FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Musical Instrument Learning Feedback Through Hand and Finger Movement Tracking

Diogo Amorim Cepa

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Luís Paulo Reis

Co-Supervisor: Eduardo Magalhães

July 25, 2019

Musical Instrument Learning Feedback Through Hand and Finger Movement Tracking

Diogo Amorim Cepa

Mestrado Integrado em Engenharia Informática e Computação

Chair: Jorge Alves da Silva

External Examiner: Artur Pereira

Supervisor: Luís Paulo Reis

July 25, 2019

Abstract

Learning a musical instrument is a highly time-consuming process and it can get frustrating, especially in the beginning phase. Sonia et al. [RSS11] defended that "playing-related musculoskeletal problems (PRMP) are common in adult musicians, and risk factors include gender, music exposure, and particularly instrument type. Emerging evidence suggests PRMP are common in children and adolescents and that risk factors may be similar". Nowadays it is getting more and more common to be a self-taught musician, and if even musicians monitored by professional teachers develop injuries self-taught ones are even more likely to develop them, as they usually don't receive any feedback on their gestures and posture.

This work aims to develop an artificially intelligent system that is capable of giving feedback on different techniques used while playing a musical instrument relying on fingers and hand tracking in order to hopefully help minimizing injuries and improve their learning process.

Hand gesture recognition methods are generally classified as non-vision based and vision-based devices [HC17]. The most common non-vision based devices are gloves that use sensors placed in a very stable way for motion capture [HC17] which may restrict movements making playing a musical instrument harder. A seemingly better option are the vision-based systems which can be divided into two groups, invasive and non-invasive. Both of them use cameras with the difference that invasive ones use some kind of form of identification placed in hands or fingers like color markers which may not affect that much the movements but could be still uncomfortable.

Using the information collected the system is expected to learn patterns of the correct gestures and techniques and compare them to new ones. For this, it was proposed to use an artificial neural network (ANN) to learn the specific gestures features and divide them into classes. After this learning phase, the network should recognize and divide the sequenced data into shorter pieces, mapping the transition movements between gestures and feed them into a Dynamic Time Warping (DTW) algorithm to align and compare those. It was also needed for the system to be able to recognize if the gestures are made correctly while playing the right notes. There are two ways of achieving this, by using computer vision [Sut14], by analyzing the sound produced and also can be done by comparing both sounds produced by the training data and test data.

In the end, was only possible to achieve the implementation without the segmentation of the data in movement transitions, but the use of the DTW was sufficient to assess the viability of the idea. By comparing the whole performance between participants, it was possible to search for a correlation between distances and level differences. Also, the alignment originated from the DTW was used to give feedback for each moment in time, resulting in a direction vector and the distance from the fingertips positions, the center of the palm and wrist position of both hands of the participants.

Resumo

Aprender um instrumento musical é um processo que consome muito tempo e pode ser realmente frustrante, especialmente na fase inicial. Sonia et al. [RSS11] defende que problemas musculoesqueléticos relacionados com práticas musicais (PRMP) são comuns em músicos adultos, e os fatores de risco incluem gênero, exposição musical e, particularmente, tipos de instrumento. Evidências emergentes sugerem que PRMP são comuns em crianças e adolescentes e que os fatores de risco podem ser semelhantes. Hoje em dia é cada vez mais comum ser um músico autodidata, e se até os músicos seguidos por professores desenvolvem lesões, os autodidatas têm ainda maior probabilidade de desenvolvê-los, pois geralmente não recebem nenhum feedback sobre seus gestos e postura.

Este trabalho tem como objetivo desenvolver um sistema de inteligência artificial capaz de dar feedback sobre diferentes técnicas usadas durante a utilização de um instrumento musical, com base no rastreamento de dedos e mãos, com o intuito de ajudar a minimizar lesões e melhorar o processo de aprendizagem.

Os métodos de reconhecimento de gestos são geralmente classificados como dispositivos não baseados em visão e baseados em visão [HC17]. Os dispositivos mais comuns não baseados em visão são luvas que usam sensores colocados de maneira muito estável para capturar movimentos [HC17], que podem restringir os movimentos e dificultar a utilização de um instrumento musical. Uma opção aparentemente melhor é a dos sistemas baseados na visão, que podem ser divididos em dois grupos: invasivo e não invasivo. Ambos utilizam câmaras com a diferença de que os invasivos usam algum tipo de identificação colocada nas mãos ou nos dedos, como marcadores de cor, que podem não afetar tanto os movimentos, mas podem se relevar desconfortáveis na mesma.

Usando as informações encontradas, espera-se que o sistema aprenda padrões dos gestos e técnicas corretas e avalie novos gestos em comparação com esses. Para isso, foi proposto o uso de redes neurais artificiais (ANN) para aprender as características específicas dos gestos e dividi-las em classes. Após essa fase de aprendizagem, a rede deve reconhecer e dividir os dados em segmentos menores, mapeando os movimentos de transição entre os gestos e alimentá-los a um algoritmo Dynamic Time Warping (DTW) para alinhar e comparar esses dados. Também era necessário que o sistema fosse capaz de reconhecer se o gesto foi feito corretamente enquanto tocava as notas certas. Há duas maneiras de conseguir isso, usando a visão de computadores [Sut14], analisando o som produzido e também pode ser feito comparando os sons produzidos pelos dados de treino e os dados de teste.

No final, só foi conseguida a implementação sem a segmentação dos dados nas transições de movimentos, mas o uso do algoritmo DTW foi suficiente para avaliar a viabilidade da ideia. Ao comparar todos o desempenhos entre os participantes, foi possível procurar uma correlação entre distâncias e diferenças de nível. Também se usou o alinhamento resultante do DTW e a partir deste foi gerado feedback. Este feedback consiste num vetor de direção e na distância das pontas dos dedos, o centro da posição da palma da mão e do pulso de ambas as mãos dos participantes, para cada momento no tempo.

Acknowledgements

This project means a lot to me as it was my idea and seeing it applied in the context of this thesis brought me a great feeling of accomplishment. This would not be possible without the help of my parents who always supported me and gave me the conditions to meet this goal until the end. I'm also grateful to my brother and sister who always were there for me.

A special thanks to Rui, who was my guinea pig for all the tests needed to conduct before the actual experiments, to my closest friends who kept my sanity during the whole period of this degree and helped me build the idea throughout the years by having stupid conversations that sometimes lead to somewhere not stupid.

Also, thank you to all the participants who gave their time and effort to be there without any obligation and made it possible to have results.

Finally, thank you to my supervisor for the good advice of using the leap motion sensor and to the professor Eduardo Magalhães who willingly joined the project and always gave honest opinions which resulted in a lot of ideas for future works and changes on the current solution.

Diogo Amorim Cepa

"If you feel blocked, do not turn to others, but look inside, in silence,
for the enemy of your progress."

Jeff Buckley

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

3D	3 Dimensional
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
DTW	Dynamic Time Warping
HCI	Humam Computer Interaction
HMM	Hidden Markov Models
PRMP	Playing-related musculoskeletal problems

Chapter 1

Introduction

When we look at music and technologies we can see that nowadays these two fields are becoming more and more related to each other. When it comes to learning a musical instrument it could be beneficial that a teacher could be 100% available to give the student feedback and, in this area, technologies could be the solution. Currently, there are a lot of examples where people tried to create applications to improve the learning curve of a musician like Yousician[1] or even games like Rocksmith[2] where musicians can see in real-time where they are making mistakes. But none of them give feedback not only on the correctness of notes and timing but also on how the student is performing the techniques.

1.1 Scope

This idea belongs to various areas in computer engineering. One of the main is Human-Computer Interaction (HCI) and Computer Graphics, more specifically Hand 3D modulation, as the user hands will be used as input for the feedback given on his movements. It also enrolls in Intelligent Systems as it will use methods from Machine Learning and Neural Networks to create an intelligence capable of learning and comparing those learn movements and input ones.

Apart from this it also contains relations to multimedia as we deal with information like sound and image in a dynamic way, processing both those signals is a necessity.

Finally, it's also inserted in the music area as the main goal is precisely the feedback of a musician's performance in comparison to the master dataset.

[1] https://yousician.com/
[2] https://rocksmith.ubisoft.com/rocksmith/en-us/home/

1.2 Problem Statement and Motivation

Not everyone has the possibility to enroll in music classes and have a personal teacher of his instrument of choice, due to lack of music schools, money or even the person's availability at the time. Those people tend to be the ones ending in the last choice of being self-taught musicians, which, as every self-taught student, end up having doubts about what they are doing and how they are doing it.

The current applications, that only help the student's learning either with providing knowledge or through sound-based feedback, can't even compare to the feedback of a teacher, who is able to not only offer the student music knowledge and sound-based feedback but also important feedback on posture and correctness of technique's gestures. It's in this late part that machine learning and human-computer interaction could come to good use. It would be amazing if one could develop an intelligent application capable of recognizing the student's gestures and give them feedback and ideally a tip on how to improve them.

This idea came in mind with the assumption that the movements performed by an intermediate musician will be more similar to the performance of an advanced musician than the movements performed by a beginner. This assumes that with experience, for a certain style of playing, people will tend to develop the same type of movements. So by showing them the changes needed to make so their movements are closer to better musicians, their skill will increase in a better way that is less conducive to the development of potential bad habits.

1.3 Aim and Goals

It's expected that this work results in an intelligent system capable of recognizing and learn specific gestures with a specific musical instrument master and compare them to the ones of a beginner giving them an estimated approximation value to the gestures of the master. All this should be done in a non-intrusive way so it does not disturb the performance of the player using the system. For these objectives, it is going to be used the Leap Motion controller as it is able to recognize the hand and fingers with precision without the need to implement a computer vision algorithm. Also needs to use dynamic gesture recognition methods, involving neural networks, to learn the specific gestures patterns and a similarity/proximity method to compare the master movements with the user.

1.4 Document Outline

Apart from this introduction, this paper contains five more chapters. Chapter 2 describes some technologies that are applicable for the proposed solution as well as an overall about the previous related works to this specific topic. Chapter 3, describes the proposed solution and how it was planned to implement it. Following it, chapter 4 shows how the solution was implemented and what was used to implement it. The description of the experiments, their results and their analysis are present in chapter 5. To closure this document, the chapter 6 will sum up the whole document elaborating conclusions on each chapter. Also, this last chapter will address the possible future work.

Introduction

Chapter 2

Literature Review

2.1 Gesture Recognition

2.1.1 Gesture Classification

As humans are able to perform an enormous number of different gestures, categorize those can result in an extensive list of classes. So to generalize this classification certain classes were adopted as common. This classification considers the different traits of gestures and was based in a variety of fields, like anthropology, cognitive science, and others. This subsection will present these generally used gestures classifications in HCI for hand recognition.

Firstly these gestures are divided into static and dynamic gestures. Static gestures contain all the gestures that don't need to take changes over time into account while dynamic gestures are the ones that take this time-varying into consideration.

Another common and general division, made by Kammer el al. [KKFW10], is classifying the gesture as online or offline. Having the online gesture mean that the gesture is processed during the performance, in real-time, while offline is only processed at the end of the gesture.

A more extensive gesture taxonomy for HCI is proposed by Karam and Schraefel [KS05] based on F. Quek el al. [QMB+02] classification. They divided gestures into deictic, gesticulation, manipulation, semaphores and sign language. Another classification more suited for hand gesture recognition purposes, which is presented by Aigner et al. [AWB+12] using concepts proposed by Karam and Schraefel. They divide gestures into five categories, as represented in figure 2.1:

- **Pointing** — action used to point to an object or indicate a direction. Based on [KS05], it's possible to say that the gesture is basically pointing with not only the index finger but any finger to tell the system the position of an object or the object itself.

- **Semaphoric** — Gesture posture and gesture's dynamics used to convey a specific meaning. Essentially, these types of gestures serve as a language of symbols to communicate with the machine [QMB+02]. Even though they are symbolic gestures their structure can be

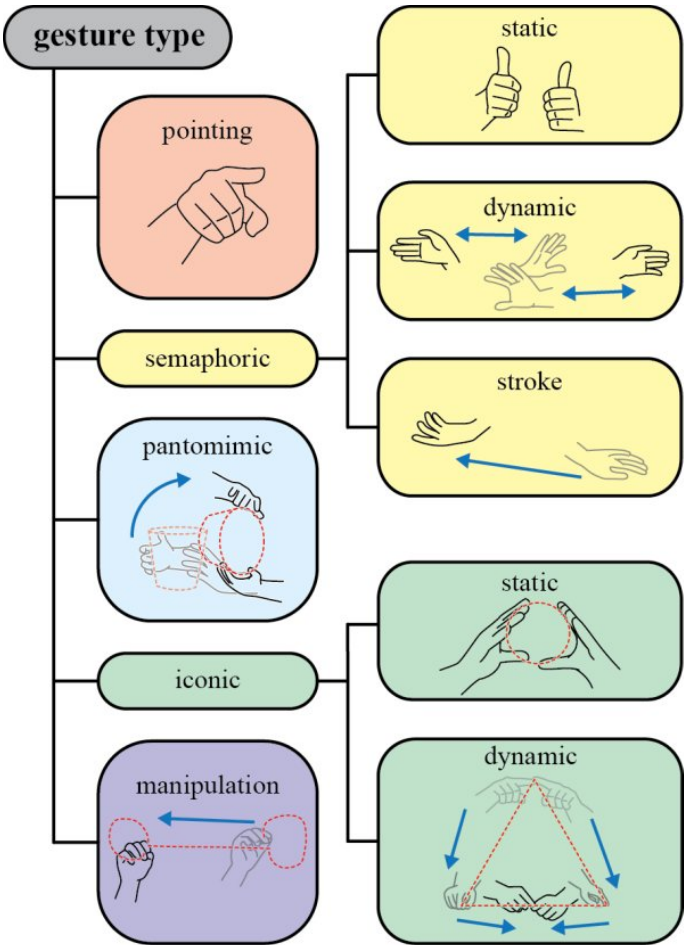

Figure 2.1: Hand gesture classification according Aigner et al.

unrelated to their actual meaning. Along with this category, the gesture can belong to three subcategories which are static, dynamic and strokes. Static represents hand postures without considering movements such as a thumbs up or an OK sign. A dynamic gesture's meaning is based on the movement of the gesture, for example, saying hello or goodbye by waving your hand. The last category is similar to dynamic movements but represents fast movements more close to stroke-like like a swipe.

- **Iconic** — Opposing semaphoric gestures, these are strictly related to their meaning. Iconic gestures are used to represent the shape, size, curvature of objects or entities. These also can be divided into static and dynamic. The first uses hand posture like making a circle or a rectangle with your index and thumb. The latter is used to represent an edge line of objects with motion paths like drawing a triangle in the air with your index.

- **Pantomimic** — Imitation of performing a specific task or activity without tools like reloading a weapon, moving an object or slicing food.

- **Manipulation** — controlling the position, scale and rotation of an object is the objective of these gestures. It's a direct interaction between the hand or tool performing the gesture and the manipulated object. Which means that the gesture is strictly connected with the movement of the object.

2.1.2 Sensors

This subsection will describe the type of sensors commonly used for gesture recognition. These sensors can be divided into two groups, non-vision based sensors and vision-based sensors.

For non-vision based sensors:

- **Wearable** — This type of devices always resemble a piece of clothing, usually gloves, armbands or full body outfits. They are often related to biomechanical and inertial technologies.

- **Biomechanical** — Takes advantage of biomechanical techniques such as electromyography to measure parameters of gestures.

- **Inertial** — they capture magnetic field variation with gyroscopes and accelerometers to estimate the type of movement.

- **Haptics** — Variety of touch devices that measure the movement through force applied or vibrations.

For vision-based sensors:

- **Video camera** — techniques used to infer hand movement from the data of a single camera using methods like color or shape-based techniques, learning detectors from pixel values and 3D model detection.

- **Multiple cameras** — Instead of one camera these techniques use multiples cameras so the input data is much closer to the real dimension, which means closer to a 3D model.

- **Depth cameras** — these devices make use of the projection of structured light, for example, infrared light. Kinect and Leap Motion are the most popular examples of this kind of device.

- **Invasive techniques** — these are techniques that require non-natural elements like color markers or even gloves or instruments.

2.1.3 Hand Representation

Representations can be 3D model based or appearance based. Each one of these classes can be subdivided in even more classes. Figure 2.2 shows this division.

For 3D model-based representations, the subdivisions are:

- **Volumetric** — this representation reproduces the shape of the hand or arm with high accuracy. They are often interpreted as a cloud of spatially ordered vertices or points. Usually

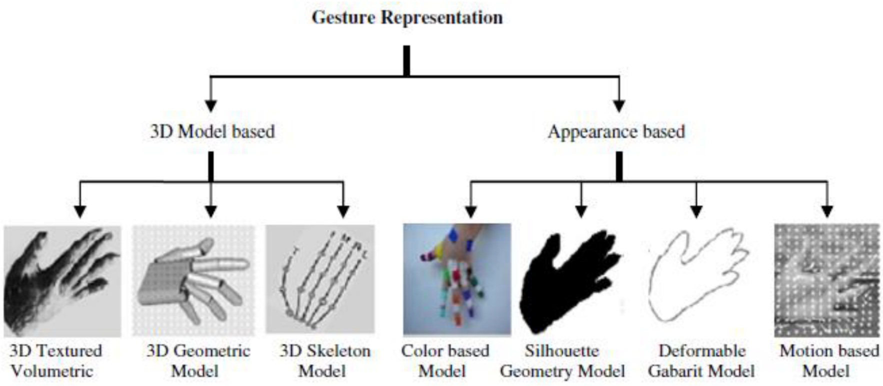

Figure 2.2: Hand representation divisions

used for computer vision or animation. But this approach is computationally heavy and hard to use in a real-time application.

- **Skeletal** — simple models of the hand, arm or even whole body with much fewer parameters than the previous one [PSH97]. The model only has a set of joint angles along with segment lengths, representing articulations and bones. This makes the program able to focus only on the important aspects of the body and because the data is much smaller the time needed to process the algorithms is shorter thus making it possible in a real-time scenario.

For appearance-based representations:

- **Color based** — uses color markers to recognize and identify specific parts of the body in order to track their motion. Bretzner et al. [BLL02] proposed this representation by using particle filtering, hierarchical models and multi-scale color features.

- **Binary silhouette** — takes advantage of the following silhouette properties: bounding box, convexity, surface, perimeter, elongation, centroid, rectangularity, and orientation. In their project, Birdal and Hassanpour [BH19] used bounding boxes for gesture recognition.

- **Deformable gabanit** — its basis is usually on deformable active contours. Ju et al. [XJBMK97] applied the active contour model, also known as snakes, to a problem of gesture recognition.

- **Motion based** — these models use the motion of an object in an image sequence to identify an object or its motion. Using an Adaboost framework to teach the models, Luo et al. [LKZF10] brought to the world the local motion histogram.

2.1.4 Gesture Recognition Methods

The recognition method should be chosen based on the aforementioned types of gesture classification.

For Static gestures:

- **Support Vector Machines** — are non-linear classifiers [Bur98]. The concept of this method is to transport data that is not linearly separable to a higher dimension so it can be separated through a linear function. This uses supervised learning, having the data previously labeled as one category or the other, so it can build a model to categorize new data.

- **Neural Networks** — Diverse types of neural networks exist nowadays. Every single one of them is based on how the human brain works. Basically, they have layers of nodes or neurons that have connections between them. So each neuron can have one or more connections to neurons of the adjacent layers. Also, they have an input layer that represents the signal received and an output layer that represents the response to that signal. The connections between neurons have weights calculated by the learning method chosen. There is even an activation function that defines if the neuron activates another neuron [XMW+18].

