDA training matrix $X_{93\times 1}$ where 93 is the datasets number:

$$X\_i_{\,1\times 26} = [A_1\,, \mu_1, \sigma_1, A_2\,, \mu_2, \sigma_2, b, A_1\,, \mu_1, \sigma_1, b, A_1\,, \mu_1, \sigma_1, b, A_1\,, \mu_1, \sigma_1, A_2\,, \mu_2, \sigma_2, b, A_1\,, \mu_1, \sigma_1, b]$$

$$X_{93\,x\,26} = \begin{bmatrix} X\_i \\ \dots \\ X\_n \end{bmatrix}$$

The training matrix is then randomly splitted to 75% of training data X_train and 25% as testing data X_test where the testing accuracy was around 80% on records within predefined window length (see 2.1).

The classifier created is the weighting vector $W_{1\times 26}$ along with the bias b_0.

2.2.2 Simulating LDA Prediction

The simulation is proceeded as follow: it loops through each dataset, moving with a window of length L=114 data points, and fit according to table 1 and the $X\_i_{\,1\times 26}$ vector is populated.

Using the $X\_i_{\,1\times 26}$ vector, the weighting vector $W_{1\times 26}$ and the bias b_0, the prediction can take place as follow:

$$y = W_{\,1\times 26} * (X_{i\,1\times 26})^T + b\_0$$

Where y determines the prediction score (±1).

A minimum threshold is set on the acceleration X-axis to increase the detection robustness of the algorithm. Once the threshold is exceeded and $y = +1$, then a swing is predicted, else the prediction score is -1.

2.3 Live Swing detection

Once the linear classifier is determined, it can be integrated in the program that collects, plots and fits live data coming from the racket as illustrated in figures 4 and 5.

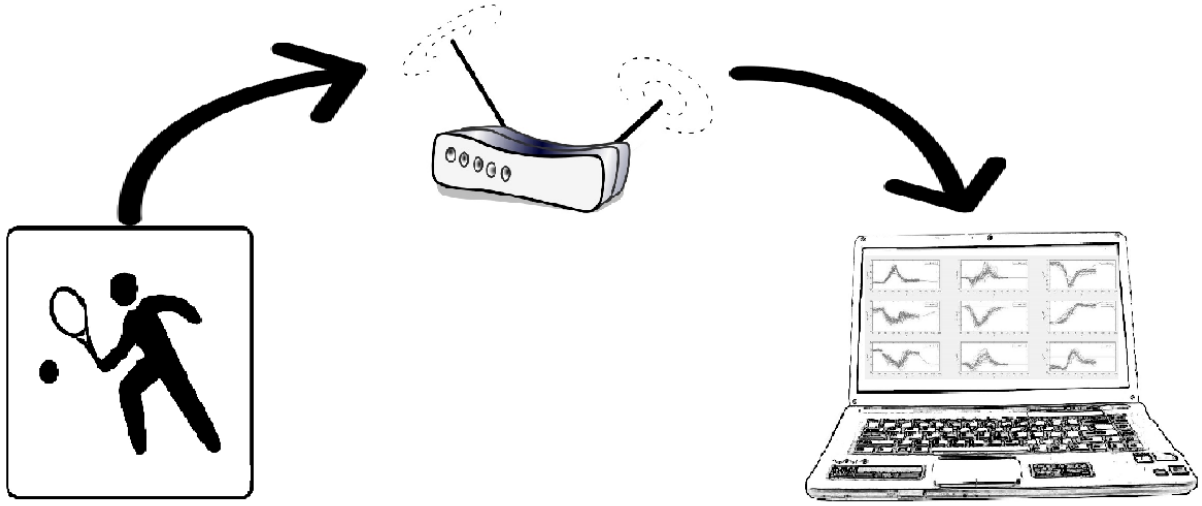
A minimum Threshold value for the X acceleration axis (see figure 4) can be configured on the program's interface to launch the fitting algorithm once a motion has exceeded that threshold. The goal is to enhance performance via fitting only when a suspected swing motion is detected.

Once the fitting algorithm is launched, it builds a window where the minimum value lies in the middle with half of the fit window length (141 datapoints as seen in figure X) on each side: 70 points on each side of the minimum value. This approach ensures correct modelling of the motion recorded. If the fit modeled is classified as a swing motion, the fit is plotted in green, otherwise the fit is plotted in red.

