ZeroToHero

A Punch Recognition & Quality Assessment System.

By

LIAM O'SHEA

Supervised By

Dr. Sion Hannuna & Professor Majid Mirmehdi

Department of Computer Science
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of BACHELOR OF SCIENCE in the Faculty of Engineering.

MAY 2014

Abstract

Here goes the abstract

Dedication and acknowledgements

Here goes the dedication.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Table of Contents

List of Tables

Table **Page**

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

Computer aided coaching has revolutionised training approaches and been a key driver in improving sports performance at the highest level. Professional athletes can now use advanced augmented coaching to accurately and reliably measure a range of metrics that are then used as indicators of performance.[17] This gives the coaching teams unique insight which they use to tailor training programs to give the athlete a goal to focus on and improve.

Huge injection of research funding in preparation for the London Olympics such as The Elite Sport Performance Research in Training (ESPRIT) Programme, meant that UK Sport was allocated £2,000,000 while the Engineering and Physical Sciences Research Council were allocated £6,100,000.[36] This money was spent on enhancing elite performance training with technology [11] [35] which allowed British Olympians to take advantage of these training methods and return with a medal record making it the most successful team since 1908.[39] These hi-tech training methods were used by Team GB Boxing [2] and have started to become incorporated in the USA Boxing team.[16]

However despite the recent success, sports technology is still in its infancy, with the majority of devices being expensive, specialist and proprietary[14]. There is an array of consumer devices such as the Fitbit, Nike+ Fuelband, Garmin Vivofit and Jawbone that function as general activity trackers but there is currently nothing that offers specialist training or coaching advice. The purpose of ZeroToHero was to use an already widespread, low cost consumer device to bring specialist boxing coaching to everyone. The low-cost is especially crucial since most boxing clubs struggle with funding, rely on volunteers and traditionally have members that are the young and least wealthy in society. ZeroToHero explores the ability of the Kinect to act as a electronic boxing trainer, offering advice on punch and stance performance.

1.1 Goals

1. To classify the type of punch being thrown
2. To classify between a good and bad punch

1.1.1 Limitations & Scope

For the course of this project the 'orthodox' stance will be used, meaning only those who are right handed will be suitable. *Cannot deal with movement while punch is being thrown *Need to be a set distance from the Kinect

1.2 Scope of Project

The 'orthodox' stance will be used, meaning only those who are right handed will be able to use this implementation.

1.2.1 Other applications/usefulness?

*Any periodic motion?

CHAPTER 2

BACKGROUND & RESEARCH

This chapter describes and explains boxing concepts which are important to understanding the goal of the project. It gives an overview of the types of punches a boxer must execute along with common errors associated with those. It also reviews relevant literature in areas covered by this thesis and explores the current cutting edge possibilities. Finally I present an overview of the potential techniques that can be used for my project after analysing a variety of different techniques for their suitability based on the literature review and cutting edge methods.

2.1 Boxing Background

Boxing requires an incredible amount of co-ordination and timing as well as the ability to rapidly execute punches in a controlled and precise manner. Unlike professional boxing which is 12 x 3 minute rounds an amateur bout is 3 x 2 minute rounds which changes the dynamics of the contest. Amateur boxing relies on a points scoring system since there is often insufficient time for knockouts, requiring amateur boxers to rely on the mastery of technique and proper form. For example dropping a guard for a split second can open you up to an experienced boxer and could spell disaster.

As someone who has boxed for over 5 years and captain of the University of Bristol's Amateur Boxing Club (UOBABC) I understand the difficulties of developing good technique and how much time and experience is required from a coach to develop a new boxer. This one-on-one time is incredibly valuable but also expensive and for the large majority very hard to get. ZerotoHero aim is to be able to identify different types of punches and to offer feedback on the quality of the movement. This will bring some much needed expert advice to a beginner who can practice in the comfort of their own home.

I am using my own experience and that of local professionals, coaches and local legend Denis Stinchcombe MBE the centre director of Riverside Youth Project and Registrar for the Western Counties for the Amateur Boxing Association.