For Dynamic gestures:

- **Time Delay Neural Networks** — This is a specially built network to manage continuous data with an adaptive architecture so it's suited to real-time problems. These neural networks give each neuron the ability to store history of inputs, resulting in each neuron not only being able to access data at the current point in time t but also to another point in the past time t_n. It's learning is usually made with back propagation[SSA04] [WA99]

- **Finite State Machines** — According to Hofzmann [Hol25] this machine is one that has a limited number of possible states. Their training is often done offline using a big amount of gestures examples having the parameters of each state derived. Basically, when recognizing, if the network reaches a final stated it is a match otherwise it is not.

- **Hidden Markov Models** — In this case, the system is always considered to be a Markov process with hidden states from which it makes statistical models. Both output probabilities and states are hidden but the transition probabilities can be calculated based on the output [VdB17].

- **Self Organizing Networks** — These types of networks use unsupervised learning and during this learning. They update their weight vector of each neuron is found and updated by self-learning. Unlike regular ANN this one uses a neighborhood function to preserve topological properties of the input [VM13].

2.2 Hidden Markov Models

This model has become the chosen method to model stochastic processes and sequences in areas like speech and handwriting recognition, and in most cases, the topology is predetermined and the parameters are estimated by the forward-backward algorithm, also known as Baum-Welch [FST98].

The main objective of the HMM is to compute the joint distribution, represented in equation 2.1 where Z are the hidden states and X are the observed states [SMR99].

$$p(X_n, Z_n) = p(Z_1)p(x_1|Z_1)\prod_{k=2}^{n} p(Z_k|Z_{k-1})p(X_k|Z_k) \tag{2.1}$$

The HMM parameters include the transition probabilities that can be a randomized value ranging from 0 to 1 or a uniform segmentation estimating the global mean and variance from the available data. These transition probabilities are represented by the equation 2.2, where i and j is the current state. This parameter represents the probability of one state transition to another state.

$$T_{ij} = p(Z_{k+1} = j|z_k = i) \tag{2.2}$$

Another parameter group is the emission probability, shown in equation 2.3, that are re-estimated with the Forward-Backward algorithm. This parameter is the probability of going to a state being on a specific state.

$$\xi_i(X) = p(X|Z_k = i) \tag{2.3}$$

The last of these parameters is the initial distribution that usually are random values from 0 to 1, and are represented by the equation 2.4.

$$\pi(i) = P(Z_k = i) \tag{2.4}$$

So applying these parameteres to the joint distribution the equation 2.5 is acheived.

$$p(X_n, Z_n) = \pi(i)\xi_{Z_1}(X_1)\prod_{k=2}^{n} T(Z_{k-1}, Z_k)\xi_{Z_k}(X_k) \tag{2.5}$$

The Forward-Backward algorithm goal is to compute $p(Z_k|X)$ which can be separated in two steps. The Forward step has the objective of computing $p(Z_k|Z_{1->K})$, which can be achieved by the equation 2.6, being $1->K$ the states from the first until the state K.

$$\alpha_k(Z_k) = \begin{cases} \sum p(X_k|Z_k)p(Z_k|Z_{k-1}\alpha_{k-1}(Z_{k-1}), & k \geq 2 \\ p(Z_k)p(X_k|Z_k), & k = 1 \end{cases} \tag{2.6}$$

The second step is the Backward, and has the goal of calculating $p(X_{(k+1)->n}|Z_k)$ which is defined by the equation 2.7.

$$\beta_k(Z_k) = \begin{cases} \sum_{z_{k+1}}^{m} \beta_{k+1}(Z_{k+1}) + (X_{k+1}|Z_{k+1})p(Z_{k+1}|Z_k), & 1 \leq k \leq n-1 \\ 1, & k = n \ \& \ \forall Z_n \end{cases} \tag{2.7}$$

Then to retrieve the hidden state values sequence the Viterbi algorithm is used to retrieve these values [SMR99].

2.3 Leap Motion

The Leap Motion device, launched in 2013 by Leap Motion inc., is a sensor-based on vision techniques targeted to the extraction of 3D data. This device is designed focusing only on hand gesture recognition and the estimated position of several bones of the hand is native to the sensor, which means it is not needed to implement computer vision algorithms to achieve this. Compared with another depth camera, the Kinect, it produces a few key points instead of the complete in-depth representation and its optimal distance is shorter. Although the captured data is more precise and, according to a study [WBRF13], its accuracy is about 0.2mm and according to the manufacturers has less than 0.01s of latency. Instead of a cloud of points, Leap Motion gives the user a set of information for each key point, in this case, bones and articulations as well as their rotation and orientation, and this information is given for each frame.

It achieves this representation by using three infrared lights and two infrared cameras separated by 4 centimeters. Due to the dissipation of the infrared lights, the optimal distance is about 80 centimeters and its frame rate ranges from 50 to 200 frames per second depending on if it is used USB 2.0 or 3.0.

Leap Motion also provides a software that is able to recognize a few gestures more precisely circle, swipe, key tap and screen tap. It also provides an extensive API but mostly directed to virtual reality usage.

2.4 Gesture proximity/similarity

This kind of feature is achievable through various methods but, from what was found while researching, none of them is focused on the comparison of the same gesture done by two different people and were only used to make a prediction of the recognized gesture. In other words, they were used for the learning phase of the recognition itself. Among the research, the most promising method found was Dynamic Time Warping (DTW).

DTW is an algorithm that matches test data with reference data where this data is represented in a time sequence of measurements or captured features. Its goal is to align the reference and test vector sequences in time using nonlinear mapping, also named as warping, resulting in an ordered set of pairs [reference, test]. The cost of this mapping is the sum of all the Euclidean distances, between the paired reference and test elements. The test sequence is matched with the reference sequence that has resulted in the global minimum cost of all the reference sequences. This is done by a recursive method going through every point in time to check which combination of reference plus test gave the minimum distance and sums this to the already obtained sum. After this, although it's the minimum cost, to find the optimal path it is needed to backtrack starting in the final point and if the test sequence is mapped to the same reference then it's a match [Sen08].

The base of the DTW algorithm is the distance function and the recursive call that will compute the minimum sum of all the distances between each point. The distance function is usually a euclidean distance, see in equation 2.8, that gives the actual spatial distance between two points, where N is the dimension of the points.

$$d(P_1,P_2) = \sqrt{\sum_{n=1}^{N}(P_{1_n} - P_{2_n})^2} \tag{2.8}$$

The recursive call combined with the distance function originates the base DTW algorithm 2.9, which starts on the last point of the matrix of all the distance combinations. Let the matrix be an n by m matrix which will start on the [n,m] position. As a first step, it calculates the distance of the initial position, and then adds the minimum of the three adjacent cells. You can imagine this as a square matrix of size two where the current cell is on the top right, and the next chosen position will be one of the other three that has the minimum sum of the following points.

$$DTW(n,m) = \min[DTW(n-1,m), DTW(n,m-1), DTW(n-1,m-1)] + distance(R_n, T_m) \tag{2.9}$$

As this algorithm could get computational demanding was invented certain conditions to optimize the resources used. These conditions made it faster as it does not have to analyze every single point combination [Mül07]. All the conditions formulas have the i_k and j_k as the matched pair of points' index and i_s and j_s as the current selected matrix position.

The boundary condition 2.10 that makes sure the algorithm starts and finishes in the beginning and the end of both time series ensuring that the alignment does not consider partially one of the sequences.

$$i_1 = 1 \;\; \& \;\; i_k = n \;\; \& \;\; j_1 = 1 \;\; \& \;\; j_k = m \tag{2.10}$$

The monotonicity condition, seen in equation 2.11, that ensures that the alignment path does not go back in time resulting eliminating the possibility of repeating features.

$$i_{s-1} \leq is_s \;\; \& \;\; j_{s-1} \leq j_s \tag{2.11}$$

Last but not the least, the continuity condition 2.12 which guarantees that the path does not jump in time making the path go through important features.

$$i_s - i_{s-1} \leq 1 \;\; \& \;\; j_s - j_{s-1} \leq 1 \tag{2.12}$$

Reyes et al. [RDE11] proposed a method that uses weighted DTW (WDTW) for gesture recognition in a skeletal modeled body motion. Their method calculates DTW costs considering weights for each body joint enabling them to focus on more important joints. Another improved method proposed by Celebi et al. [CATA13] is to use multi-dimensional DTW (MDTW), calculating a weight for each key point as well one for each gesture class. This makes DTW costs much more discriminant by only using the information about the importance of joint to the specific gesture class. With this method, they outperformed both the conventional DTW and Reyes et al. method in a gesture recognition case.

From the above mentioned, it is possible to know that an MDTW can be calculated using 2.13 and a WDTW combined with an MDTW will result in 2.14, where M is the number of dimensions.

$$DTW(R,T) = \sum_{m}^{M} DTW(R_m, T_m) \tag{2.13}$$

$$DTW(R,T) = \sum_{m}^{M} W_m * DTW(r_m, Tm) \tag{2.14}$$

Even though in the past work found is not actually calculating a similarity it seems that it is possible to use DTW for this purpose as the core of the DTW is the calculation of signals dissimilarity.

2.5 Open Sound Control

Open Sound Control (OSC) is a communication protocol between computers, sound synthesizers and other multimedia devices that brings advantages of modern technology to electronic musical instruments, such as interoperability, accuracy, flexibility and better organization and documentation [FS09].

It provides features needed for real-time control of sound and other media processing without losing flexibility and remaining simple to implement. These features include:

- Open-ended, dynamic, URL-style symbolic naming scheme
- Symbolic and high-resolution numeric argument data
- Pattern matching language to specify multiple recipients of a single message
- High-resolution time tags
- "Bundles" of messages whose effects should be simultaneous
- Query system to dynamically find capabilities of an OSC server and documentation

This protocol was developed at UC Berkeley Center for New Music and Audio Technology, where it is still subject to ongoing research. Regardless of this, there are dozens of implementations of OSC in various programming languages and applied in multiple fields including computer-based interfaces for musical expression, distributed music systems, inter-process communication, and even communication within a single application.

2.6 Related Works

Recently there has been some research and experiments on piano and guitar playing using vision-based sensors but always reaching some kind of problems related to latency or even the quality of data collected and even the hand recognition algorithm itself.

In 2007, Chutisant and Hideo [KS07] developed a system capable of recognizing a bass player fingertips using only computer vision. This was achieved by an initial segmentation of the hand region using a Bayesian classifier as for the fingertip detection they used 2 Local linear mapping networks with Gabor-filters to separate each individual finger and then recognize the fingertip. In this case, they used images for training data and then tested with video and realized that the developed work didn't track index and little finger with much accuracy.

In 2013, Akiya and Manabu [OH13] presented a marker-less piano finger recognition using sequential depth images. The whole objective was to build a good recognition system of the piano playing so it could be used in an automated lesson. For this, they created a code combining the note names and finger names. With this code they could easily compare if the combination of the finger and note was the right one or if the beginner should use another finger or the same finger in other note and they also consider the wrist rotation. For this, they used a learning module and a recognition module. The recognition module uses the Nearest Neighbor search algorithm to compare the input images with the dictionary learned in the learning module and for the capture, they used the Kinect sensor. With this set, they achieved quite good results, on simple pieces it could give a 100% recognition rate while on a more complex piece they achieved 91.6%. This was due other to the fast tempo of the song or the occlusion of thumb finger in some movements.

More recently, in 2018, Yoshitaka and Youji [KO18] researched the use of Convolutional Neural Networks(CNN) for guitar fingering recognition also using Kinect for the capture of data. For this project they also used images as the training dataset and then compare it to check the recognition rate of each finger. They got some interesting results as they not only analyzed the left hand but also the right-hand movements and it was in the picking direction (right-hand movements) they got the best results. Even though the overall estimation of all fingers and the respective strings were around 94%, they achieved a good recognition but only on a limited part of the guitar fretboard, from 7th to 9th fret.

Not as related as the previous ones, Lu et al. [LTC16] studied the dynamic hand gesture recognition with a leap motion controller. For this they decided, against the common approaches they described, to use a Hidden Conditional Neural Field classifier to divide gestures features and recognize the different gestures. They achieved an 89.5 percent of accuracy with the leap motion sensor which they consider a satisfactory recognition.

2.7 Conclusions

Regarding the aforementioned, it's possible to assume that the best sensor for this case is the Leap Motion as it's not needed a distance higher than the optimal one and at that distance it gives better results than other vision-based sensors. It also brings the advantage of not having to implement the hand and finger recognition, as it is provided by the sensor, which means that it is only needed to make the learning of the specific gestures patterns and then a posterior comparison between new inputs and the learned data.

Differently from past works, where they almost only correlate fingertips with the note played with Kinect, this one will try to check not only the correctness of notes and fingers combination but also how those gestures are made and how to change from one to another mainly focusing on chords instead of fingerings. Also instead of comparing them to the piece in global the objective will be correctness for the type of gesture in comparison to the learning dataset, which also won't be images but real-time learning data.

Chapter 3

Methodological Approach

3.1 Introduction

Based on the aforementioned research, this chapter will describe the problem, how the problem was addressed and which technologies and methods were planned to be used.

3.2 Problem Summary

The main problem is how to give feedback to a beginner on very specific gestures used while playing a musical instrument in a completely independent manner without having human supervision.

How well can an approach of gesture comparison between experienced players and beginners improve their learning curve and prevent the creation of bad habits in their playstyle which will make it harder to learn more advanced techniques? Musicians are always inspired by the type of play of the musicians they look up to, and analyzing how they play is a major contribution to their learning and discovering new and better ways to perform the same techniques, chords or even progressions. So this approach is based on this presumption where the beginner will get feedback based on more experienced musicians, and this could be applied also to experienced musicians, for example, let's say a musician wants to learn how to play a certain style of music, he could use this comparison limiting the learning data to that specific style musicians and this way compare his own playing and see why they sound better. It could even be limited to a very specific technique, like in the case of guitars, the Hendrix chord that uses the thumb to sound the root note of a power chord.

Ultimately, this approach will analyze how well leap motion can perform in that specific scenario and how the conjugation of machine learning methods with multi-dimensional DTW can help to achieve the analysis of very specific movements. There could arise problems with the

Figure 3.1: Proposed setting of the sensor for piano

latency of captured data and synchronizing leap motion with sound capture. Also choosing parameters and features will be critical for the good function of a solution using machine learning methods.

3.3 Proposed Solution

Briefly, the proposed solution will consist on using the Leap Motion sensor to capture and save all the data required for learning and testing phases. To this data apply a dynamic gesture recognition method to extract the features and train the network the specific gestures patterns and conjugate this with a multi-dimensional DTW adapted version. This DTW aims to analyze the similarity between gestures and changes between those gestures in each of the specific information saved by the leap motion sensor, providing a detailed comparison between the reference data and the test data.

For the data, there will be studied and planned which are the basic to intermediate chords, scales and beginner-friendly common progressions to develop a set of exercises that both parts, experienced and beginners, will have to complete.

To capture data, the Leap Motion sensor is going to be placed directly on top of the middle of the piano in order to capture the maximum number of keys possible with one sensor as illustrated in the figure 3.1.

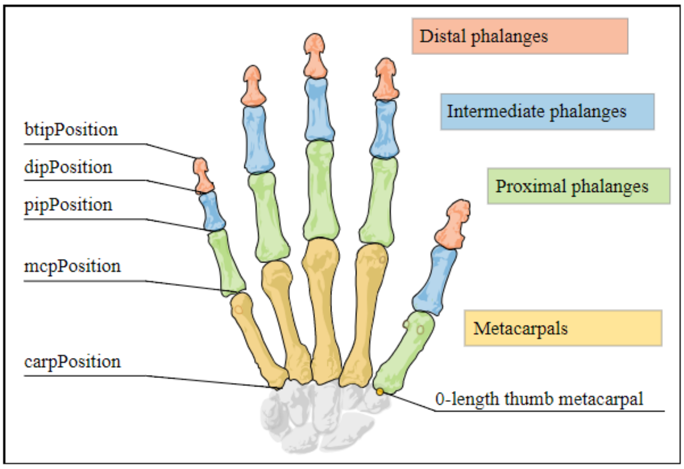

Figure 3.2: Leap Motion hand representation

After capturing this data, Leap Motion will provide it in a set of frame information containing 3D position vectors, orientation and rotation values for each of the phalanges and metacarpal bones of the hand and also the position for the joints as illustrated in the figure 3.2.

The experienced musician provided data will be used to train the neural network to recognize in real-time which gesture is being executed. So for each frame data, the neural network will output the gesture it recognizes or no gesture if it doesn't recognize one at all. For each gesture recognized, the system will save the time frame when it was first recognized and when it stopped being recognized. This saved time frames will have a big purpose for the DTW algorithm.

The DTW algorithm will learn from the experienced dataset in the following way. It takes all the combinations possible from the pairs [gesture, next gesture] basing on those time frames saved and will learn that specific transition from one gesture to other with all the frames data collected between those two time frames, and this will be used as the reference data. The same logic will be applied to the testing data that will be used to be compared with the reference data. This DTW algorithm also allows comparing the sound signal between the experienced player and the beginner. This is represented in figure 3.3. It's also possible to capture the sound and by analyzing the pitch know if the user is playing the right notes or not, but as the main goal is comparing the gestures and due to lack of good results on research it was chosen to stick with the DTW version.

As leap motion gives information on each one of the hand bones, this architecture should be able to give feedback for each of these bones information regarding that the DTW will compare all the different dimensions of data contained in the frames.

This output information will be presented to the user in order to inform him in which parts of the hand he is or not performing like an experienced musician for each gesture and gesture transition.

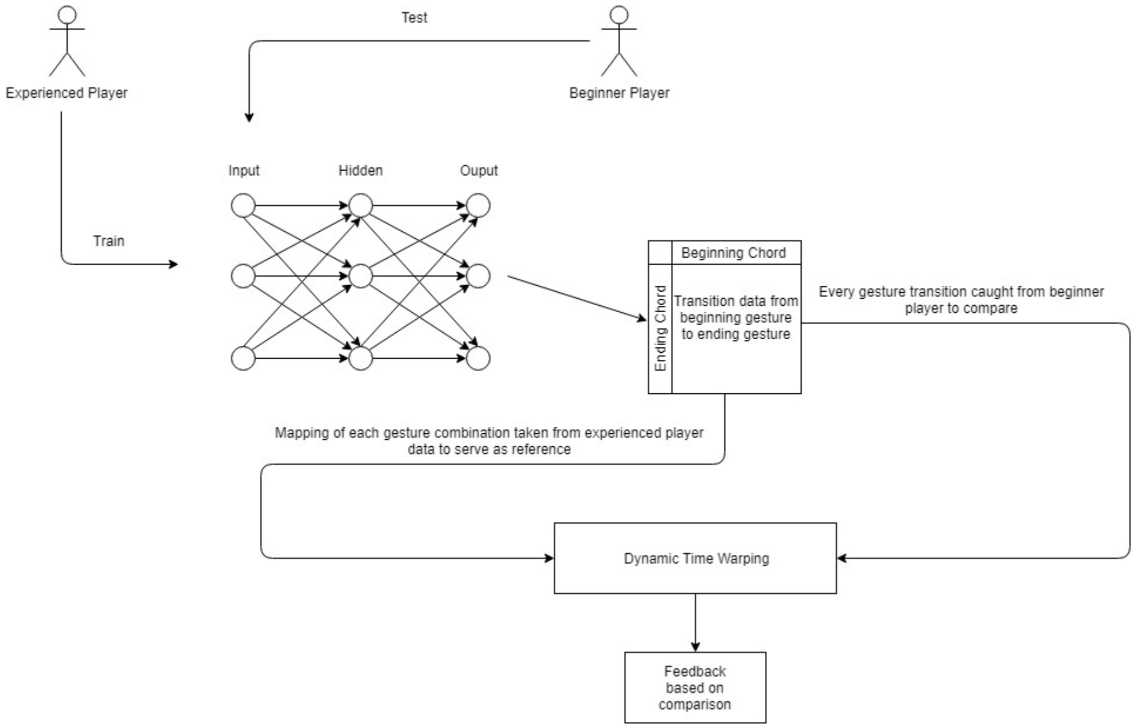

Figure 3.3: Flow scheme of the proposed solution

3.4 Experiments

One of the most essential aspects of the proposed solution is data gathering, if not the most essential. For this, it will be chosen pieces for beginner, intermediate and advanced levels and each participant will play each of the pieces corresponding to their current level four times and their movements and sound will be recorded. Also, there is a need to assess their current level to know in which category are they. To do this a questionnaire will be developed asking about their grade in piano education, the age they had when they started playing, the number of education years and how much time they usually spend practicing per week. Ultimately, there will be another form with questions regarding if the setup of the sensor hindered their performance in any way.