3 Project Setup

The architecture model of the project is client-server as illustrated in the following figure:

Figure 2: Client-server architecture model approach used in the project



Client Side	Server Side
• Reading IMU's sensors' values • Broadcasting sensors' data	• Live data plotting • On-the-fly classification of data to detect swings • Display game statistics

3.1 Hardware

An MPU-9150™ 9-axis motion tracking device from invensense® [Gyroscope / Compass / Accelerometer] is embedded in the racket on a 3D printed platform along with an Intel Edison serving as a sensor data broadcaster through a client application written in C++ as shown in the following figure:

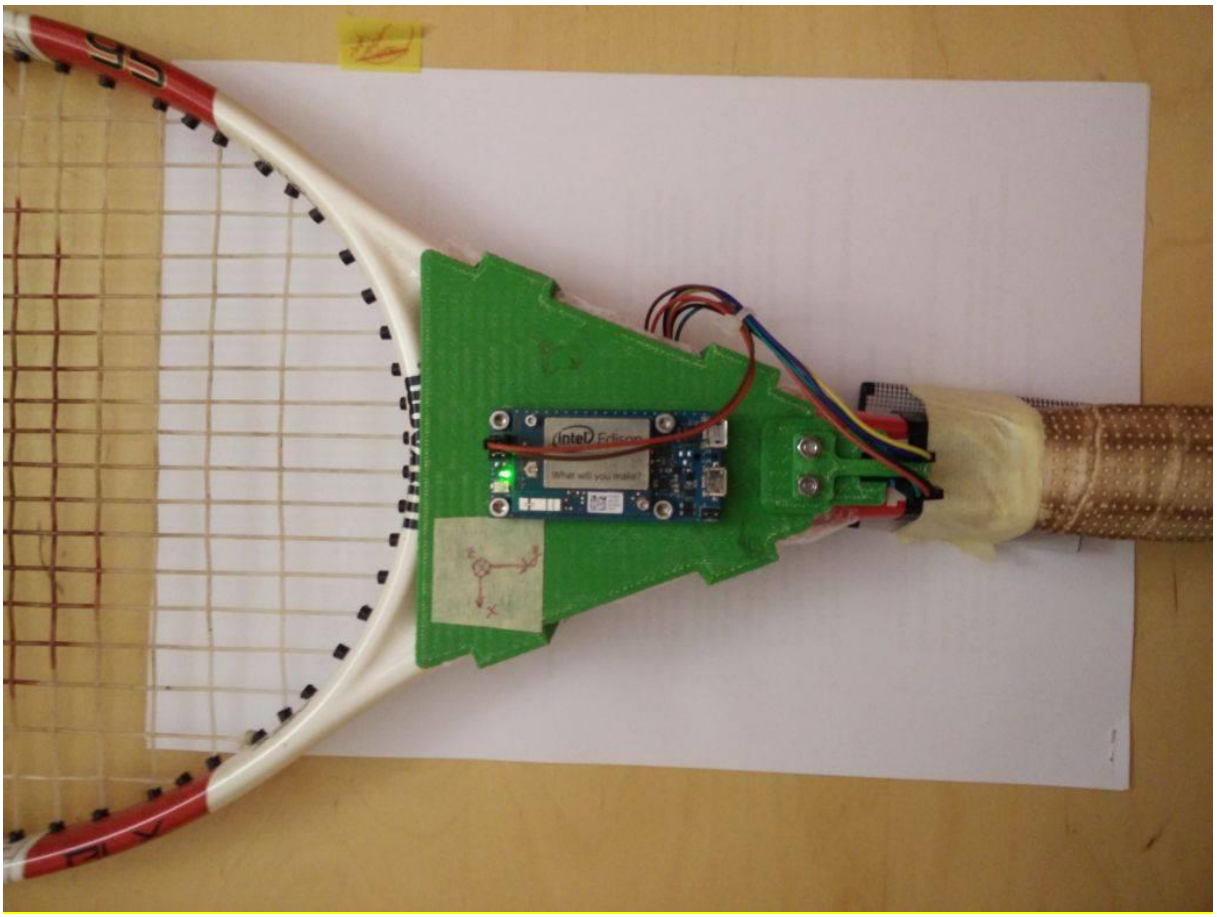


Figure 3: Racket hardware setup

The server side application is written in C++/Python running on QT. It may run on a Linux or Mac OS running machine with the following recommended requirements:

The script used to determine the swing model is written in C++/Python. It may run on a Linux or Mac OS running machine

3.2 Software

Flowchart to be inserted:Suresh input.