2.1.1 Motivation

This research is borne out of a desire to improve access and cost to boxing coaching which are problems I have encountered first hand through the University Boxing Club. In a wider context it could be used in developing countries where physical access to coaches with the required expertise may be difficult as well as local clubs in the UK. Every year UOBABC takes in new members that are total beginners. We spend an enormous amount of time and effort helping them learn the basics and encourage people to practice at home. The problem from a boxers perspective is that it is incredibly hard to spot your own faults, especially without experience. If it was possible to practice at home with the benefits of coaching it would bring massive improvements to a trainees ability. The current most effective way to train as an individual is to stand in front of a mirror and observe yourself while shadow boxing. It could also be used as a way to introduce younger children to the sport since the Kinect is incredibly popular in that demographic and so is a good choice from an inclusion perspective.

2.2 Boxing Technique

For the scope of this project I am going to focus on the most common orthodox stance. There are tens of slight variations on each punch but I am going to focus on the core foundations and important principles from which these can be built.

2.2.1 Stance

The most fundamental building block of boxing is the stance, that is how you hold and position your body as well as the placement & orientation of your feet. A good stance is crucial since it allows the boxer to be well balanced and light on their feet, allowing fast movement in any direction as well as the ability to quickly duck, weave, slip and bob and lay back to avoid punches. It is also crucial for offence since the power from punches come from the transfer of weight from one leg to another which requires a very specific twisting hip movement. Often beginners forget this crucial step and so I'm hoping to use this unique trait to help me judge quality later on. A successful stance should have the following characteristics:

- Left foot forward, right foot back with a distance slightly wider than shoulder width with a 45 degree angle twist.
- Right heel of the ground at all times with weight distribution mostly on your back leg.

- Chin tucked down.
- Right hand on the right hand side of your chin, left hand should be a few inches in front of the left side of the face.
- Elbows tucked in to protect the torso section. . . .

2.2.2 Punches

Jab

The elbow should stay tucked in while the left fist extends with palms facing inwards before twisting your wrist at the last moment. The natural thing to do is extend the punch with palms facing down, unfortunately this immediately makes the elbow stick out which allows the opponent to easily see you are about to throw a punch (telegraphing) while opening up your body for a counter attack. The punch should also finish so your arm is fully extended which helps to extend your reach and protect your chin before speedily returning it to the guard position.
Target Characteristic: Elbow movement

Cross

The cross is designed as your heavy straight punch and as such is slower but more powerful. To get a snappy and powerful punch it is important to transfer your weight rapidly from your back leg to your front leg, twisting your hips.
Target Characteristic: Twisting of the hip
Target Characteristic: Distribution of weight to the front foot

Hooks

Your elbow should be raised to shoulder height and your fist and shoulder should be at 90 degrees to each other. A transfer of weight between the front foot and back with the twisting of the hip is essential.
Target Characteristic: Elbow being raised to parallel
Target Characteristic: Hip twist resulting in weight transfer from front to back

Uppercuts

This required the fighter to crouch down into the squat position and throw a punch vertically upwards, with the aim of striking the opponent's chin. Target Characteristic: Sufficient crouching before releasing the punch
Target Characteristic: Directly vertical punch, keeping guard close at all times.

2.3 Kinect

This section provides some background information about Microsoft Kinect that is important for understanding the features and limitations of Kinect Analysis. According to Microsoft, the Kinect has worldwide sales of approximately 28 million units and contains an RGB camera, an infrared (IR) emitter and an IR depth sensor as well as a multi-array microphone. The interaction space of the Kinect is limited by the field of view of the Kinect cameras. The Kinect has a 43°vertical by 57°horizontal field of view. The Kinect sensor can be tilted using a built-in tilt motor. Tilting the Kinect increases the interaction space by +27 and −27 degrees. The Kinect sensor provides sensor data in form of data streams. It can capture audio, color and depth data. In addition, it can process the depth data to generate skeleton data. Therefore, the Kinect offers four different data streams that can be accessed: audio stream, colour stream, depth stream and skeleton stream. The streams can deliver at most 30 frames per second (FPS) using a resolution of 640×480 which drops to 12 FPS with a resolution of 1280×960.