After the data gathering is done, it will be fed into the comparison module and output the results given by it, while trying various setups of the parameters.

3.5 Limitations and Changes in the Architecture

With the advances of the implementation, it was proven not possible in the period available to develop and specify the specific gestures for the pieces, which was a necessity for the HMM modules. So a different path was taken, changing it from trying to give feedback on the transition to these gestures to evaluate the total performance using the distances resulting from the comparison module. This means that the HMM module was discarded and the comparison module instead of receiving data for the combinations of [initial gesture, final gesture] would receive the whole recording putting the expected results first thought different from the beginning.

In consideration of the changes, this dissertation followed the path of an assumption that the result of the comparison would have more dissimilarity the higher the difference in skill level between the reference and the test. It was also thought that by comparing the different performances of the same person it would be possible for the result to assess the consistency of that person in that specific piece.

To achieve this, the architecture of the methodological approach changed from segmentation of the data using the HMM or NN and using the DTW to match the data segmented of the tests with the references to a more detailed implementation of the DTW to use it as a means of assessment between the difference in the comparisons made, which will follow the structure described in the following section.

Chapter 4

Implementation

This chapter addresses the implementation details of the project while considering its limitations and different approaches. Based on the previously explained methodological approach, it starts by explaining the limitations, Leap Motion Data Recorder development and ending with the Comparison using Dynamic Time Warping.

4.1 Leap Motion data recorder

This section will explain how the Leap Motion data was recorded addressing the sensor position, how the recorder was implemented and how the recording was planned.

4.1.1 Sensor Position

It is extremely important to have the sensor in the same position between recordings and at the same time have the best ratio between pianos keys covered and the optimal area of the sensor capture. Regarding this, it is known that the optimal capture area of the Leap Motion is a cube between 30 centimeters and 60 centimeters of distance from its center. Also to have the most piano keys covered it should be position in the center of the keys.

Various positions were tested considering the aforementioned, positioning it in the center of the piano, which means a 0 value for the x-axis in its center, and varying the other variables, the y-axis, the z-axis and its rotation. Regarding the z-axis, the sensor was position between the minimum and maximum optimal ranges, and it has proved more efficient and with much more capture range closer to the maximum range. Also, it would hinder way less the pianist with more altitude, so the chosen distance was 55 centimeters from the piano keys. Otherwise, the y-axis and the rotation were more of an empirical method based on the visualization of data.

It would be expected to have a value 0 for the y-axis as well and for the sensor to be parallel to the piano keys, but as the keys reflected a lot of infrared light it was decreasing the quality and the hand representation confidence of the data as there was much less contrast between the

Figure 4.1: Final setup of the Leap Motion and piano keys names.

hands and the background. To try to avoid this was tried to move the sensor further away from the piano and more into the piano, and although both options provided less direct reflection the capture with the first option, as the second failed to capture the hands quite frequently. Regardless of moving the sensor so it was not directly on top of the piano keys, it was still failing the hand capture with some considerable rate, so changing its rotation was the option that fixed that to a certain extension. By rotating it around the axis y towards the piano direction the quality of data received was improving.

After all this analysis, the sensor was positioned in a way that it would give the values x, y and z of [0, 0.2, 55] centimeters while pointing at the top of the key D4 as the figure 4.1 represents.

This was as important aspect of the implementation, as it needed to devise a way to hold the sensor in that specific position while considering conditions: it must not hinder the performance of the pianist and the pianist should be able to read the sheets without effort.

To achieve these, various methods were tested putting the performance on a higher level of importance than reading the sheet. The lack of available resources led to far from perfect initial setups ending with a not optimal but a functional one. Firstly, was used a speaker support with a book cover with a circle hole holding the sensor with which the first positioning tests were made, illustrated in figure 4.2. The second option, represented in figure 4.3, was to replicate the positioning of a support placed in the back of the piano, this one worked much better with acceptable results but it would completely erase the possibility of reading the sheet. The final and used setup was a lamp support with a movable arm having the cover and lamp removed which gave better data quality and much more freedom to position the sensor, visible in figure 4.4.

Figure 4.2: First Leap Motion Setup

Figure 4.3: Second Leap Motion Setup

Figure 4.4: Third and final Leap Motion Setup

4.1.2 Unity and Leap Motion API

The Leap Motion provides an extensive API that is written in C, and there are various implementations to transcribe that API to other languages in previous versions, but recently they launched the version 4 or Orion version 2 that currently only has an implementation in C# for Unity. As it is reported that this version corrects some issues with overlapping elements like hiding the thumb behind the palm and this type of movement is used a lot in piano it was chosen to use this recent version. Also, Unity gives the possibility to visually analyze the current representation of the hand in real-time through the graphical 3D representation in the Unity world. Although the Leap Motion API doesn't have either a record data or even a serialize data features, and their data structure is not serializable, so the use of commonly used libraries like json.net was not possible.

Based on the previous mentioned, the recorder was fully implemented in C# using the Unity engine to create the features of sensor position validation, start recording, stop recording, and save recorded data to a file in order to use on the comparison module.

To use the Leap Motion in Unity a basic set of steps were needed to follow so the API was available. First of all, it was need to download the SDK from their website and after that manually add the assets provided to the Unity project. With this done, the project could follow further implementation and make use of the present samples offered by the assets imported. These samples made easier to understand how actually the API functioned and boosted the implementation time of the graphical representation of the hands in the Unity scene.

To be able to access Leap Motion data it was needed to serialized the field leapProvider and override its onUpdateFrame function with a custom one. This custom function will take care of the data gathering, if the state of the app is in recording, and the sensor position validation, in case the app is in the state of validation. In the rest of the states, there are the options of revalidating sensor position and save data. Also in the recording state, there is the option to stop recording.

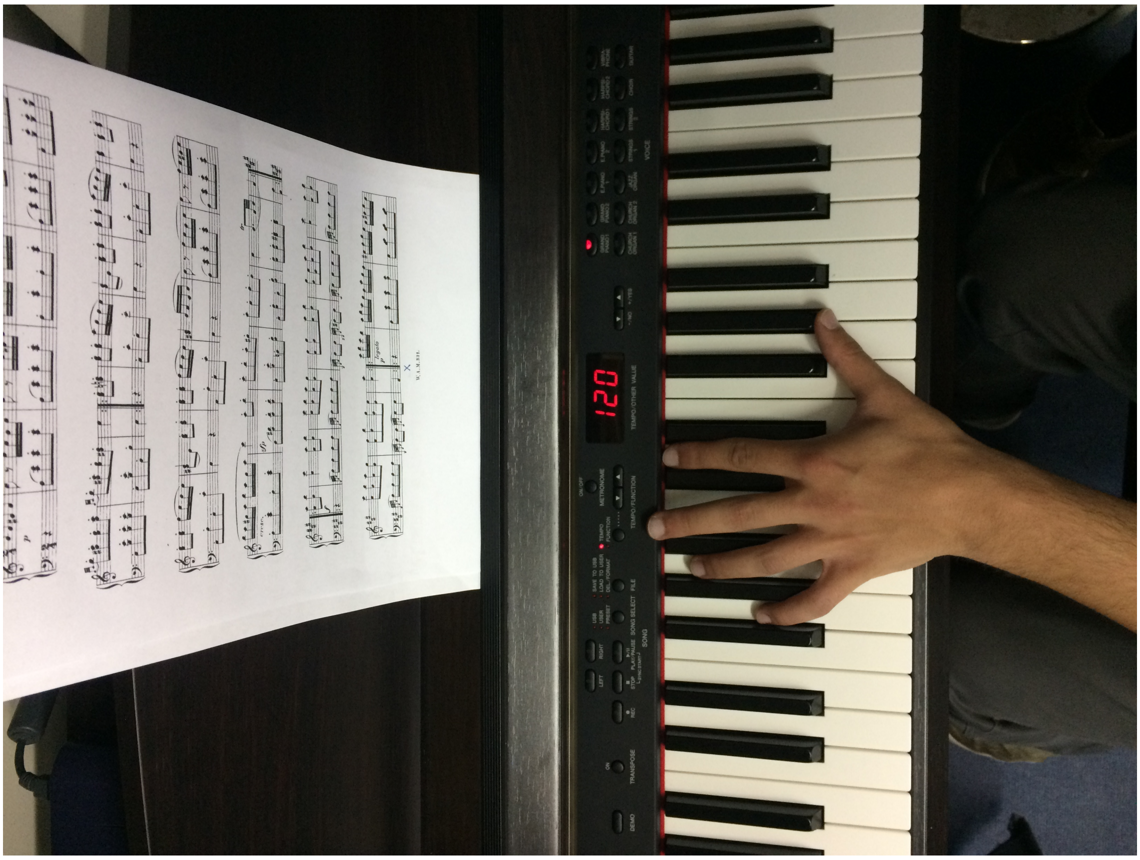

Figure 4.5: Validation hand position.

4.1.3 Validation Feature

One of the most important features developed for the quality of results and also one of the limitations of this dissertation. To achieve this type of feature the best option would be to have static elements recognized by the sensor position in two different specific places so if the sensor would capture them with the same values it would be in the same position. As this type of feature doesn't exist in the provided API and was not found resources to make the sensor recognize elements that are not hands, a different approach was used.

Instead of using static elements, it was used the left index finger on the key D4 of the piano with its fingertip leaning to the top end of the key, as can be seen in figure 4.5. This way it was an approximated static point of reference that joined with the infrared cameras. As the values of the sensor oscillate given its precision and also because of the lower quality originating from the infrared light reflected by the keys, it needed to make the validation position with a certain error otherwise it would be unfeasible to react in time to click the green button to validate the sensor position that only appears when in the right position. This error is a range of 6 millimeters. As the support was going to be fixed, a screenshot of the images of the camera with the correct position and this point reference validation were used to validate the sensor position. Also, the application is capable of indicating where to move the sensor to reach the values of the specified point using

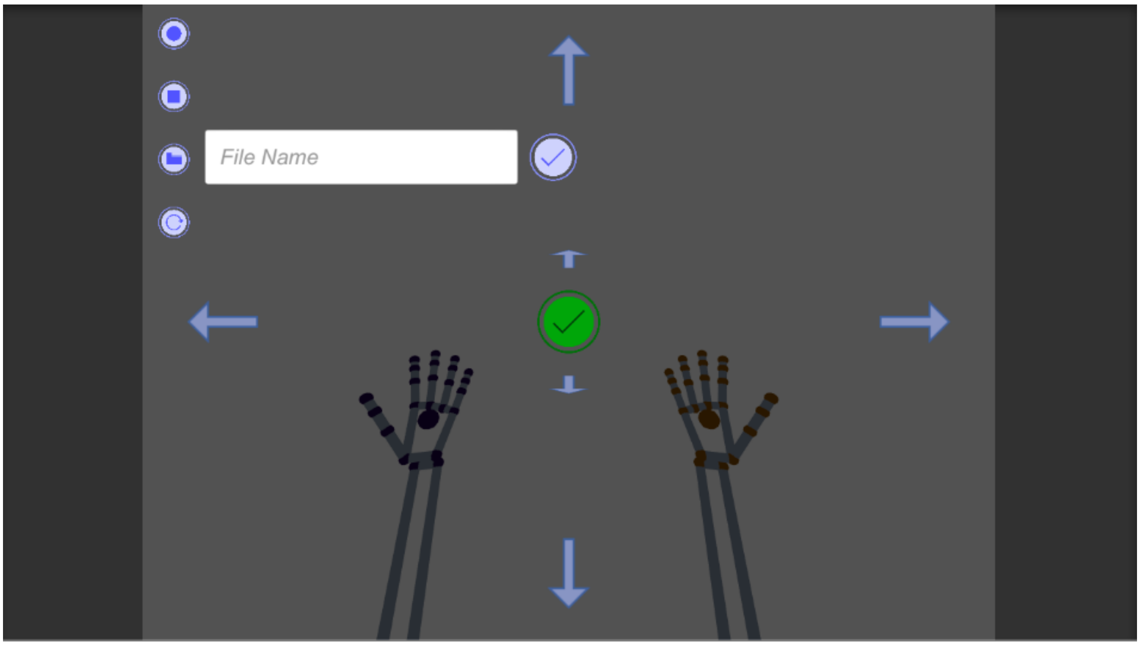

Figure 4.6: Recorder interface.

the blue arrows, this feature is showed in figure 4.6.

4.1.4 Recording Features

As the data was not serializable, it was required to create new serializable classes to hold those values and then serialize them to export to a file. So there were created classes that follow the same type of structure of the Leap Motion base data which are FrameData, HandData, FingerData and BoneData.

The FrameData class is composed by a float Timestamp, a current FramesPerSecond and a list of objects of HandData.

The HandData has a boolean variable IsLeft which indicates if the hand is the left hand or not, many variables of Vector3 type for the palm position, palm velocity, palm normal, palm direction, stabilized palm position, and wrist position. It also has a field for the hand representation confidence which ranges from 0 to 1 and a list of FingerData objects.

Following the previous class, the FingerData has a string field for the type of finger named Type that can have the values "TYPE_THUMB", "TYPE_INDEX", "TYPE_MIDDLE", "TYPE_RING" and "TYPE_PINKY". Other two fields of Vector3 are for the fingertip position and the direction vector. It also has variables for its width and length as well as one for checking if the finger is extended. Lastly, it has a list of BoneData objects.

The last class, BoneData, has a string for the bone type equal to the FingerData but with the values "TYPE_METACARPAL", "TYPE_PROXIMAL", "TYPE_INTERMEDIATE" and "TYPE_DISTAL". Also has two Vector3 fields for the center of the bone and the bones direction.

The state of the application is controlled by the interface shown in figure 4.6 and all of the following references to buttons and inputs refer to the same figure.

It is only possible to record data after validating the position of the sensor by clicking the green button. The application starts recording when the top left button is clicked. It changes the state of the application to recording and instead of the onUpdateFrame function checking if the position is valid it begins to converting each Leap Motion frame to the custom FrameData and adds it to a private array of frames.

It is possible to stop recording by clicking the stop button, which is the second of the top-left buttons, that will change the state to not recording and will enable the option of saving data. By clicking the folder button, third of the top-left buttons, it will appear a text input to insert the name of the file and saves it by clicking the arrow button that shows up with it. If instead of saving it is clicked to record again, all of the data previously saved is discarded as this action is considered as a bad recording that needs to restart.

Finally, the last of the top-left buttons is the revalidation button, which allows to enter the validation state and validate the sensor position again during the same run time.

4.1.5 Saved file format

Clicking the saving button will generate a JSON file with the text inserted in the input, with the data present in the recording data array serialized with json.net. The structure follows the exact same structure as the classes created and it is represented in the listing 4.1 omitting the complete arrays of data.

```
[
  {
    "Timestamp": 0.00146484375,
    "FramesPerSecond": 115.35173,
    "Hands": [
      {
        "IsLeft": true/false,
        "PalmPosition": {"x": 106.079582, "y": -84.9041748, "z": 459.945862},
        "PalmVelocity": {"x": -4.350255, "y": -7.432712, "z": -16.4543667},
        "PalmNormal": {"x": -170.086334, "y": -353.415527, "z": 919.874},
        "PalmDirection": {"x": 63.5860329, "y": 927.594055, "z": 368.1386},
        "StabilizedPalmPosition": {"x": 0.0, "y": 0.0, "z": 120.0},
        "WristPosition": {"x": 113.075768, "y": -144.255447, "z": 431.388916},
        "Confidence": 0.743473053,
        "Fingers": [
          {
            "Type": {FINGER_TYPE},
            "TipPosition": {"x": 52.24978, "y": -72.759285, "z": 487.773071},
```

```
                    "Direction": {"x": -374.272339, "y": 724.5624, "z": 578.7307},
                    "Width": 100.0,
                    "Length": 3.75485539,
                    "IsExtended": true,
                    "Bones": [
                      {
                        "Type": {BONE_TYPE},
                        "Center": {"x": 84.19068, "y": -128.018082, "z": 446.022},
                        "Direction": {"x": 0.0, "y": 0.0, "z": 0.0}
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
```

Listing 4.1: Json file output scheme

4.2 Sound

In the end, there was not sufficient time to make a sound analysis implementation, but the recorded synchronization was implemented either way. To do this was used the OSC connection the recorder implemented previously with Reaper, as it was the technology suggested to record on for its features and low learning curve for simple tasks and also had integrated compatibility with the OSC.

To achieve this, it was needed for the recorder, when changing states, to send messages to Reaper indicating it to start or stop the sound recording. So an OSC object was created in the scene that would establish a connection with the reaper, taking inputs of the input port, the output IP and output port. These last values were given as the IP and port of the input OSC of the reaper project. After this, the recorder would use this created OSC connection and send messages to the addresses "/record" and "/stop" for recording and stop recording respectively. There was no need to implement custom methods for the reaper actions as these were present in the default actions given.

4.3 Comparison using Dynamic Time Warping

This section will describe the implementation choices taken regarding the Comparison module that uses the DTW algorithm, dividing it into distance function and general algorithm.

4.3.1 Python and fastDTW

To implement the comparison module, Python was the chosen language because of its simplicity and quantity of available libraries. In fact, instead of implementing the DTW algorithm and its optimizations, a library name fastDTW was found that already does this with the option of providing a radius of analysis and a custom distance function. All this while taking advantage of the better execution times of CPython and instead of having the original algorithm of time complexity of $O(n^2)$ this implementation gives an accurate approximation while trying to have a linear complexity in both time and space, so this implementation is faster than a regular Python implementation of the DTW [SC07].

The fastDTW library receives as mandatory inputs the two time series to be analyzed and returns the resulting distance between them as well as the resulting chosen path. There are also optional parameters like the radius and the distance function, which were the ones used in the after-mentioned implementation.

4.3.2 Input

This module needs to be fed with the recorded data from the recorder module, so for this when you run the program there is a window pop-up asking to select the reference files and after loading them another window appears to select the reference files. This way will fasten the process of analyzing the results as it is only needed to run the comparison module once per each of the recorded piece files comparing every test file with every reference in one go.

4.3.3 Data Preparation

The DTW is an algorithm to compute the dissimilarity between signals, this means that the data needs to be in the format of a time series which is not the case of the output format of the recorded data. To achieve this the JSON files are deserialized to an object and then processed to new class Recording which will hold the data for a single JSON file. This class will have a list of timestamps, a list of current frames per second and two objects of another class Hand for the data of the right and left hands.