4 Results

The following are screenshots of the server side application while running:

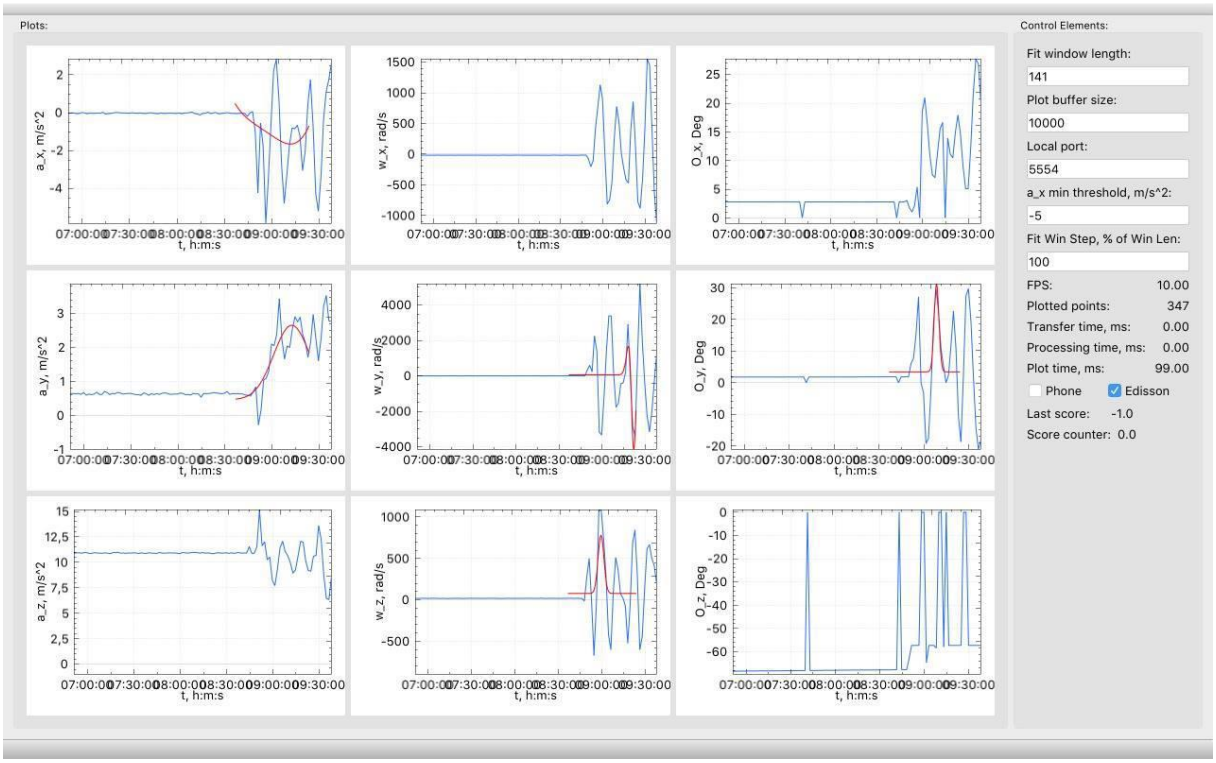


Figure 4: Data plotted and predicted correctly as a wrong swing move

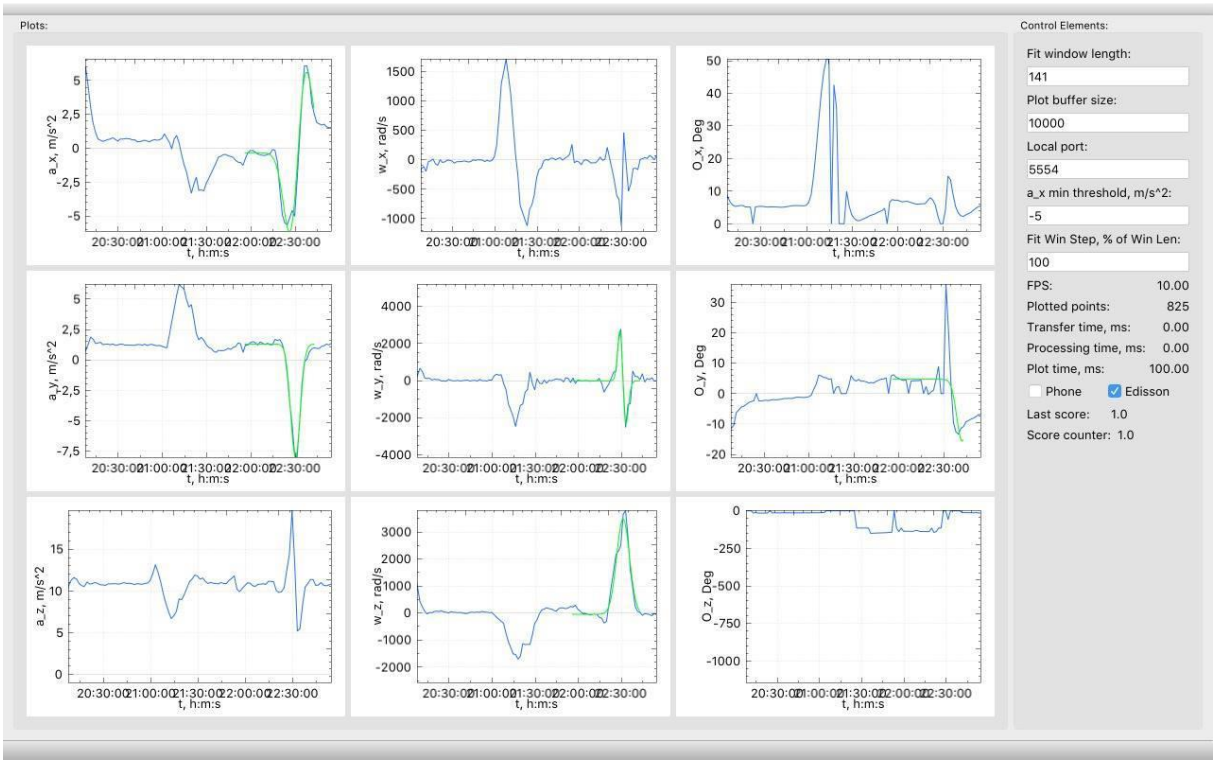


Figure 5: Data plotted and predicted correctly as a wrong swing move

Control elements are explained in the following table:

Table 2 Control elements of the server user interface

Control Element	Description
Fit window length	The number of data points of the desired window to be fitted. The value is defined manually from correct swing motion data
Plot buffer size	Determines how many points are plotted on the x axis of the plot.
Local port	The port number to listen to
a_x min threshold	The minimum Acceleration X axis threshold that once exceeded, the fitting algorithm is launched. (Explained in section 2.2)
Fit window step % of window length	Percentage of the window length to be shifted. Default is 100%
Phone/Edissoon checkbutton	Determines whether the server is listening from a smartphone sensor broadcaster app or from the Intel Edisson

5 Conclusions

The challenge addressed in this project revolves around reducing the dimensionality of swing model defined: LDA algorithm is used to easily classify 26 features of gaussian distribution models representing 5 sensor data channels by creating a linear classifier used to seperate good swing motions from other motions.

The system can be improved through the following:

- Record more data for the training set for better swing prediction accuracy.
- Train a classifier with nonlinear characteristics.
- Better signal filtering.
- Train the system automatically while playing.
- Qualitify swing motions by comparing it to the training set.
- Provide live feedback along with game statistics and estimated calories burned through a smartwatch

6 References:

[1] Navsharan Singh. (2014, Juin 14). How do you put spin on a tennis ball? [Blog post]. Retrieved from http://www.quora.com/How-do-you-put-spin-on-a-tennis-ball

[2] Jonathan Richard Shewchuk. Introduction to Machine Learning, University of California, Berkley. Retrieved from https://people.eecs.berkeley.edu/~jrs/189/lec/07.pdf

[3] Courtney Taylor. (2016, August 30). Formula for the Normal Distribution or Bell Curve. Retrieved from http://www.thoughtco.com/normal-distribution-bell-curve-formula-3126278

09.08.2017