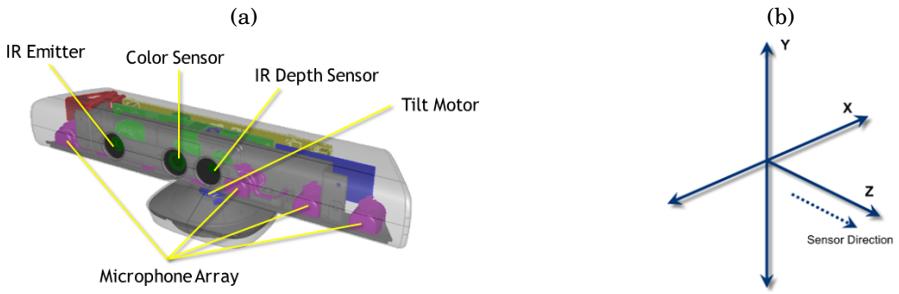


FIGURE 2.1. (a) Kinect (b) Kinect Dimensions

Kinect	Array Specifications
Viewing angle	43° vertical by 57° horizontal field of view
Vertical tilt range	±27°
Frame rate (depth and color stream)	30 frames per second (FPS)
Audio format	16-kHz, 24-bit mono pulse code modulation (PCM)
Audio input characteristics	A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing including acoustic echo cancellation and noise suppression
Accelerometer characteristics	A 2G/4G/8G accelerometer configured for the 2G range, with a 1° accuracy upper limit.

FIGURE 2.2. Kinect Specifications

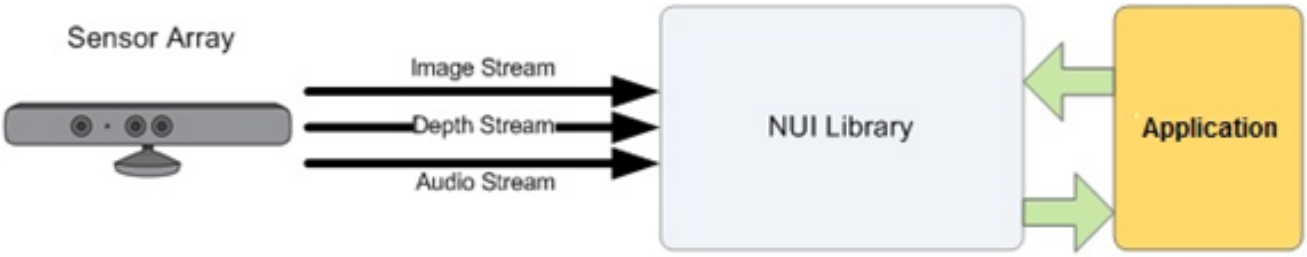


FIGURE 2.3. Kinect Specifications

Depth & Infrared Stream

The depth sensor generates invisible IR light to determine an object's depth from the sensor. The primary use for the IR stream is to improve external camera calibration using a test pattern observed from both the RGB and IR camera to more accurately determine how to map coordinates from one camera space to another. [23] The NUI API uses the depth stream to detect the presence of humans in front of the sensor.[6] Skeletal tracking is optimized to recognize users facing the Kinect, so sideways poses provide some challenges because parts of the body are not visible to the sensor.

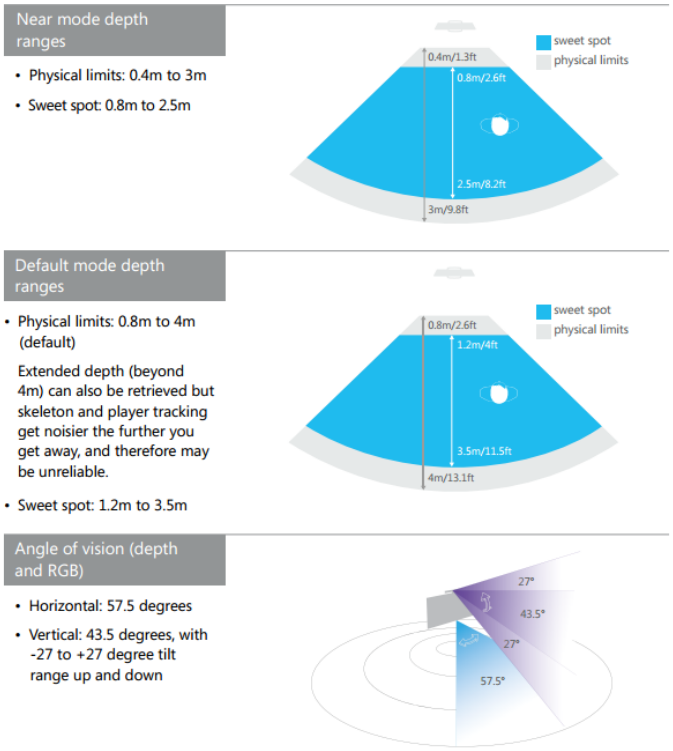


FIGURE 2.4. Kinect Depth Specifications.