As the sensor can have data loss, which means that sometimes it doesn't recognize one or both hands, there was a need to add timestamps to this class Hand as well so that it did not exist more time stamps than actual hand representations. The Hand class will have lists containing all the values through time of the fields timestamps, confidence, palm position, palm velocity, palm normal, palm direction, wrist position and an object of Finger for every finger type. Likewise, the Finger class will follow the same structure, and for the same reason as the Hand class will have its list of timestamps. So this class will have arrays for the fields, timestamps, fingertip position, direction, width, length, is extended and an object of Bone for each of the bone types. Once again, the Bone class also has that array of timestamps in case of non-existence and will have arrays for the timestamps, the center position and the direction of the bone. All of these classes are shown in the listing 4.2

```
class Recording:
def __init__(self):
    self.timestamps = list()
    self.fps = list()
    self.left_hand = Hand.Hand()
    self.right_hand = Hand.Hand()

class Hand:
def __init__(self):
    self.timestamps = []
    self.confidence = []
    self.palm_position = []
    self.palm_velocity = []
    self.palm_normal = []
    self.palm_direction = []
    self.wrist_position = []
    self.thumb = Finger.Finger()
    self.index = Finger.Finger()
    self.middle = Finger.Finger()
    self.ring = Finger.Finger()
    self.pinky = Finger.Finger()

class Finger:
def __init__(self):
    self.timestamps = []
    self.tip_position = []
    self.direction = []
    self.width = []
    self.length = []
    self.is_extended = []
    self.metacarpal = Bone.Bone()
    self.proximal = Bone.Bone()
    self.intermediate = Bone.Bone()
    self.distal = Bone.Bone()

class Bone:
def __init__(self):
    self.timestamps = []
    self.center = []
    self.direction = []
```

Listing 4.2: Classes for reformatting recording data in a structure of time series

Also, for the posterior representation of every position through time, this same arrays of data were divided into each of the axes, x, y and z, so it could be represented graphically to visually assess the distances even though this alternative takes account the differences in time while the algorithm does not.

4.3.4 Distance function

Three different versions of the distance function were outlined. The first was the base euclidean distance function applied to a 3D context which is represented in equation 4.1

$$d(P_1,P_2) = \sqrt{(P_{1_x} - P_{2_x})^2 + (P_{1_y} - P_{2_y})^2 + (P_{1_z} - P_{2_z})^2} \tag{4.1}$$

As the Leap Motion data provided a hand representation confidence and with lower confidence its error in values increase, it was thought of a way to utilize this in the algorithm in order to see if this factor would influence the results to a better result. The idea resulting is to use an inverse of the average of the hand representation confidence of the hand that both points belong to, which is represented by the equation 4.2. The reason for it to be an inverse is that one would want the algorithm to prioritize values with higher confidence and the confidence is a value between zero and one, so if it was the regular average it would give a lower distance, which the algorithm tends to choose, instead a higher one, which the algorithm tends to avoid.

$$d(P_1,P_2) = \frac{1}{0.5 * (P_{1_{conf}} + P_{2_{conf}})} * \sqrt{(P_{1_x} - P_{2_x})^2 + (P_{1_y} - P_{2_y})^2 + (P_{1_z} - P_{2_z})^2} \tag{4.2}$$

After this, it was thought that it would be beneficial to only prejudice distances with lower representation than accepted, so it came to the idea using the confidence equation only if it is lower than a certain threshold and use the regular euclidean distance otherwise. This is showed in equation 4.3.

$$d(P_1,P_2) = \begin{cases} \frac{1}{0.5*(P_{1_{conf}} + P_{2_{conf}})} * \sqrt{(P_{1_x} - P_{2_x})^2 + (P_{1_y} - P_{2_y})^2 + (P_{1_z} - P_{2_z})^2}, & \text{if } \frac{P_{1_{conf}} + P_{2_{conf}}}{2} \leq \theta \\ \sqrt{(P_{1_x} - P_{2_x})^2 + (P_{1_y} - P_{2_y})^2 + (P_{1_z} - P_{2_z})^2}, & \text{otherwise} \end{cases} \tag{4.3}$$

4.3.5 Weighted Dynamic Time Warping

In this phase, with all the data already prepared and although using a faster accurate approximation of the DTW, the resources and time taken to analyze all of the points and data given by the Recorder module would still take too much time. So to decrease this factor, certain points of the hand were chosen to lessen this data. After brainstorming about this, it was opted to analyze eight different points per hand, which means that each comparison will still run eight DTWs for each hand, which are the five fingertip positions, the palm position, the wrist position and the palm velocity. This means that for each of these points will be given a distance value.

Also the thought that some of these points might be more influential in the performance than others led to the use of the weights for each of them, represented by the equation 4.4 for one of the hands having the W_x being the weight and the F, W, P and V being the indicators for Finger, Wrist, Palm Position and Palm Velocity respectively.

$$HandDTW = W_F * \sum DTW_F + W_W * DTW_W + W_P * DTW_P + W_V * DTW_V \tag{4.4}$$

After this, the total distance of comparison is given by the sum of the two hands, given by the equation 4.5.

$$FinalDTW = LeftHandDTW + RightHandDTW \tag{4.5}$$

As the values of the DTWs, before the multiplications with the weights, are always the same they are calculated first and saved for posterior calculation making the analysis of how the variation of the weights will affect the matching of the tests with references. This way there is no need to needlessly execute the singular DTWs multiple times for different weights and as the distance function and the radius are the only factors that will change those values for the same comparison.

4.3.6 Output

As output for this module, the system will create a folder named graphs in which for every comparison will create a new folder named "[reference file name]-[test file name]" with a .csv file for the results for every specific DTW calculation as well as three images with plots of every axis, x, y and z, of every considered point overlapping the reference and the test time series with the name structure of "[reference file name]-[test file name]-[point name]-[axis]", illustrated in the figure 4.7.

Figure 4.7: Representation of output graphs of one point for axis x, y and z respectively.

Apart from this, it also creates another .csv file containing tables of results of total distances for each combination of references and tests, and the best match[1] for the left hand, the right hand and the total distance for all the combinations of weights. This file follows the name structure of "[piece name]-[distance function]-[radius]".

4.3.7 Feedback

A feedback component was added to the program, and it takes the resulting path of the various DTW's for each position, which is in the structure of an array with elements of [reference index, test index], and build new time series for both the test and the reference with the alignment of both. So each point of the smaller time series is given a matched point of the bigger one, and

[1]The best match is purely the result with the lowest distance value, which means the lower dissimilarity

by comparing these points it is possible to calculate the direction vector from the test point to the reference point and the distance that the points are apart from each other. Also, it seemed convenient to also output average, minimum and maximum distances that a specific position of the hand had. So by implementing this a JSON is created, with the structure represented in listing 4.3, for each comparison made, having the name of "reference participant-test participant-feedback" containing a right and left-hand element, and each of these has an array of the feedback for the various analyzed positions. These positions will have the position name, the average, minimum and maximum distances and an array of elements for each point in time with the structure of [direction vector, distance]. This direction vector is a 3D vector, that is represented as [x, y, z].

```
{
    {hand(left/right) - String}: [{
        "position":{position name - String},
        "average":{float}
        "minimum":{float}
        "maximum":{float}
        "feedback":[ [{direction vector - [float,float,float]}, {distance -
            float}] ]
    }]
}
```

Listing 4.3: Feedback output json file structure

Chapter 5

Experiments, Results and Analysis

The current chapter presents experiments process description and the results of the current state implementation attached with the discussion regarding the interpretation of these results about their quality, importance and limitations.

5.1 Experiments

For the data gathering were chosen four pieces, two for the beginner level, one intermediate level and one for the advanced level. These pieces were chosen based on research about which pieces correspond to which level according to the piano course's program of the National Conservatory of Music of Portugal and came to choice trying to maximize the number of people that would have played that piece before. The only one not following this rule was the advanced piece, as it would be hard enough to gather advanced players, and even harder to gather advanced players that had studied the same advanced piece. Also, this advanced piece would be less significant for the context of the application of this solution, but still interesting to see the potential on a professional level usage of the idea.

For the beginners, the pieces chosen were the complete Musette in D sharp by Bach, often used by schools in the earlier learning phases, and Fur Elise by Beethoven, which is an intermediate piece but the first movement is beginner level and only this part was considered.

For the intermediate, was chosen the Alla Turca by Mozart, as it is a piece of widely recognized music and even if not studied before the melody is recognized by almost everyone and probably will make things easier for the participants to learn it instead of a piece never heard before.

For the advanced piece, was chosen a Nocturne in F sharp major of Op15 n2 by Chopin which is a grade 10 piece, randomly chosen from the available list, while considering the range it occupies on the piano keys so it wouldn't go out of the sensor range. Although chosen, the time necessary to study this piece was not enough for the participants so it was not performed by any of them.

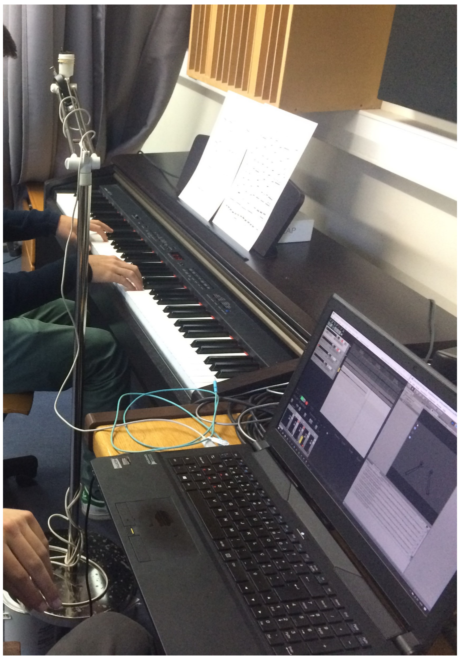

Figure 5.1: Recording one of the participants.

In total there were eleven participants, most of them belonging to the faculty community, with ages ranging from eighteen to forty-four years old. As for their levels, it was replicated the Ollen Musical Sophistication Index's[1] (OMSI) questions and the result saved by each of the participants. Also was noted which pieces a participant would play, and which of those they are comfortable playing. This information is resumed in the table 5.1.

For each piece, each participant performed four times, as in figure 5.1, and at the end of each piece, the participant was asked to name the two top performances between those four. After performing all of the pieces, the participant was asked to answer three questions about the influence the sensor had on their performance.

On each run, it was recorded the Leap Motion data using the implemented recorder and the sound produced by the piano in Reaper using an M-AUDIO Fast Track Pro interface. This recording environment is illustrated in figure 5.2.

Each piece except the Musette was shortened, meaning only an excerpt of it was asked to be performed. Also, the participants were indicated to follow the sheet disregarding the repetition indications. So if a piece had the structure of AABA, it was asked to perform only ABA in order to shorten the size of data while analyzing the same type of movements either way. These indications can be seen in the appendix.

In the end, seven participants performed Musette having six of them being comfortable with the piece, ten performed Fur Elise being all ten comfortable playing it and seven of the participants performed Alla Turca and only three of them said to be comfortable playing it.

[1] http://marcs-survey.uws.edu.au/OMSI/

Table 5.1: Data gathered from the participants

Participants	Age when started	Age now	Years of lessons	Index	Musette		Fur Elise		Alla Turca	
					Best	Second Best	Best	Second Best	Best	Second Best
1	8	29	17	445	3	4	3	2	x	x
2	10	21	7	535	4	2	4	2	2	1
3	18	20	0	13	x	x	4	3	x	x
4	6	44	14	833	4	2	2	3	2	3
5	4	21	17	209	x	x	1	2	1	2
6	6	22	12	96	4	3	2	1	x	x
7	6	18	12	415	3	4	4	1	1	4
8	6	22	8	493	x	x	3	2	3	2
9	10	18	5	421	x	x	2	1	3	4
10	4	25	10	293	2	3	3	4	4	3
11	8	25	5	158	2	4	x	x	x	x

Table 5.2: Data loss from the participants performances

Participants	Musette				Fur Elise				Alla Turca			
	1	2	3	4	1	2	3	4	1	2	3	4
1	0%	1.2%	0%	1.9%	0.1%	18.9%	16.7%	21.6%	x	x	x	x
2	1.1%	3.2%	2.2%	3.4%	3.8%	17.8%	0.4%	0%	5.6%	0.3%	2.8%	0.4%
3	x	x	x	x	0%	1.7%	0%	0%	x	x	x	x
4	0%	0%	0%	0.8%	53.4%	1.9%	9.8%	10.5%	5.2%	1.1%	3.2%	2.8%
5	x	x	x	x	3.4%	4.5%	6%	2.8%	10.7%	24.6%	1.2%	25.2%
6	0%	0%	0%	0%	0%	0%	1.3%	0%	x	x	x	x
7	0.2%	0%	0%	0%	0%	1.3%	0%	0%	0%	0%	0%	0%
8	x	x	x	x	0%	0%	0%	1.5%	5.8%	0%	0%	13.3%
9	x	x	x	x	0%	0%	0%	0%	0%	0%	0.1%	0.5%
10	0%	0%	0%	0%	9.5%	0%	0%	0%	2.1%	1.1%	0.9%	0.8%
11	9.2%	0%	0.7%	0.4%	x	x	x	x	x	x	x	x

Table 5.3: Hand confidence from the participants performances

Participants	Musette				Fur Elise				Alla Turca			
	1	2	3	4	1	2	3	4	1	2	3	4
1	75%	75%	75%	75%	75%	73%	73%	73%	x	x	x	x
2	69%	71%	68%	69%	73%	72%	73%	73%	72%	72%	73%	72%
3	x	x	x	x	75%	76%	75%	76%	x	x	x	x
4	68%	68%	68%	67%	55%	73%	72%	71%	69%	70%	69%	69%
5	x	x	x	x	69%	70%	69%	71%	63%	62%	62%	62%
6	77%	77%	78%	77%	76%	77%	78%	78%	x	x	x	x
7	78%	78%	78%	76%	76%	76%	76%	76%	77%	77%	77%	77%
8	x	x	x	x	73%	75%	74%	74%	74%	73%	73%	73%
9	x	x	x	x	70%	70%	70%	70%	70%	70%	70%	69%
10	72%	72%	72%	70%	74%	74%	73%	74%	70%	71%	71%	71%
11	74%	75%	74%	75%	x	x	x	x	x	x	x	x

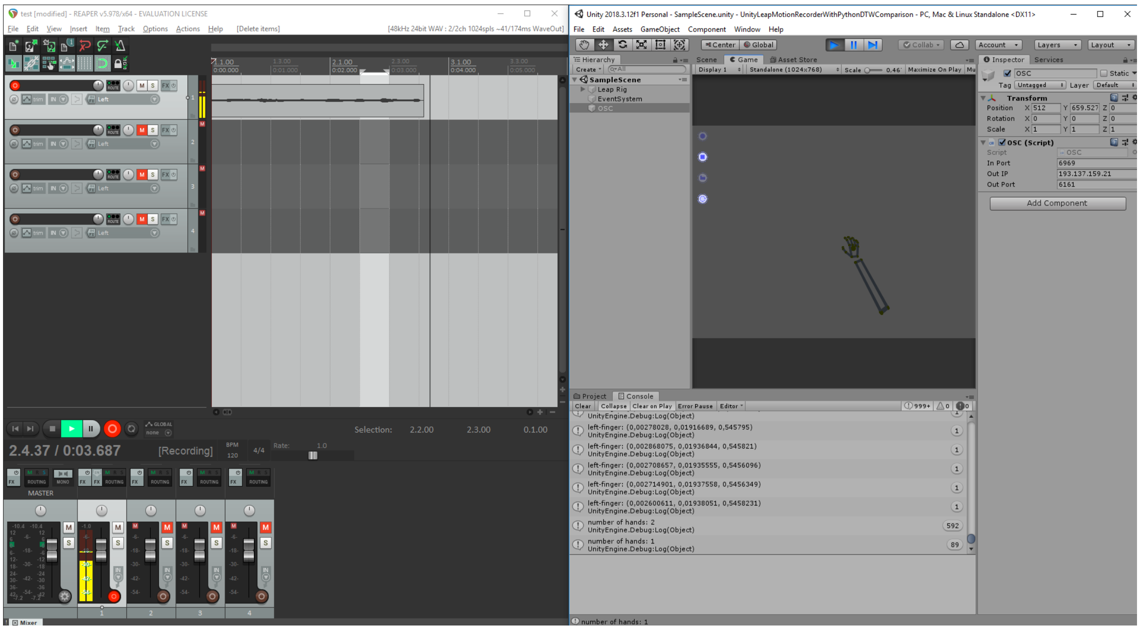

Figure 5.2: Recording environment.

5.2 Results and Analysis

After recording and gathering data with the participants it was time to test if the assumption that the higher the distance the higher the difference in level. To do this, three different options were chosen to classify the level, the OMSI, the number of years the participant had lessons and the total number of years since they first started playing. For each of these, would be calculated the difference between participants and then correlated with the resulting values of the comparison module. Before calculating the Pearson correlation the data was normalized, having the OMSI divided by 1000 because it ranges from 0 to 1000, as the years of lessons ranged from 0 to 17 was divided by 17 and lastly, the years of experience were divided by 36 as they ranged from 2 to 38. For the distances, the global minimum and maximum were used to normalize the data, so the minimum and the maximum were stored and then to the current distance the minimum was subtracted and then divided by the difference between the minimum and the maximum.

The data loss and hand confidence values could have influenced the results and the correlations calculated from those results. Based on tables 5.2 and 5.3, it is possible to see that the data loss was not significant except for one case. As that case is not one of the two best performances of that participant it won't alter the results, so the data loss wasn't that much of a factor. On the other hand, the average of the hand confidence shows mostly values from 70% to 78% which might have altered results. This low confidence is mainly due to the infra red light reflected by the keys, which might be a limitation to the recognition.

After running various tries with hand-chosen weights ranging from 0.5 to 1.0 with comparisons between all the participants, the best value for the correlation coefficient was 0.36 but only for the Musette piece and for the years of experience. The other two pieces produced correlation values close to 0.1. For the rest of the classifications, the result was even worse, having the OMSI value being the worst of the three.

As these values did not show any correlation between the level and the resulting distance, it was tried to optimize the weights focusing on maximizing this correlation by trying all combinations between 0.1 and 1 with an increment of 0.1. In every case, the best correlation were the values of 1, 0.1, 1 and 0.1 for the weights for the palm position, velocity, wrist position and fingertip position, respectively. For the Musette piece, a decent correlation was found with 0.59, but for the rest of the pieces, it just stayed with no correlation at all.

Then the same process was done but only considering the distances between the chosen references and the rest of the participants. The two participants with a higher sum of the years of experience, their age minus the age they started playing, and the years of lessons they had. So for the Musette and the Fur Elise pieces, the references chosen were the participant one and the participant four and for the Alla Turca as the participant one decided to not play that one the second chosen participant was the participant 5. The first participant was a person in the last year of the piano course in Escola Superior de Música e Artes do Espetáculo (ESMAE), the participant four had the complementary, twelve years of lessons, and a specialization done afterward, but practiced regularly for thirty-eight years. The fifth participant had 17 years of lessons equivalent to years of experience. In this new approach, the OMSI had the best results as for the other two being completely inconclusive. As can be seen in tables 5.7, 5.8 and 5.9, the distance function that had the best results was always the normal Euclidean distance function for the OMSI classification. Also it is visible that the results become to get closer to the assumption initially made when the difficulty of the pieces starts increasing and the Alla Turca gets acceptable values: Although they are still above the acceptable threshold of 0.5, they are still too close to this reference value and as the other two pieces don't show a correlation it shows that this assumption is not necessarily true or at least there's no strict correlation between the distances and the level shown in this particular dataset.

This could mean that there is no correlation between the distance in movements in the difference of level. Also, the difficulty of classifying someone in terms of proficiency in an instrument is always a hard process, as it doesn't exist defined rules to do this. The skill comes with training without a doubt but the time, effort and the quality of the practice are variables that this skill level depends and the latest two are hard to quantify. Having private lessons could mean a better practice regime but not necessarily true. Although the later, by only comparing to chosen references, starts showing some evidences that with an ideal close to perfection performance and with the increase of complexity in movements needed for the piece there might be some clear correlation and validation of the assumption, indicating that people with closer levels use more similar movements than the ones with higher difference of levels between them. Also, the lower correlation in the first song might mean that the movements needed to perform are so simple that there is not much difference between beginners, intermediates and advanced players.

Ultimately, a person's proficiency in the piano doesn't equal to a person's proficiency in that piece, although it would be easier to learn a piece for the person of higher level, if you compare the performance where a lower-level person has studied more the piece or knows the piece by heart and the higher-level person did not, it is possible to notice that the person that has studied more the piece had fewer hesitations, more fluid movements and much fewer errors as this person is more comfortable in that piece. This happened in the experiments made as the higher-level participants, which were best fit to act as references, did not study the pieces and were playing it by sight-reading the sheet and sometimes it was perceived that participants with less experience produced a more fluid sound as they didn't need to think as much as them to play the piece, so this cases might have produced outliers in the dataset and as the dataset is not that big to produce a good correlation they might influence this value too much.

Concerning the weights, although the best cases of correlations were the same in the first process, the second had always different best options, and as the first came out too inconclusive it was not reached a conclusion in relation to the values of weights that were better. In other words, which of these evaluated points had the most influence in the resulting final distance. So the initial thought was kept and assumed that the fingers and velocity were the ones more propitious to differences and the palm and wrist position were the most likely to be equal or closer to the reference, having lower values for the first two and higher values for the later ones.

The different distance functions, although changing the values of the correlations a little, on the resulting matches did not show to make that much of a difference as the matches stayed practically the same in every case and the same was verified with the weights that although changing the resulting distance did not change the person who was a match.

Still, regarding analysis between different people, there was also produced feedback for each of these comparisons for each of the matched points in the DTW. So the alignments resultant of the DTW were used to compare the direction and distance from the test to the reference of each moment in time. These results showed that for the closer participants to the reference had visibly lower average and minimum distances than the others. Although this only shows that they were obviously closer to the movements of the reference it is not possible to infer more than that it's possible to actually give feedback between any two people performing by having their recorded movements aligned with the DTW algorithm. It also shows that if the segmentation module would be implemented the DTW would be capable to classify the different movements to similar ones, matching those and then only giving feedback to those specific segmented movements.

A second analysis was made regarding the proximity of tries between the same participant, trying to assess if the perceived quality of their tries was reflected in the proximity distance of those. In other words, it was analyzed if better quality performances of the same person were closer to each other than lower quality ones. For half of the participants performances, 12 out of 24, this came up to be confirmed, and the ones who did not match were mostly participants who were not so sure which of their performances were the best two, and this can be seen in cases where the match is another performance but the values are really close to each other. This data can be seen in the tables 5.4, 5.6 and 5.5 for each piece.

As this analysis focus only on the performance rating in comparison to the comparisons to a reference, it showed that it's not possible to give an accurate estimation of the skill level of

Table 5.4: Comparison of distances between performances of the same person of the piece Musette

	2nd best			other			other			lowest value	correct match?
	left hand	right hand	total	left hand	right hand	total	left hand	right hand	total		
p1	285245.6504	333394.4015	618640.0519	324572.6904	417066.8345	741639.5248	231965.0522	303298.7666	535263.8187	535263.8187	other
p2	340371.9325	451163.6568	791535.5893	355496.3511	482207.5799	837703.9309	253529.8823	412151.737	665681.6193	665681.6193	other
p4	264404.6643	245818.9032	510223.5675	305507.1007	266719.8066	572226.9074	269146.5989	253748.8543	522895.4533	510223.5675	match
p6	269986.3336	357315.4186	627301.7521	430052.2571	546366.033	976418.29	280326.771	302222.1201	582548.8911	582548.8911	other
p7	224029.6537	266803.9429	490833.5966	262356.2664	331292.5078	593648.7742	203905.8987	289853.7404	493759.6391	490833.5966	match
p10	230482.2742	217451.8989	447934.1731	228250.3391	235246.5556	463496.8948	175426.743951122	217359.8519	392786.5959	392786.5959	other
p11	400479.7396	294302.2415	694781.9812	410806.7016	356400.0245	767206.7262	385602.509	329553.8894	715156.3984	694781.9812	match

Table 5.5: Comparison of distances between performances of the same person of the piece Fur Elise.

	2nd best			other			other			lowest value	correct match?
	left hand	right hand	total	left hand	right hand	total	left hand	right hand	total		
p1	315223.436	318263.405	633486.841	333794.0499	364090.0629	697884.1128	282664.9802	387241.9809	669906.9611	633486.841	other
p2	427692.2072	387024.6876	814716.8948	556026.9769	346665.1707	902692.1476	519954.1585	418890.3113	938844.4698	814716.8948	match
p3	248748.221	156660.0116	405408.2325	180062.2406	223088.1963	403150.4369	203856.5087	218209.499	422066.0077	403150.4369	other
p4	336688.6014	225282.2956	561970.897	924057.3077	510409.0501	1434466.358	301253.3166	267324.9234	568578.24	561970.897	match
p5	383769.3343	349397.185	733166.5193	304025.0503	323046.8836	627071.9339	326762.4793	362040.1206	688802.6	627071.9339	other
p6	610818.0051	485734.6755	1096552.681	663792.6007	468878.7325	1132671.333	747402.9039	512462.2303	1259865.134	1096552.681	match
p7	309971.0362	295237.1972	605208.2335	365237.492	266550.2283	631787.7203	271969.5383	363364.5043	635334.0426	605208.2335	match
p8	243390.7332	295042.2819	538433.0151	269845.2534	225480.0435	495325.2969	202899.436	154335.2201	357234.656	357234.656	other
p9	212677.0963	180677.7211	393354.8174	197015.3459	163629.3169	360644.6628	204170.9358	173665.6746	377836.6104	360644.6628	other
p10	319365.1365	390565.1702	709930.3067	301289.4162	339078.4265	640367.8427	226986.8777	235362.7466	462349.6244	462349.6244	other

Table 5.6: Comparison of distances between performances of the same person of the piece Alla Turca.

	2nd best			other			other			lowest value	correct match?
	left hand	right hand	total	left hand	right hand	total	left hand	right hand	total		
p2	669009.2537	651657.8502	1320667.104	562808.8918	496815.6142	1059624.506	460470.9336	551841.6156	1012312.549	1012312.549	other
p4	392730.849	562958.3912	955689.2402	409115.7781	596635.0527	1005750.831	496778.7877	642937.451	1139716.239	955689.2402	match
p5	327038.5252	292856.0768	619894.6021	379462.8154	337025.1231	716487.9385	361437.2917	312111.8394	673549.131	619894.6021	match
p7	313418.9773	452660.192	766079.1693	512747.4418	789372.9933	1302120.435	319021.3729	468584.9931	787606.366	766079.1693	match
p8	288742.296	348906.8372	637649.1332	312843.8323	482851.7458	795695.5781	355501.9959	386748.9356	742250.9315	637649.1332	match
p9	390207.5494	415745.3198	805952.8691	449367.5561	339814.6174	789182.1735	368455.6379	294754.6668	663210.3047	663210.3047	other
p10	266056.5046	344119.079	610175.5836	475317.2558	812541.1015	1287858.357	439465.2418	787598.9174	1227064.159	610175.5836	match

Table 5.7: Correlations in comparison to the reference and different classifications of level of the piece Musette.

Weights				Normal			Confidence			Threshold		
Finger	Velocity	Wrist	Palm	OMSI	Lessons	Experience	OMSI	Lessons	Experience	OMSI	Lessons	Experience
0.5	0.5	0.8	0.8	0.1141	0.0641	-0.4926	-0.0228	0.1490	-0.6234	0.0198	0.1083	-0.5887
0.5	0.5	0.8	1	0.1055	0.0684	-0.5023	-0.0294	0.1520	-0.6300	0.0132	0.1119	-0.5959
0.5	0.5	1	0.8	0.0937	0.0744	-0.5113	-0.0403	0.1573	-0.6372	0.0021	0.1169	-0.6035
0.5	0.5	1	1	0.0863	0.0780	-0.5195	-0.0458	0.1597	-0.6427	-0.0035	0.1198	-0.6096
0.5	0.7	0.8	0.8	0.1252	0.0304	-0.4530	-0.0218	0.1229	-0.5985	0.0209	0.0790	-0.5607
0.5	0.7	0.8	1	0.1167	0.0359	-0.4643	-0.0281	0.1269	-0.6064	0.0145	0.0836	-0.5693
0.5	0.7	1	0.8	0.1055	0.0417	-0.4732	-0.0385	0.1322	-0.6134	0.0040	0.0885	-0.5767
0.5	0.7	1	1	0.0980	0.0466	-0.4831	-0.0438	0.1356	-0.6201	-0.0014	0.0926	-0.5841
0.5	1	0.8	0.8	0.1383	-0.0137	-0.3968	-0.0202	0.0875	-0.5606	0.0220	0.0400	-0.5190
0.5	1	0.8	1	0.1302	-0.0070	-0.4100	-0.0262	0.0927	-0.5701	0.0162	0.0458	-0.5293
0.5	1	1	0.8	0.1199	-0.0015	-0.4186	-0.0358	0.0978	-0.5770	0.0064	0.0506	-0.5364
0.5	1	1	1	0.1126	0.0045	-0.4303	-0.0409	0.1024	-0.5853	0.0013	0.0558	-0.5455
0.7	0.5	0.8	0.8	0.1447	0.0732	-0.4853	0.0103	0.1557	-0.6139	0.0533	0.1168	-0.5796
0.7	0.5	0.8	1	0.1363	0.0765	-0.4937	0.0035	0.1579	-0.6199	0.0465	0.1195	-0.5861
0.7	0.5	1	0.8	0.1262	0.0815	-0.5015	-0.0059	0.1624	-0.6262	0.0369	0.1237	-0.5927
0.7	0.5	1	1	0.1187	0.0843	-0.5089	-0.0118	0.1643	-0.6314	0.0308	0.1260	-0.5983
0.7	0.7	0.8	0.8	0.1531	0.0444	-0.4535	0.0098	0.1336	-0.5948	0.0529	0.0918	-0.5579
0.7	0.7	0.8	1	0.1448	0.0486	-0.4630	0.0033	0.1366	-0.6015	0.0463	0.0953	-0.5652
0.7	0.7	1	0.8	0.1352	0.0535	-0.4707	-0.0057	0.1411	-0.6077	0.0371	0.0995	-0.5717
0.7	0.7	1	1	0.1278	0.0573	-0.4791	-0.0115	0.1437	-0.6136	0.0313	0.1026	-0.5782
0.7	1	0.8	0.8	0.1631	0.0055	-0.4071	0.0091	0.1029	-0.5649	0.0518	0.0576	-0.5247
0.7	1	0.8	1	0.1552	0.0107	-0.4180	0.0030	0.1069	-0.5727	0.0458	0.0621	-0.5332
0.7	1	1	0.8	0.1462	0.0155	-0.4255	-0.0055	0.1112	-0.5788	0.0371	0.0661	-0.5394
0.7	1	1	1	0.1390	0.0202	-0.4353	-0.0109	0.1148	-0.5858	0.0317	0.0702	-0.5470
1	0.5	0.8	0.8	0.1751	0.0821	-0.4766	0.0439	0.1620	-0.6027	0.0872	0.1251	-0.5687
1	0.5	0.8	1	0.1674	0.0845	-0.4837	0.0373	0.1636	-0.6079	0.0806	0.1270	-0.5743
1	0.5	1	0.8	0.1592	0.0885	-0.4901	0.0296	0.1673	-0.6132	0.0727	0.1305	-0.5799
1	0.5	1	1	0.1522	0.0907	-0.4965	0.0236	0.1687	-0.6179	0.0667	0.1322	-0.5849
1	0.7	0.8	0.8	0.1813	0.0586	-0.4522	0.0425	0.1442	-0.5888	0.0858	0.1048	-0.5529
1	0.7	0.8	1	0.1737	0.0617	-0.4600	0.0362	0.1464	-0.5945	0.0795	0.1073	-0.5590
1	0.7	1	0.8	0.1658	0.0657	-0.4664	0.0286	0.1500	-0.5997	0.0718	0.1107	-0.5644
1	0.7	1	1	0.1588	0.0684	-0.4734	0.0229	0.1519	-0.6048	0.0661	0.1129	-0.5699
1	1	0.8	0.8	0.1888	0.0260	-0.4160	0.0404	0.1189	-0.5668	0.0835	0.0762	-0.5281
1	1	0.8	1	0.1815	0.0298	-0.4247	0.0344	0.1217	-0.5731	0.0776	0.0794	-0.5349
1	1	1	0.8	0.1740	0.0337	-0.4309	0.0272	0.1253	-0.5782	0.0703	0.0828	-0.5401
1	1	1	1	0.1672	0.0372	-0.4389	0.0218	0.1279	-0.5839	0.0649	0.0857	-0.5463

performance based on the resultant distance dissimilarity. This could be possible by having a different and more prepared reference dataset where the reference participants prepared the piece with extreme care to the best of their ability, so it would be considered a close to perfection performance of that piece. By having a performance fulfilling these conditions the distance to that performance would be an indicator of a distance to that level of perfection in that specific piece.

Table 5.8: Correlations in comparison to the reference and different classifications of level of the piece Fur Elise.

Weights				Normal			Confidence			Threshold		
Finger	Velocity	Wrist	Palm	OMSI	Lessons	Experience	OMSI	Lessons	Experience	OMSI	Lessons	Experience
0.5	0.5	0.8	0.8	0.3964	-0.4140	0.1271	0.3614	-0.4286	0.0874	0.3225	-0.4108	0.0242
0.5	0.5	0.8	1	0.3942	-0.4108	0.1254	0.3596	-0.4248	0.0859	0.3211	-0.4069	0.0237
0.5	0.5	1	0.8	0.3912	-0.4123	0.1204	0.3560	-0.4265	0.0807	0.3177	-0.4082	0.0183
0.5	0.5	1	1	0.3892	-0.4092	0.1190	0.3545	-0.4229	0.0794	0.3166	-0.4046	0.0180
0.5	0.7	0.8	0.8	0.3936	-0.4300	0.1174	0.3562	-0.4490	0.0749	0.3140	-0.4307	0.0064
0.5	0.7	0.8	1	0.3917	-0.4264	0.1162	0.3548	-0.4447	0.0740	0.3130	-0.4263	0.0065
0.5	0.7	1	0.8	0.3888	-0.4278	0.1114	0.3513	-0.4463	0.0689	0.3097	-0.4275	0.0013
0.5	0.7	1	1	0.3871	-0.4244	0.1104	0.3501	-0.4422	0.0682	0.3089	-0.4233	0.0017
0.5	1	0.8	0.8	0.3892	-0.4500	0.1045	0.3483	-0.4748	0.0582	0.3016	-0.4555	-0.0173
0.5	1	0.8	1	0.3877	-0.4461	0.1038	0.3474	-0.4700	0.0579	0.3013	-0.4507	-0.0163
0.5	1	1	0.8	0.3849	-0.4474	0.0993	0.3440	-0.4715	0.0531	0.2981	-0.4517	-0.0212
0.5	1	1	1	0.3836	-0.4436	0.0988	0.3433	-0.4669	0.0530	0.2979	-0.4472	-0.0201
0.7	0.5	0.8	0.8	0.4084	-0.4065	0.1471	0.3750	-0.4184	0.1094	0.3379	-0.4018	0.0487
0.7	0.5	0.8	1	0.4062	-0.4040	0.1450	0.3731	-0.4156	0.1074	0.3362	-0.3988	0.0473
0.7	0.5	1	0.8	0.4038	-0.4054	0.1408	0.3702	-0.4171	0.1031	0.3335	-0.4000	0.0429
0.7	0.5	1	1	0.4018	-0.4030	0.1390	0.3685	-0.4144	0.1014	0.3320	-0.3972	0.0419
0.7	0.7	0.8	0.8	0.4060	-0.4205	0.1382	0.3707	-0.4362	0.0983	0.3306	-0.4192	0.0328
0.7	0.7	0.8	1	0.4040	-0.4178	0.1365	0.3690	-0.4329	0.0967	0.3293	-0.4159	0.0321
0.7	0.7	1	0.8	0.4016	-0.4190	0.1325	0.3662	-0.4344	0.0926	0.3266	-0.4170	0.0278
0.7	0.7	1	1	0.3998	-0.4164	0.1310	0.3647	-0.4313	0.0913	0.3254	-0.4138	0.0273
0.7	1	0.8	0.8	0.4021	-0.4387	0.1261	0.3639	-0.4594	0.0830	0.3200	-0.4419	0.0113
0.7	1	0.8	1	0.4003	-0.4356	0.1249	0.3626	-0.4557	0.0820	0.3191	-0.4381	0.0113
0.7	1	1	0.8	0.3981	-0.4368	0.1211	0.3599	-0.4571	0.0781	0.3164	-0.4391	0.0072
0.7	1	1	1	0.3965	-0.4338	0.1201	0.3587	-0.4536	0.0773	0.3157	-0.4355	0.0073
1	0.5	0.8	0.8	0.4198	-0.3987	0.1664	0.3877	-0.4080	0.1304	0.3523	-0.3924	0.0724
1	0.5	0.8	1	0.4177	-0.3969	0.1642	0.3859	-0.4060	0.1283	0.3506	-0.3903	0.0707
1	0.5	1	0.8	0.4159	-0.3980	0.1610	0.3837	-0.4073	0.1250	0.3485	-0.3913	0.0672
1	0.5	1	1	0.4139	-0.3963	0.1590	0.3820	-0.4054	0.1230	0.3469	-0.3893	0.0657
1	0.7	0.8	0.8	0.4178	-0.4105	0.1588	0.3843	-0.4228	0.1211	0.3465	-0.4071	0.0591
1	0.7	0.8	1	0.4158	-0.4085	0.1568	0.3825	-0.4206	0.1193	0.3450	-0.4047	0.0578
1	0.7	1	0.8	0.4140	-0.4096	0.1537	0.3804	-0.4218	0.1161	0.3429	-0.4056	0.0544
1	0.7	1	1	0.4122	-0.4076	0.1520	0.3788	-0.4196	0.1144	0.3415	-0.4033	0.0533
1	1	0.8	0.8	0.4145	-0.4263	0.1481	0.3789	-0.4428	0.1080	0.3378	-0.4267	0.0406
1	1	0.8	1	0.4128	-0.4240	0.1466	0.3774	-0.4402	0.1066	0.3366	-0.4240	0.0398
1	1	1	0.8	0.4110	-0.4250	0.1436	0.3753	-0.4413	0.1034	0.3346	-0.4249	0.0366
1	1	1	1	0.4094	-0.4228	0.1422	0.3739	-0.4388	0.1021	0.3335	-0.4222	0.0359

Table 5.9: Correlations in comparison to the reference and different classifications of level of the piece Alla Turca.

Weights				Normal			Confidence			Threshold		
Finger	Velocity	Wrist	Palm	OMSI	Lessons	Experience	OMSI	Lessons	Experience	OMSI	Lessons	Experience
0.5	0.5	0.8	0.8	0.5479	-0.2247	0.3891	0.4356	-0.0450	0.1934	0.2939	-0.0095	0.0003
0.5	0.5	0.8	1	0.5506	-0.2104	0.3866	0.4374	-0.0289	0.1906	0.2976	0.0034	0.0013
0.5	0.5	1	0.8	0.5606	-0.2163	0.3937	0.4489	-0.0348	0.1981	0.3073	-0.0016	0.0067
0.5	0.5	1	1	0.5625	-0.2026	0.3909	0.4500	-0.0194	0.1951	0.3103	0.0107	0.0074
0.5	0.7	0.8	0.8	0.5384	-0.3331	0.4208	0.4397	-0.1763	0.2373	0.2875	-0.1142	0.0167
0.5	0.7	0.8	1	0.5429	-0.3188	0.4193	0.4432	-0.1587	0.2345	0.2918	-0.1002	0.0172
0.5	0.7	1	0.8	0.5514	-0.3241	0.4253	0.4533	-0.1646	0.2413	0.3003	-0.1053	0.0220
0.5	0.7	1	1	0.5552	-0.3103	0.4235	0.4560	-0.1475	0.2383	0.3040	-0.0917	0.0223
0.5	1	0.8	0.8	0.5132	-0.4367	0.4419	0.4275	-0.3115	0.2756	0.2732	-0.2249	0.0341
0.5	1	0.8	1	0.5193	-0.4247	0.4419	0.4328	-0.2956	0.2738	0.2779	-0.2117	0.0342
0.5	1	1	0.8	0.5259	-0.4290	0.4466	0.4408	-0.3008	0.2794	0.2848	-0.2163	0.0383
0.5	1	1	1	0.5314	-0.4171	0.4463	0.4454	-0.2851	0.2774	0.2891	-0.2034	0.0384
0.7	0.5	0.8	0.8	0.5249	-0.1444	0.3469	0.4001	0.0445	0.1445	0.2718	0.0626	-0.0258
0.7	0.5	0.8	1	0.5270	-0.1341	0.3453	0.4020	0.0555	0.1433	0.2752	0.0715	-0.0242
0.7	0.5	1	0.8	0.5359	-0.1391	0.3515	0.4118	0.0509	0.1495	0.2834	0.0676	-0.0198
0.7	0.5	1	1	0.5374	-0.1291	0.3498	0.4131	0.0614	0.1481	0.2863	0.0761	-0.0184
0.7	0.7	0.8	0.8	0.5269	-0.2526	0.3853	0.4147	-0.0784	0.1904	0.2719	-0.0356	-0.0090
0.7	0.7	0.8	1	0.5301	-0.2412	0.3839	0.4173	-0.0651	0.1886	0.2755	-0.0250	-0.0080
0.7	0.7	1	0.8	0.5379	-0.2459	0.3894	0.4263	-0.0698	0.1944	0.2829	-0.0291	-0.0039
0.7	0.7	1	1	0.5407	-0.2348	0.3879	0.4283	-0.0570	0.1925	0.2861	-0.0189	-0.0031
0.7	1	0.8	0.8	0.5141	-0.3667	0.4172	0.4168	-0.2198	0.2374	0.2653	-0.1486	0.0107
0.7	1	0.8	1	0.5187	-0.3560	0.4168	0.4206	-0.2061	0.2357	0.2693	-0.1377	0.0112
0.7	1	1	0.8	0.5250	-0.3601	0.4213	0.4281	-0.2107	0.2408	0.2755	-0.1416	0.0148
0.7	1	1	1	0.5291	-0.3495	0.4207	0.4315	-0.1973	0.2390	0.2792	-0.1309	0.0152
1	0.5	0.8	0.8	0.4944	-0.0582	0.2983	0.3592	0.1338	0.0935	0.2469	0.1352	-0.0520
1	0.5	0.8	1	0.4963	-0.0517	0.2979	0.3613	0.1402	0.0937	0.2499	0.1406	-0.0501
1	0.5	1	0.8	0.5037	-0.0556	0.3030	0.3691	0.1370	0.0985	0.2565	0.1379	-0.0467
1	0.5	1	1	0.5052	-0.0494	0.3024	0.3709	0.1431	0.0985	0.2593	0.1431	-0.0449
1	0.7	0.8	0.8	0.5061	-0.1583	0.3395	0.3801	0.0286	0.1365	0.2516	0.0499	-0.0367
1	0.7	0.8	1	0.5085	-0.1503	0.3388	0.3823	0.0373	0.1358	0.2547	0.0569	-0.0353
1	0.7	1	0.8	0.5153	-0.1541	0.3435	0.3898	0.0338	0.1405	0.2609	0.0539	-0.0320
1	0.7	1	1	0.5173	-0.1463	0.3426	0.3916	0.0422	0.1398	0.2637	0.0607	-0.0307
1	1	0.8	0.8	0.5073	-0.2756	0.3807	0.3953	-0.1065	0.1871	0.2525	-0.0575	-0.0168
1	1	0.8	1	0.5105	-0.2669	0.3802	0.3980	-0.0962	0.1860	0.2557	-0.0493	-0.0159
1	1	1	0.8	0.5163	-0.2705	0.3843	0.4047	-0.0999	0.1904	0.2611	-0.0524	-0.0129
1	1	1	1	0.5192	-0.2620	0.3836	0.4071	-0.0898	0.1892	0.2641	-0.0444	-0.0122

Chapter 6

Conclusions and Future Work

This chapter will sum up the past chapters pointing out the conclusions made with the work of each one. After that will address the possible future work ideas and changes to improve the solution implemented.

6.1 Conclusions

In the literature review, although various projects were similar to the idea, all of them were focused on the recognition of specific gestures and only their fingertip position verification. So it was only tried to check if the player was doing a specific gesture, mostly done with HMM because of the use of dynamic gestures. Also in the projects where the DTW was utilized, it was mainly in a classification objective and the path originated from the alignment was discarded or used to make a fusion between the two time series. None of the solutions found focused on a comparison of the same movement between two people. But what was found indicated that was possible to arrange combinations of methods or slightly change what has been used to achieve this feature. Also from the sensors found, the ones that would not hinder performance were camera-based sensors, and from those, the Leap motion seemed the best option and as it was available it was chosen.

So the methodological approach delineated was recording the movements data with the Leap Motion and at the same time record the sound of that performance. During the recording would be used HMM or Neural Networks to segment the data in terms of changes between recognized gestures and this segmented data would be the comparison object of the references and tests, trying to give feedback on gesture changes. The DTW would align the test to the reference eliminating the different timings from the equation and this way purely analyzing the gesture done even if it was done slower or faster, giving a dissimilarity factor of this comparison.

After the beginning of implementation, it was realized that the segmentation module would take too much work in terms of defining the specific gestures, so this segmentation was left behind to future work and the module of comparison with DTW was done but considering the full performance. This way, the resulting output was the overall distance of the full performance between the test and the reference. The positioning of the sensor although not ideal, was functional with the setback of being harder to positioning between runs but still possible with the help of the Leap Motion cameras images and the validation feature done in the recorder. Still, it came with a possible maximum error of 0.6 millimeters in the positioning. Even with the limitations found during the implementation, it was considered that this work is still a good representation of the potential use of DTW in this idea.

The experiments done lasted for two days, having achieved a total of eleven participants. Seven performed the Musette and Alla Turca pieces, and ten performed the piece Fur Elise. The only possible error of positioning of the sensor is between the participants from day one in relation to ones from day two. Given this, in the analysis done, it did not seem that a misplacement of the sensor happened or at least it was not that big to be noticed as if there was an error in the positioning it would be expected to participants of the same day have a lower dissimilarity factor result between them in relation to participants from the other day, which was not the case.

The analysis showed that if every participant was compared between them there was no correlation between the dissimilarity and the level in every classification, but if they were compared only to the references there was an indication that could be true that the dissimilarity increases with the level difference but only for the OMSI classification. Although there was still not a clear correlation in all pieces, it is expected that with more prepared reference participants this would be more visible. Also, the algorithm showed to be relevant and good with comparing performances of the same person, so it could be used to assess the consistency of a musician as it is also an indicator of skill level as more skilled musicians will perform a piece with more consistency of movements. Still, this could be more an indicator of how much that specific piece was studied too. Lastly, the analysis showed that the hand confidence is mostly between 70 and 80 percent which can lead to some error factors and this was due to the piano keys reflecting a lot of infrared light and the contrast between the hands and the keys was not clear to the sensor. So if there was a way to decrease these reflections in the piano keys the results would have been more accurate and possibly better. Also, the amount of data loss in the experiments was much less than what was expected to happen based on the tests done before, and most of the cases were due to the participant rest the hand in their lap.

In the end, although there was an expectation for more conclusive results, the proposed solution can still give feedback with good accuracy of positions even if there was not so much correlation between the level and the distance. And the various paths that the solution can be used for shows that its worth research and experiment more with this type of solution, even in different areas and not only musicians.

6.2 Future Work

The implemented solution caresses the segmentation module, and this is an obvious future work to be done. This module could be implemented with HMM considering dynamic gesture recognition or a more basic static recognition with a simpler neural network. Still, this would mean a lot of work in the specification of the gestures needed and could be independent gestures from piece to piece. With the addition of this module, the implemented comparison module should be tweaked in order to consider the use of the DTW results to classify and assign the current segmentation to a specify segmented reference.

Also, a better dataset should be acquired, having the participants for the references better prepared and chosen, ideally having a music school cooperate with this selection for a more informed decision on which person is best fit to serve as a reference, and a more variety of participants levels to test. In this phase, there is the possibility of gathering participants for various styles and not only the classical one, so if the participant wants to play jazz or other styles, the comparison module would use a reference from that specific style. Still, regarding the references, a group of references could be collected and then a fusion of the movements done with the DTW or even the average of positions so it would serve as the reference time series. Also in relation to this, instead of defining the gestures needed, a self-learning machine learning method could be utilized to infer the patterns utilized by the musicians, but this would mean that a lot of training data is needed.

By using the results, the feedback given with the direction vector combined with the distance could be graphically represented in time for an intuitive way of interpretation for the user. This could be tried to do in real-time, which would mean more changes in the solution to achieve that as the analysis of the recordings of the current state is done after collecting all of the data. Still, if the segmentation is done it would be possible to give the feedback right after the recognition as the comparison module does not take a long time to make a single comparison. To this feedback, there could be assigned some threshold values representing an acceptable distance for considering the specific movement a properly perform movement. In other words, if the resulting distance is lower than the correct threshold it would be considered correct. More threshold values could be utilized to give different rankings of those movements like perfect, great, good, acceptable and bad. This could be also applied to the version with the fusion of movements data from various references, where the threshold of perfect would the deviation of the references in comparison to the final time series.

There are various ways to utilize this type of idea, it can be seen of it evolving to various paths. With the ranking option of the movement, it is possible to imagine it as a type of gamification of the learning process, like in Rocksmith, as in those types of applications usually only give feedback after the notes are played. It could be applied as a post-analysis of the performance, building a report with the performance throughout the piece.

Another possibility is using the sound instead of using the gestures to make the alignment of the time series. By using the alignment sound it's possible to see at which time stamps of the sounds were matched and build a module to use those matches to modify the recorded leap motion data, in this way the gesture would be aligned as well and after that it is only needed to give the feedback the same way the current solution does. Still addressing the sound, a musical approach analysis could be made to give further feedback to the player.

Ultimately, another possibility of evolving the work to another path is using the sound analysis and try to correlate the sound features and the movements done, and see if there are patterns of movements to produce a specific type of sound, expressiveness or even analyze the feelings perceived by listeners of the performance and try to discover the patterns of the movements that make people feel different emotions. Obviously, this last one would involve more areas as it brings a component of psychology and much more musical knowledge to the project.

References

[AWB$^+$12] Roland Aigner, Daniel Wigdor, Hrvoje Benko, Michael Haller, David Lindlbauer, Alexandra Ion, Shengdong Zhao, and Jeffrey Koh. Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci. 11 2012.

[BH19] Ahmet Birdal and Reza Hassanpour. Region based hand gesture recognition. 02 2019.

[BLL02] Lars Bretzner, Ivan Laptev, and Tony Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. *Face and Gesture FG'02*, 08 2002.

[Bur98] Christopher Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 01 1998.

[CATA13] S Celebi, A.S. Aydin, T.T. Temiz, and T Arici. Gesture recognition using skeleton data with weighted dynamic time warping. *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications*, 1:620–625, 01 2013.

[FS09] Adrian Freed and Andrew Schmeder. Features and future of open sound control version 1.1 for nime. In *NIME*, volume 4, page 2009, 2009.

[FST98] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[HC17] Panduranga H.T and Mani. C. *Sensors for Virtual Musical Environment: A Short Survey*, pages 148–157. 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017.

[Hol25] GJ Holzmann. Finite state machine: Ebook. *URL: http://www. spinroot. com/spin/-Doc/Book91 PDF*, 1, 1925.

[KKFW10] Dietrich Kammer, Mandy Keck, Georg Freitag, and Markus Wacker. Taxonomy and overview of multi-touch frameworks: Architecture, scope and features. 01 2010.

[KO18] Yoshitaka Kashiwagi and Youji Ochi. A study of left fingering detection using cnn for guitar learning. pages 14–17, 03 2018.

[KS05] Maria Karam and m.c Schraefel. A taxonomy of gestures in human computer interactions. *Electronics and Computer Science*, 01 2005.

[KS07] Chutisant Kerdvibulvech and Hideo Saito. Real-time guitar chord estimation by stereo cameras for supporting guitarists. 01 2007.

[LKZF10] Qingshan Luo, Xiaodong Kong, Guihua Zeng, and Jianping Fan. Human action detection via boosted local motion histograms. *Mach. Vis. Appl.*, 21:377–389, 04 2010.

[LTC16] W. Lu, Z. Tong, and J. Chu. Dynamic hand gesture recognition with leap motion controller, Sep. 2016.

[Mül07] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[OH13] Akiya Oka and Manabu Hashimoto. Marker-less piano fingering recognition using sequential depth images. *FCV 2013 - Proceedings of the 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, pages 1–4, 01 2013.

[PSH97] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):677–695, 1997.

[QMB+02] Francis Quek, David Mcneill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl-Erik Mccullough, and Rashid Ansari. Multimodal human discourse: Gesture and speech. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 9:171–193, 10 2002.

[RDE11] Miguel Reyes, Gabriel Dominguez, and Sergio Escalera. Featureweighting in dynamic timewarping for gesture recognition in depth data. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1182–1188, 11 2011.

[RSS11] Sonia Ranelli, Leon Straker, and Anne Smith. *Playing-related Musculoskeletal Problems in Children Learning Instrumental Music: The Association Between Problem Location and Gender, Age, and Music Exposure Factors*, page 123. Medical Problems of Performing Artists: Volume 26 Number 3, September 2011.

[SC07] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[Sen08] Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40, 2008.

[SMR99] Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. Learning hidden markov model structure for information extraction. In *AAAI-99 workshop on machine learning for information extraction*, pages 37–42, 1999.

[SSA04] Leonid Sigal, Stan Sclaroff, and Vassilis Athitsos. Skin color-based video segmentation under time-varying illumination. ieee trans pattern anal mach intell. *IEEE transactions on pattern analysis and machine intelligence*, 26:862–77, 08 2004.

[Sut14] Potcharapol Suteparuk. Detection of piano keys pressed in video. 2014.

[VdB17] Antal Van den Bosch. *Hidden Markov Models*. 01 2017.

[VM13] Perumal Venkatesan and M Mullai. Self organizing neural network in prediction. 02 2013.

[WA99] Christian Wohler and Joachim K Anlauf. An adaptable time-delay neural-network algorithm for image sequence analysis. *IEEE Transactions on Neural Networks*, 10(6):1531–1536, 1999.

[WBRF13] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.

[XJBMK97] Shanon X. Ju, Michael Black, Scott Minneman, and Don Kimber. Analysis of gesture and action in technical talks for video indexing. 09 1997.

[XMW$^+$18] Weiming Xiang, Patrick Musau, Ayana Wild, Diego Manzanas Lopez, Nathaniel Hamilton, Xiaodong Yang, Joel Rosenfeld, and Taylor Johnson. Verification for machine learning, autonomy, and neural networks survey. 10 2018.

REFERENCES

Appendix A

Questionnaires

Depois das conclusões e antes das referências bibliográficas, apresenta-se neste anexo numerado o texto usado para preencher a dissertação.

A.1 Participants gathering questionnaire

This was the questionnaire created for gathering and know in which day the participant was available to perform so the participants would be assigned to a slot of half an hour in the time table of that day.

Comparison of performance of pianists with different levels

The objective of this form is to gather information about the participants and their availability.

No personal information will be used as they will be referred as anonymous participants.

* Required

1. **Name** *

2. **Surname** *

3. **Email** *

4. **Pieces I would like to play** *
Check all that apply.

- [ ] Bach, J. S. - Musette (beginner)
- [ ] Beethoven, L. V. - Fur Elise, first 50 seconds (beginner)
- [ ] Mozart, W. A. - Alla Turca, first 50 seconds(intermediate)
- [ ] Chopin, F. - Nocturne Op. 15 No. 2 (advanced)

5. **Which day will you preffer to record on?** *
Check all that apply.

- [ ] Wednesday
- [ ] Thursday

A.2 Level assessment and feedback questionnaire

This was the questionnaire created for the level assessment using the OMSI questions and also the feedback about the sensor positioning.

Comparison of performance of pianists with different levels

The objective of this form is to estimate the musician level of each participant and receive their feedback on the experience.

None of the personal information will be used as they will be referred as anonymous participants.

* Required

1. **Name** *

2. **Surname** *

3. **Email** *

4. **Date of birth** *

   Example: December 15, 2012

5. **Pieces you will play** *

   Check all that apply.

   - [ ] Bach, J. S. - Musette (beginner)
   - [ ] Beethoven, L. V. - Fur Elise, first movement (beginner) (about 50s)
   - [ ] Mozart, W. A. - Alla Turca, first movement (intermediate) (about 50s)
   - [ ] Chopin, F. - Nocturne Op. 15 No. 2 (advanced)
   - [ ] None

6. **Pieces that you feel confortable playing** *

   Check all that apply.

   - [ ] Bach, J. S. - Musette (beginner)
   - [ ] Beethoven, L. V. - Fur Elise, first movement (beginner) (about 50s)
   - [ ] Mozart, W. A. - Alla Turca, first movement (intermediate) (about 50s)
   - [ ] Chopin, F. - Nocturne Op. 15 No. 2 (advanced)
   - [ ] None

Ollen Musical Sophistication Index

The Ollen Musical Sophistication Index (OMSI) is a tool to aid researchers in classifying their research participants as more or less musically sophisticated.

In order to obtain your score, please enter an answer for every question unless you are specifically directed to skip one.

7. **At what age did you begin sustained musical activity?** *
"Sustained musical activity" might include regular music lessons or daily musical practice that lasted for at least three consecutive years. If you have never been musically active for a sustained time period, answer with zero.

8. **How many years of private music lessons have you received?** *
If you have received lessons on more than one instrument, including voice, give the number of years for the one instrument/voice you've studied longest.If you have never received private lessons, answer with zero.

9. **For how many years have you engaged in regular, daily practice of a musical instrument or singing?** *
"Daily" can be defined as 5 to 7 days per week. A "year" can be defined as 10 to 12 months. If you have never practiced regularly, or have practiced regularly for fewer than 10 months, answer with zero.

10. **Which category comes nearest to the amount of time you currently spend practicing an instrument (or voice)?** *
Count individual practice time only; no group rehearsals.
Mark only one oval.

- I rarely or never practice singing or playing an instrument
- About 1 hour per month
- About 1 hour per week
- About 15 minutes per day
- About 1 hour per day
- More than 2 hours per day

11. **Have you ever enrolled in any music courses offered at college (or university)?** *
Mark only one oval.

- Yes
- No - Skip next question

12. **(If Yes) How much college-level coursework in music have you completed?If more than one category applies, select your most recently completed level.**
Mark only one oval.

- None
- 1 or 2 NON-major courses (e.g., music appreciation, playing or singing in an ensemble)
- 3 or more courses for NON-majors
- An introductory or preparatiory music program for Bachelor's level work
- 1 year of full-time coursework in a Bachelor of Music degree program (or equivalent)
- 2 years of full-time coursework in a Bachelor of Music degree program (or equivalent)
- 3 or more years of full-time coursework in a Bachelor of Music degree program (or equivalent)
- Completion of a Bachelor of Music degree program (or equivalent)
- One or more graduate-level music courses or degrees

13. **Which option best describes your experience at composing music?** *
Mark only one oval.

- Have never composed any music
- Have composed bits and pieces, but have never completed a piece of music
- Have composed one or more completed pieces, but none have been performed
- Have composed pieces as assignments or projects for one or more music classes; one or more of my pieces have been performed and/or recorded within the context of my educational environment
- Have composed pieces that have been performed for a local audience
- Have composed pieces that have been performed for a regional or national audience (e.g., nationally known performer or ensemble, major concert venue, broadly distributed recording)

14. **To the best of your memory, how many live concerts (or any style, with free or paid admission) have you attended as an audience member in the past 12 months?** *
Please do not include regular religious services in your count, but you may include special musical productions or events.
Mark only one oval.

- None
- 1 - 4
- 5 - 8
- 9 - 12
- 13 or more

15. **Which title best describes you?** *
Mark only one oval.

- Nonmusician
- Music-loving nonmusician
- Amateur musician
- Serious amateur musician
- Semiprofessional musician
- Professional musician

16. **Result**

Feedback

17. **Based on the experience, give your opinion about the following statements**
Mark only one oval per row.

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
The sensor setup didn't hinder my performance					
The sensor setup didn't make reading the sheet harder					
I felt confortable with the setup					
I would use this idea to get feedback on my playing					

18. **In each piece which of the attempts were the two best performances?** *
Check all that apply.

	First	Second	Third	Fourth
Bach, J. S. - Musette				
Beethoven, L. V. - Fur Elise				
Mozart, W. A. - Alla Turca				
Chopin, F. - Nocturne Op. 15 No. 2				

A.3 Comparisons between every participant

This appendix will show a comparison between every participant and the matches resultant to that. It shows only one piece and for one distance function as the quantity of data would be too much to insert here. So in this case is the Fur Elise piece with the normal euclidean distance function and with a radius of 1 and changing the weights values. This is representative of what was done for comparing only with the chosen references as well, as the first column are the references and the row are the tests.

Finger Weight - 0.5 Velocity Weight - 0.8 Wrist Weight - 0.8 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1527008.69	1141201.584	1191445.712	1210705.952	2284877.096	1175486.337	1352522.655	1089660.841	971811.0491	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1527008.69	0	1404658.919	968363.2824	1509512.845	1559033.649	1479381.529	1282710.512	1066209.859	1272260.722	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1141201.584	1404658.919	0	1065211.759	1073253.478	2014152.223	1492823.912	807910.1972	1134779.021	1070481.195	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1191445.712	968363.2824	1065211.759	0	1244423.324	1718933.278	1258038.164	1113618.681	920736.722	1032525.003	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1210705.952	1509512.845	1073253.478	1244423.324	0	2239151.156	1545334.287	1436500.718	1238664.85	1187702.312	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2284877.096	1559033.649	2014152.223	1718933.278	2239151.156	0	2113271.013	1712502.008	1869355.09	2005791.852	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1175486.337	1479381.529	1492823.912	1258038.164	1545334.287	2113271.013	0	1573398.28	1274607.794	1126996.957	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1352522.655	1282710.512	807910.1972	1113618.681	1436500.718	1712502.008	1573398.28	0	1143196.398	1173297.888	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1089660.841	1066209.859	1134779.021	920736.722	1238664.85	1869355.09	1274607.794	1143196.398	0	997561.2892	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	971811.0491	1272260.722	1070481.195	1032525.003	1187702.312	2005791.852	1126996.957	1173297.888	997561.2892	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.5 Velocity Weight - 0.8 Wrist Weight - 0.8 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1576196.886	1173660.17	1227800.865	1243203.496	2362767.489	1207675.291	1397231.661	1119308.41	995662.1327	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1576196.886	0	1450598.958	993022.1155	1555661.791	1603766.494	1520173.925	1324842.854	1094462.442	1310613.743	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1173660.17	1450598.958	0	1100184.698	1104362.648	2081729.905	1537609.631	833280.7374	1166238.22	1099227.814	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1227800.865	993022.1155	1100184.698	0	1280139.45	1773546.214	1293439.799	1150517.801	946323.686	1063090.71	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1243203.496	1555661.791	1104362.648	1280139.45	0	2315595.166	1589202.744	1482779.686	1274309.253	1219510.562	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2362767.489	1603766.494	2081729.905	1773546.214	2315595.166	0	2172908.257	1766172.055	1930446.13	2069766.274	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1207675.291	1520173.925	1537609.631	1293439.799	1589202.744	2172908.257	0	1622148.918	1309780.235	1156617.293	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1397231.661	1324842.854	833280.7374	1150517.801	1482779.686	1766172.055	1622148.918	0	1177777.883	1210223.923	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1119308.41	1094462.442	1166238.22	946323.686	1274309.253	1930446.13	1309780.235	1177777.883	0	1021094.739	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	995662.1327	1310613.743	1099227.814	1063090.71	1219510.562	2069766.274	1156617.293	1210223.923	1021094.739	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.5 Velocity Weight - 0.8 Wrist Weight - 1.0 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1572663.076	1173239.783	1227745.804	1241451.286	2359135.826	1208244.429	1393482.315	1119143.386	996520.647	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1572663.076	0	1447548.925	996389.8555	1555675.648	1608583.479	1522457.288	1322864.876	1096161.486	1311571.046	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1173239.783	1447548.925	0	1099250.797	1105310.137	2081584.576	1539840.529	834520.8871	1166796.828	1101617.731	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1227745.804	996389.8555	1099250.797	0	1282944.851	1777246.233	1296647.596	1150156.743	948921.7183	1067087.377	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1241451.286	1555675.648	1105310.137	1282944.851	0	2316562.385	1592205.624	1479951.995	1275317.493	1219597.243	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2359135.826	1608583.479	2081584.576	1777246.233	2316562.385	0	2174526.555	1768289.041	1930013.604	2072890.99	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1208244.429	1522457.288	1539840.529	1296647.596	1592205.624	2174526.555	0	1620928.18	1311406.946	1159699.719	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1393482.315	1322864.876	834520.8871	1150156.743	1479951.995	1768289.041	1620928.18	0	1176856.793	1209009.733	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1119143.386	1096161.486	1166796.828	948921.7183	1275317.493	1930013.604	1311406.946	1176856.793	0	1021878.421	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	996520.647	1311571.046	1101617.731	1067087.377	1219597.243	2072890.99	1159699.719	1209009.733	1021878.421	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.5 Velocity Weight - 0.8 Wrist Weight - 1.0 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1621851.272	1205698.37	1264100.957	1273948.83	2437026.219	1240433.382	1438191.321	1148790.954	1020371.731	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1621851.272	0	1493488.965	1021048.689	1601824.593	1653316.324	1563249.683	1364997.218	1124414.069	1349924.066	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1205698.37	1493488.965	0	1134223.736	1136419.306	2149162.258	1584626.248	859891.4273	1198256.028	1130364.349	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1264100.957	1021048.689	1134223.736	0	1318660.977	1831859.168	1332049.232	1187055.863	974508.6823	1097653.084	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1273948.83	1601824.593	1136419.306	1318660.977	0	2393006.395	1636074.081	1526230.963	1310961.896	1251405.494	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2437026.219	1653316.324	2149162.258	1831859.168	2393006.395	0	2234163.799	1821959.088	1991104.645	2136865.413	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1240433.382	1563249.683	1584626.248	1332049.232	1636074.081	2234163.799	0	1669678.818	1346579.388	1189320.055	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1438191.321	1364997.218	859891.4273	1187055.863	1526230.963	1821959.088	1669678.818	0	1211438.277	1245935.768	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1148790.954	1124414.069	1198256.028	974508.6823	1310961.896	1991104.645	1346579.388	1211438.277	0	1045411.871	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1020371.731	1349924.066	1130364.349	1097653.084	1251405.494	2136865.413	1189320.055	1245935.768	1045411.871	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.5 Velocity Weight - 1.0 Wrist Weight - 0.8 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1648924.966	1246561.008	1299989.392	1336629.557	2453476.503	1297579.304	1454136.23	1198608.915	1079218.024	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1648924.966	0	1503633.635	1074470.617	1630722.015	1713275.832	1629100.223	1386366.049	1185910.481	1389611.551	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1246561.008	1503633.635	0	1156418.476	1168672.656	2153340.539	1625618.526	867895.4151	1244474.003	1180306.382	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1299989.392	1074470.617	1156418.476	0	1361844.315	1881360.494	1392207.144	1208255.478	1024342.2	1137319.524	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1336629.557	1630722.015	1168672.656	1361844.315	0	2390256.222	1703285.703	1548154.427	1354766.245	1311978.852	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2453476.503	1713275.832	2153340.539	1881360.494	2390256.222	0	2314351.331	1852080.822	2048116.191	2176259.096	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1297579.304	1629100.223	1625618.526	1392207.144	1703285.703	2314351.331	0	1703571.158	1407584.084	1254030.579	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1454136.23	1386366.049	867895.4151	1208255.478	1548154.427	1852080.822	1703571.158	0	1250605.491	1272553.629	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1198608.915	1185910.481	1244474.003	1024342.2	1354766.245	2048116.191	1407584.084	1250605.491	0	1114506.806	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1079218.024	1389611.551	1180306.382	1137319.524	1311978.852	2176259.096	1254030.579	1272553.629	1114506.806	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.5 Velocity Weight - 1.0 Wrist Weight - 0.8 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1698113.161	1279019.594	1336344.545	1369127.101	2531366.897	1329768.257	1498845.236	1228256.484	1103069.108	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1698113.161	0	1549573.674	1099129.45	1676870.961	1758008.677	1669892.618	1428498.392	1214163.063	1427964.572	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1279019.594	1549573.674	0	1191391.415	1199781.825	2220918.221	1670404.245	893265.9553	1275933.202	1209053	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1336344.545	1099129.45	1191391.415	0	1397560.442	1935973.429	1427608.779	1245154.598	1049929.164	1167885.231	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1369127.101	1676870.961	1199781.825	1397560.442	0	2466700.232	1747154.16	1594433.395	1390410.648	1343787.102	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2531366.897	1758008.677	2220918.221	1935973.429	2466700.232	0	2373988.576	1905750.868	2109207.232	2240233.519	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1329768.257	1669892.618	1670404.245	1427608.779	1747154.16	2373988.576	0	1752321.796	1442756.526	1283650.915	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1498845.236	1428498.392	893265.9553	1245154.598	1594433.395	1905750.868	1752321.796	0	1285186.976	1309479.664	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1228256.484	1214163.063	1275933.202	1049929.164	1390410.648	2109207.232	1442756.526	1285186.976	0	1138040.255	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1103069.108	1427964.572	1209053	1167885.231	1343787.102	2240233.519	1283650.915	1309479.664	1138040.255	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.5 Velocity Weight - 1.0 Wrist Weight - 1.0 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1694579.352	1278599.207	1336289.484	1367374.891	2527735.233	1330337.395	1495095.89	1228091.46	1103927.622	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1694579.352	0	1546523.641	1102497.19	1676884.818	1762825.662	1672175.981	1426520.414	1215862.108	1428921.874	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1278599.207	1546523.641	0	1190457.515	1200729.315	2220772.892	1672635.144	894506.105	1276491.81	1211442.917	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1336289.484	1102497.19	1190457.515	0	1400365.842	1939673.448	1430816.576	1244793.54	1052527.196	1171881.898	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1367374.891	1676884.818	1200729.315	1400365.842	0	2467667.451	1750157.041	1591605.704	1391418.888	1343873.783	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2527735.233	1762825.662	2220772.892	1939673.448	2467667.451	0	2375606.873	1907867.855	2108774.706	2243358.234	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1330337.395	1672175.981	1672635.144	1430816.576	1750157.041	2375606.873	0	1751101.058	1444383.237	1286733.341	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1495095.89	1426520.414	894506.105	1244793.54	1591605.704	1907867.855	1751101.058	0	1284265.885	1308265.474	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1228091.46	1215862.108	1276491.81	1052527.196	1391418.888	2108774.706	1444383.237	1284265.885	0	1138823.937	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1103927.622	1428921.874	1211442.917	1171881.898	1343873.783	2243358.234	1286733.341	1308265.474	1138823.937	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.5 Velocity Weight - 1.0 Wrist Weight - 1.0 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1743767.547	1311057.794	1372644.636	1399872.435	2605625.627	1362526.349	1539804.896	1257739.028	1127778.706	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1743767.547	0	1592463.68	1127156.023	1723033.764	1807558.508	1712968.377	1468652.756	1244114.69	1467274.895	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1311057.794	1592463.68	0	1225430.454	1231838.484	2288350.574	1717420.863	919876.6452	1307951.01	1240189.536	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1372644.636	1127156.023	1225430.454	0	1436081.969	1994286.384	1466218.212	1281692.66	1078114.16	1202447.605	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1399872.435	1723033.764	1231838.484	1436081.969	0	2544111.46	1794025.497	1637884.672	1427063.291	1375682.034	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2605625.627	1807558.508	2288350.574	1994286.384	2544111.46	0	2435244.118	1961537.901	2169865.746	2307332.657	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1362526.349	1712968.377	1717420.863	1466218.212	1794025.497	2435244.118	0	1799851.696	1479555.679	1316353.677	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1539804.896	1468652.756	919876.6452	1281692.66	1637884.672	1961537.901	1799851.696	0	1318847.37	1345191.509	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1257739.028	1244114.69	1307951.01	1078114.16	1427063.291	2169865.746	1479555.679	1318847.37	0	1162357.387	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1127778.706	1467274.895	1240189.536	1202447.605	1375682.034	2307332.657	1316353.677	1345191.509	1162357.387	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 0.8 Wrist Weight - 0.8 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1790997.995	1325912.281	1378105.719	1392321.958	2685630.283	1346416.854	1593880.133	1256600.079	1110987.218	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1790997.995	0	1666034.867	1101640.21	1771684.513	1785007.335	1697395.185	1498287.125	1208046.074	1469142.334	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1325912.281	1666034.867	0	1234946.549	1248818.861	2381095.751	1730598.355	951927.9593	1311615.447	1227140.329	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1378105.719	1101640.21	1234946.549	0	1435538.822	1965941.621	1428165.351	1290147.788	1037227.51	1173658.842	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1392321.958	1771684.513	1248818.861	1435538.822	0	2646875.133	1765562.065	1688886.678	1432693.284	1362015.68	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2685630.283	1785007.335	2381095.751	1965941.621	2646875.133	0	2443422.451	1999045.384	2136280.076	2325643.304	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1346416.854	1697395.185	1730598.355	1428165.351	1765562.065	2443422.451	0	1840432.126	1456534.294	1274824.989	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1593880.133	1498287.125	951927.9593	1290147.788	1688886.678	1999045.384	1840432.126	0	1319433.403	1367587.25	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1256600.079	1208046.074	1311615.447	1037227.51	1432693.284	2136280.076	1456534.294	1319433.403	0	1132912.048	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1110987.218	1469142.334	1227140.329	1173658.842	1362015.68	2325643.304	1274824.989	1367587.25	1132912.048	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 0.8 Wrist Weight - 0.8 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1840186.19	1358370.867	1414460.871	1424819.503	2763520.677	1378605.807	1638589.139	1286247.647	1134838.302	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1840186.19	0	1711974.906	1126299.043	1817833.459	1829740.18	1738187.581	1540419.468	1236298.657	1507495.355	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1358370.867	1711974.906	0	1269919.488	1279928.03	2448673.434	1775384.074	977298.4995	1343074.647	1255886.947	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1414460.871	1126299.043	1269919.488	0	1471254.948	2020554.557	1463566.987	1327046.908	1062814.474	1204224.549	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1424819.503	1817833.459	1279928.03	1471254.948	0	2723319.142	1809430.522	1735165.646	1468337.687	1393823.931	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2763520.677	1829740.18	2448673.434	2020554.557	2723319.142	0	2503059.695	2052715.43	2197371.116	2389617.727	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1378605.807	1738187.581	1775384.074	1463566.987	1809430.522	2503059.695	0	1889182.764	1491706.736	1304445.325	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1638589.139	1540419.468	977298.4995	1327046.908	1735165.646	2052715.43	1889182.764	0	1354014.888	1404513.285	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1286247.647	1236298.657	1343074.647	1062814.474	1468337.687	2197371.116	1491706.736	1354014.888	0	1156445.498	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1134838.302	1507495.355	1255886.947	1204224.549	1393823.931	2389617.727	1304445.325	1404513.285	1156445.498	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 0.8 Wrist Weight - 1.0 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1836652.381	1357950.481	1414405.81	1423067.293	2759889.013	1379174.945	1634839.793	1286082.623	1135696.816	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1836652.381	0	1708924.873	1129666.783	1817847.315	1834557.165	1740470.944	1538441.489	1237997.701	1508452.657	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1357950.481	1708924.873	0	1268985.588	1280875.52	2448528.104	1777614.973	978538.6492	1343633.254	1258276.864	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1414405.81	1129666.783	1268985.588	0	1474060.349	2024254.575	1466774.784	1326685.85	1065412.506	1208221.216	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1423067.293	1817847.315	1280875.52	1474060.349	0	2724286.361	1812433.402	1732337.955	1469345.928	1393910.612	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2759889.013	1834557.165	2448528.104	2024254.575	2724286.361	0	2504677.993	2054832.416	2196938.59	2392742.442	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1379174.945	1740470.944	1777614.973	1466774.784	1812433.402	2504677.993	0	1887962.026	1493333.447	1307527.75	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1634839.793	1538441.489	978538.6492	1326685.85	1732337.955	2054832.416	1887962.026	0	1353093.797	1403299.095	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1286082.623	1237997.701	1343633.254	1065412.506	1469345.928	2196938.59	1493333.447	1353093.797	0	1157229.18	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1135696.816	1508452.657	1258276.864	1208221.216	1393910.612	2392742.442	1307527.75	1403299.095	1157229.18	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 0.8 Wrist Weight - 1.0 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1885840.576	1390409.067	1450760.963	1455564.837	2837779.407	1411363.898	1679548.799	1315730.192	1159547.899	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1885840.576	0	1754864.913	1154325.616	1863996.261	1879290.01	1781263.339	1580573.832	1266250.284	1546805.678	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1390409.067	1754864.913	0	1303958.527	1311984.689	2516105.786	1822400.692	1003909.189	1375092.454	1287023.482	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1450760.963	1154325.616	1303958.527	0	1509776.475	2078867.511	1502176.42	1363584.97	1090999.47	1238786.923	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1455564.837	1863996.261	1311984.689	1509776.475	0	2800730.371	1856301.859	1778616.923	1504990.331	1425718.863	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2837779.407	1879290.01	2516105.786	2078867.511	2800730.371	0	2564315.237	2108502.463	2258029.631	2456716.865	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1411363.898	1781263.339	1822400.692	1502176.42	1856301.859	2564315.237	0	1936712.664	1528505.889	1337148.087	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1679548.799	1580573.832	1003909.189	1363584.97	1778616.923	2108502.463	1936712.664	0	1387675.282	1440225.13	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1315730.192	1266250.284	1375092.454	1090999.47	1504990.331	2258029.631	1528505.889	1387675.282	0	1180762.63	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1159547.899	1546805.678	1287023.482	1238786.923	1425718.863	2456716.865	1337148.087	1440225.13	1180762.63	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 1.0 Wrist Weight - 0.8 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1912914.271	1431271.705	1486649.398	1518245.564	2854229.691	1468509.82	1695493.708	1365548.153	1218394.193	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1912914.271	0	1765009.583	1207747.544	1892893.683	1939249.518	1847113.879	1601942.663	1327746.696	1586493.163	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1431271.705	1765009.583	0	1326153.267	1344238.039	2520284.067	1863392.969	1011913.177	1421310.429	1336965.515	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1486649.398	1207747.544	1326153.267	0	1552959.813	2128368.836	1562334.332	1384784.585	1140832.988	1278453.363	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1518245.564	1892893.683	1344238.039	1552959.813	0	2797980.198	1923513.482	1800540.387	1548794.679	1486292.221	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2854229.691	1939249.518	2520284.067	2128368.836	2797980.198	0	2644502.769	2138624.197	2315041.177	2496110.548	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1468509.82	1847113.879	1863392.969	1562334.332	1923513.482	2644502.769	0	1970605.004	1589510.585	1401858.61	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1695493.708	1601942.663	1011913.177	1384784.585	1800540.387	2138624.197	1970605.004	0	1426842.496	1466842.991	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1365548.153	1327746.696	1421310.429	1140832.988	1548794.679	2315041.177	1589510.585	1426842.496	0	1249857.565	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1218394.193	1586493.163	1336965.515	1278453.363	1486292.221	2496110.548	1401858.61	1466842.991	1249857.565	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 1.0 Wrist Weight - 0.8 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1962102.466	1463730.291	1523004.551	1550743.108	2932120.085	1500698.774	1740202.714	1395195.721	1242245.277	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1962102.466	0	1810949.622	1232406.378	1939042.629	1983982.363	1887906.274	1644075.005	1355999.278	1624846.183	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1463730.291	1810949.622	0	1361126.206	1375347.208	2587861.749	1908178.688	1037283.717	1452769.629	1365712.134	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1523004.551	1232406.378	1361126.206	0	1588675.94	2182981.772	1597735.967	1421683.705	1166419.952	1309019.07	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1550743.108	1939042.629	1375347.208	1588675.94	0	2874424.208	1967381.938	1846819.355	1584439.082	1518100.471	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2932120.085	1983982.363	2587861.749	2182981.772	2874424.208	0	2704140.014	2192294.243	2376132.218	2560084.971	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1500698.774	1887906.274	1908178.688	1597735.967	1967381.938	2704140.014	0	2019355.642	1624683.027	1431478.947	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1740202.714	1644075.005	1037283.717	1421683.705	1846819.355	2192294.243	2019355.642	0	1461423.981	1503769.026	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1395195.721	1355999.278	1452769.629	1166419.952	1584439.082	2376132.218	1624683.027	1461423.981	0	1273391.014	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1242245.277	1624846.183	1365712.134	1309019.07	1518100.471	2560084.971	1431478.947	1503769.026	1273391.014	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 1.0 Wrist Weight - 1.0 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	1958568.657	1463309.905	1522949.49	1548990.898	2928488.421	1501267.912	1736453.368	1395030.697	1243103.791	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	1958568.657	0	1807899.589	1235774.118	1939056.486	1988799.348	1890189.637	1642097.027	1357698.323	1625803.486	fur-p4-2	fur-p4-2	fur-p4-2

fur-p3-4	1463309.905	1807899.589	0	1360192.305	1376294.697	2587716.42	1910409.587	1038523.867	1453328.236	1368102.051	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1522949.49	1235774.118	1360192.305	0	1591481.34	2186681.79	1600943.764	1421322.647	1169017.984	1313015.737	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1548990.898	1939056.486	1376294.697	1591481.34	0	2875391.427	1970384.819	1843991.664	1585447.322	1518187.152	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	2928488.421	1988799.348	2587716.42	2186681.79	2875391.427	0	2705758.311	2194411.23	2375699.692	2563209.687	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1501267.912	1890189.637	1910409.587	1600943.764	1970384.819	2705758.311	0	2018134.905	1626309.738	1434561.372	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1736453.368	1642097.027	1038523.867	1421322.647	1843991.664	2194411.23	2018134.905	0	1460502.89	1502554.836	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1395030.697	1357698.323	1453328.236	1169017.984	1585447.322	2375699.692	1626309.738	1460502.89	0	1274174.696	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1243103.791	1625803.486	1368102.051	1313015.737	1518187.152	2563209.687	1434561.372	1502554.836	1274174.696	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 0.7 Velocity Weight - 1.0 Wrist Weight - 1.0 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2007756.852	1495768.491	1559304.643	1581488.442	3006378.815	1533456.865	1781162.373	1424678.266	1266954.875	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2007756.852	0	1853839.629	1260432.951	1985205.431	2033532.193	1930982.033	1684229.37	1385950.905	1664156.507	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1495768.491	1853839.629	0	1395165.245	1407403.867	2655294.102	1955195.306	1063894.407	1484787.436	1396848.669	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1559304.643	1260432.951	1395165.245	0	1627197.467	2241294.726	1636345.4	1458221.767	1194604.948	1343581.444	fur-p2-4	fur-p10-3	fur-p9-2
Fur-p5-1	1581488.442	1985205.431	1407403.867	1627197.467	0	2951835.437	2014253.276	1890270.632	1621091.725	1549995.403	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3006378.815	2033532.193	2655294.102	2241294.726	2951835.437	0	2765395.556	2248081.276	2436790.732	2627184.109	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1533456.865	1930982.033	1955195.306	1636345.4	2014253.276	2765395.556	0	2066885.542	1661482.18	1464181.708	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1781162.373	1684229.37	1063894.407	1458221.767	1890270.632	2248081.276	2066885.542	0	1495084.375	1539480.871	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1424678.266	1385950.905	1484787.436	1194604.948	1621091.725	2436790.732	1661482.18	1495084.375	0	1297708.146	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1266954.875	1664156.507	1396848.669	1343581.444	1549995.403	2627184.109	1464181.708	1539480.871	1297708.146	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 0.8 Wrist Weight - 0.8 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2186981.952	1602978.327	1658095.728	1664745.969	3286760.065	1602812.628	1955916.35	1507008.935	1319751.471	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2186981.952	0	2058098.789	1301555.601	2164942.015	2123967.863	2024415.669	1821652.045	1420800.397	1764464.751	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1602978.327	2058098.789	0	1489548.735	1512166.935	2931511.044	2087260.02	1167954.602	1576870.087	1462129.029	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1658095.728	1301555.601	1489548.735	0	1722212.069	2336454.135	1683356.133	1554941.449	1211963.691	1385359.601	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1664745.969	2164942.015	1512166.935	1722212.069	0	3258461.097	2095903.733	2067465.618	1723735.936	1623485.734	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3286760.065	2123967.863	2931511.044	2336454.135	3258461.097	0	2938649.608	2428860.446	2536667.555	2805420.483	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1602812.628	2024415.669	2087260.02	1683356.133	2095903.733	2938649.608	0	2240982.896	1729424.046	1496567.036	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1955916.35	1821652.045	1167954.602	1554941.449	2067465.618	2428860.446	2240982.896	0	1583788.911	1659021.293	fur-p3-4	fur-p4-2	fur-p3-4
fur-p9-2	1507008.935	1420800.397	1576870.087	1211963.691	1723735.936	2536667.555	1729424.046	1583788.911	0	1335938.187	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1319751.471	1764464.751	1462129.029	1385359.601	1623485.734	2805420.483	1496567.036	1659021.293	1335938.187	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 0.8 Wrist Weight - 0.8 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2236170.147	1635436.913	1694450.881	1697243.513	3364650.459	1635001.582	2000625.356	1536656.503	1343602.555	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2236170.147	0	2104038.829	1326214.434	2211090.96	2168700.708	2065208.065	1863784.388	1449052.979	1802817.772	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1635436.913	2104038.829	0	1524521.675	1543276.104	2999088.726	2132045.739	1193325.143	1608329.286	1490875.647	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1694450.881	1326214.434	1524521.675	0	1757928.195	2391067.07	1718757.769	1591840.569	1237550.655	1415925.307	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1697243.513	2211090.96	1543276.104	1757928.195	0	3334905.107	2139772.189	2113744.586	1759380.339	1655293.984	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3364650.459	2168700.708	2999088.726	2391067.07	3334905.107	0	2998286.852	2482530.493	2597758.595	2869394.906	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1635001.582	2065208.065	2132045.739	1718757.769	2139772.189	2998286.852	0	2289733.533	1764596.488	1526187.372	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	2000625.356	1863784.388	1193325.143	1591840.569	2113744.586	2482530.493	2289733.533	0	1618370.395	1695947.328	fur-p3-4	fur-p4-2	fur-p3-4
fur-p9-2	1536656.503	1449052.979	1608329.286	1237550.655	1759380.339	2597758.595	1764596.488	1618370.395	0	1359471.636	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1343602.555	1802817.772	1490875.647	1415925.307	1655293.984	2869394.906	1526187.372	1695947.328	1359471.636	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 0.8 Wrist Weight - 1.0 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2232636.338	1635016.527	1694395.82	1695491.303	3361018.795	1635570.72	1996876.01	1536491.479	1344461.069	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2232636.338	0	2100988.796	1329582.174	2211104.817	2173517.694	2067491.427	1861806.409	1450752.024	1803775.075	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1635016.527	2100988.796	0	1523587.774	1544223.594	2998943.396	2134276.638	1194565.292	1608887.894	1493265.564	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1694395.82	1329582.174	1523587.774	0	1760733.596	2394767.089	1721965.566	1591479.51	1240148.687	1419921.975	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1695491.303	2211104.817	1544223.594	1760733.596	0	3335872.326	2142775.07	2110916.896	1760388.579	1655380.666	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3361018.795	2173517.694	2998943.396	2394767.089	3335872.326	0	2999905.15	2484647.479	2597326.069	2872519.621	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1635570.72	2067491.427	2134276.638	1721965.566	2142775.07	2999905.15	0	2288512.796	1766223.198	1529269.798	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	1996876.01	1861806.409	1194565.292	1591479.51	2110916.896	2484647.479	2288512.796	0	1617449.305	1694733.138	fur-p3-4	fur-p4-2	fur-p3-4
fur-p9-2	1536491.479	1450752.024	1608887.894	1240148.687	1760388.579	2597326.069	1766223.198	1617449.305	0	1360255.318	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1344461.069	1803775.075	1493265.564	1419921.975	1655380.666	2872519.621	1529269.798	1694733.138	1360255.318	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 0.8 Wrist Weight - 1.0 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2281824.533	1667475.113	1730750.973	1727988.847	3438909.189	1667759.673	2041585.015	1566139.048	1368312.153	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2281824.533	0	2146928.835	1354241.007	2257253.763	2218250.539	2108283.823	1903938.752	1479004.606	1842128.096	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1667475.113	2146928.835	0	1558560.713	1575332.763	3066521.079	2179062.356	1219935.833	1640347.093	1522012.182	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1730750.973	1354241.007	1558560.713	0	1796449.722	2449380.025	1757367.202	1628378.63	1265735.651	1450487.681	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1727988.847	2257253.763	1575332.763	1796449.722	0	3412316.335	2186643.526	2157195.864	1796032.982	1687188.916	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3438909.189	2218250.539	3066521.079	2449380.025	3412316.335	0	3059542.394	2538317.525	2658417.11	2936494.044	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1667759.673	2108283.823	2179062.356	1757367.202	2186643.526	3059542.394	0	2337263.433	1801395.64	1558890.134	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	2041585.015	1903938.752	1219935.833	1628378.63	2157195.864	2538317.525	2337263.433	0	1652030.789	1731659.173	fur-p3-4	fur-p4-2	fur-p3-4
fur-p9-2	1566139.048	1479004.606	1640347.093	1265735.651	1796032.982	2658417.11	1801395.64	1652030.789	0	1383788.768	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1368312.153	1842128.096	1522012.182	1450487.681	1687188.916	2936494.044	1558890.134	1731659.173	1383788.768	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 1.0 Wrist Weight - 0.8 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2308898.228	1708337.751	1766639.407	1790669.574	3455359.473	1724905.595	2057529.925	1615957.009	1427158.446	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2308898.228	0	2157073.505	1407662.935	2286151.185	2278210.046	2174134.362	1925307.583	1540501.018	1881815.58	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1708337.751	2157073.505	0	1580755.453	1607586.113	3070699.359	2220054.634	1227939.82	1686565.068	1571954.215	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1766639.407	1407662.935	1580755.453	0	1839633.06	2498881.35	1817525.114	1649578.245	1315569.169	1490154.122	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1790669.574	2286151.185	1607586.113	1839633.06	0	3409566.163	2253855.149	2179119.327	1839837.331	1747762.274	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3455359.473	2278210.046	3070699.359	2498881.35	3409566.163	0	3139729.926	2568439.26	2715428.656	2975887.727	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1724905.595	2174134.362	2220054.634	1817525.114	2253855.149	3139729.926	0	2371155.774	1862400.337	1623600.658	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	2057529.925	1925307.583	1227939.82	1649578.245	2179119.327	2568439.26	2371155.774	0	1691198.003	1758277.034	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1615957.009	1540501.018	1686565.068	1315569.169	1839837.331	2715428.656	1862400.337	1691198.003	0	1452883.703	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1427158.446	1881815.58	1571954.215	1490154.122	1747762.274	2975887.727	1623600.658	1758277.034	1452883.703	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 1.0 Wrist Weight - 0.8 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2358086.423	1740796.337	1802994.56	1823167.118	3533249.867	1757094.548	2102238.931	1645604.577	1451009.53	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2358086.423	0	2203013.545	1432321.768	2332300.131	2322942.891	2214926.758	1967439.926	1568753.601	1920168.601	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1740796.337	2203013.545	0	1615728.392	1638695.282	3138277.041	2264840.353	1253310.161	1718024.268	1600700.843	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1802994.56	1432321.768	1615728.392	0	1875349.187	2553494.286	1852926.749	1686477.365	1341156.133	1520719.828	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1823167.118	2332300.131	1638695.282	1875349.187	0	3486010.173	2297723.606	2225398.295	1875481.734	1779570.525	fur-p1-3	fur-p10-3	fur-p3-4

fur-p6-2	3533249.867	2322942.891	3138277.041	2553494.286	3486010.173	0	3199367.171	2622109.306	2776519.697	3039862.15	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1757094.548	2214926.758	2264840.353	1852926.749	2297723.606	3199367.171	0	2419906.411	1897572.778	1653220.994	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	2102238.931	1967439.926	1253310.361	1686477.365	2225398.295	2622109.306	2419906.411	0	1725779.488	1795203.069	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1645604.577	1568753.601	1718024.268	1341156.133	1875481.734	2776519.697	1897572.778	1725779.488	0	1476417.153	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1451009.53	1920168.601	1600700.834	1520719.828	1779570.525	3039862.15	1653220.994	1795203.069	1476417.153	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 1.0 Wrist Weight - 1.0 Palm Weight - 0.8

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2354552.614	1740375.951	1802939.499	1821414.908	3529618.203	1757663.686	2098489.585	1645439.553	1451868.044	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2354552.614	0	2199963.512	1435689.508	2332313.987	2327759.877	2217210.121	1965461.947	1570452.645	1921125.904	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1740375.951	2199963.512	0	1614794.492	1639642.771	3138131.712	2267071.252	1254550.51	1718582.876	1603090.751	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1802939.499	1435689.508	1614794.492	0	1878154.587	2557194.304	1856134.546	1686116.307	1343754.165	1524716.496	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1821414.908	2332313.987	1639642.771	1878154.587	0	3486977.392	2300726.486	2222570.604	1876489.974	1779657.206	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3529618.203	2327759.877	3138131.712	2557194.304	3486977.392	0	3200985.468	2624226.292	2776087.171	3042986.865	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1757663.686	2217210.121	2267071.252	1856134.546	2300726.486	3200985.468	0	2418685.674	1899199.489	1656303.419	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	2098489.585	1965461.947	1254550.51	1686116.307	2222570.604	2624226.292	2418685.674	0	1724858.397	1793988.879	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1645439.553	1570452.645	1718582.876	1343754.165	1876489.974	2776087.171	1899199.489	1724858.397	0	1477200.835	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1451868.044	1921125.904	1603090.751	1524716.496	1779657.206	3042986.865	1656303.419	1793988.879	1477200.835	0	fur-p1-3	fur-p9-2	fur-p1-3

Finger Weight - 1.0 Velocity Weight - 1.0 Wrist Weight - 1.0 Palm Weight - 1.0

	fur-p1-3	fur-p2-4	fur-p3-4	fur-p4-2	Fur-p5-1	fur-p6-2	fur-p7-4	fur-p8-3	fur-p9-2	fur-p10-3	match left hand	match right hand	match total
fur-p1-3	0	2403740.809	1772834.537	1839294.652	1853912.452	3607508.597	1789852.64	2143198.59	1675087.122	1475719.128	fur-p10-3	fur-p10-3	fur-p10-3
fur-p2-4	2403740.809	0	2245903.551	1460348.342	2378462.933	2372492.722	2258002.516	2007594.29	1598705.227	1959478.925	fur-p4-2	fur-p4-2	fur-p4-2
fur-p3-4	1772834.537	2245903.551	0	1649767.431	1670751.941	3205709.394	2311856.971	1279921.05	1750042.075	1631837.369	fur-p8-3	fur-p7-4	fur-p8-3
fur-p4-2	1839294.652	1460348.342	1649767.431	0	1913870.714	2611807.24	1891536.182	1723015.427	1369341.129	1555282.202	fur-p9-2	fur-p10-3	fur-p9-2
Fur-p5-1	1853912.452	2378462.933	1670751.941	1913870.714	0	3563421.401	2344594.943	2268849.572	1912134.377	1811465.456	fur-p1-3	fur-p10-3	fur-p3-4
fur-p6-2	3607508.597	2372492.722	3205709.394	2611807.24	3563421.401	0	3260622.713	2677896.339	2837178.211	3106961.288	fur-p2-4	fur-p10-3	fur-p2-4
fur-p7-4	1789852.64	2258002.516	2311856.971	1891536.182	2344594.943	3260622.713	0	2467436.312	1934371.931	1685923.756	fur-p10-3	fur-p10-3	fur-p10-3
fur-p8-3	2143198.59	2007594.29	1279921.05	1723015.427	2268849.572	2677896.339	2467436.312	0	1759439.882	1830914.914	fur-p3-4	fur-p3-4	fur-p3-4
fur-p9-2	1675087.122	1598705.227	1750042.075	1369341.129	1912134.377	2837178.211	1934371.931	1759439.882	0	1500734.285	fur-p4-2	fur-p10-3	fur-p4-2
fur-p10-3	1475719.128	1959478.925	1631837.369	1555282.202	1811465.456	3106961.288	1685923.756	1830914.914	1500734.285	0	fur-p1-3	fur-p9-2	fur-p1-3