2.3.1 Skeleton & Joint Tracking

The Kinects default tracking mode can track up to 20 joints per skeleton providing the subject is standing relatively face on and are fully visible to the sensor. Although the Kinect is capable of different modes (e.g. sitting) this is not relevant in the context of my project. Each skeleton frame contains the position of each joint as well as information about the tracking quality. Joints can have one of three different tracking states, Not Tracked (0), Inferred (1) and Tracked (2), this flag is useful as an indicator of the quality of the measurements you are receiving for a particular joint. When possible, tracked joints are used to help calculate the position of those joints that cannot be directly tracked hence the ability to infer joints. The Kinects default tracking mode is designed to track people who are standing and fully visible to the sensor. The default range requires skeletons to be at least 80 centimetres away from the device to be tracked properly.

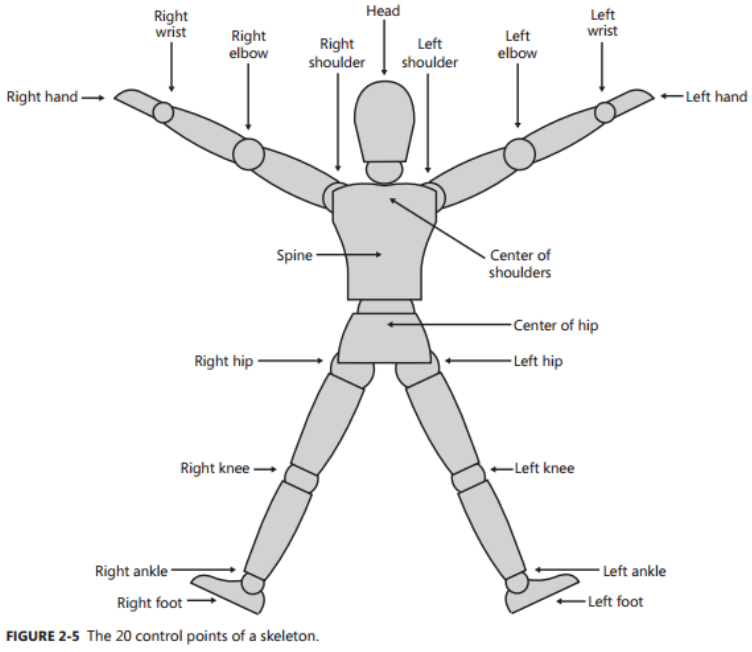


FIGURE 2.5. Kinect Joint Tracking Skeleton.

2.4 Literature Review

The Kinect is currently a rich area of research in its own right and it straddles the fields of Image Processing and Computer Vision as well as Human computer Interaction which are exciting and popular areas of research. Recognising punches is an example of activity recognition in which there has been a fairly large body of work using the Kinect since it was released to PC in 2012 by Microsoft.

With the development of such a low cost device researchers have shifted their focus onto action analysis and gesture recognition based on depth maps. I wanted to see what was capable with the Kinect at the cutting edge of research so I consulted the most recent papers from the conference on Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV) & European Conference on Computer Vision (ECCV) as well as an array of other resources to find the most relevant and useful research.

Examples of current work are a customisable hand gesture recognition interface [5], physical therapy instruction [9][8], Karate instruction [4] and quality assessment of the reach and grasp of stroke survivors [37].

Oya caliktutan developed a graph-based method for aligning two dynamic skeleton sequences with the aim of action recognition and 'objective quantification of goodness' of said action. [7] Depth maps are used similarly to the real-time skeleton tracking algorithm developed by Shotton. [33]

In 2013 a controlled study of 16 people who were blind or who had low vision was ran to test the usefulness of Eyes-Free Yoga: An Exergame Using Depth Cameras for Blind & Low Vision Exercise. The purpose of this was to teach the participants yoga poses using audio feedback and a positive response from participants.[29] The study took the Kinect joint positions and calculated joint angles while using heuristics for each pose. However the study only measured success using 6 unique static poses that were required to be held for an extended period of time. In contrast boxing requires fast repeated movements with many of the movements sharing similarities which make them harder to differentiate. Therefore I did not find this useful for evaluation the feasibility of my project.

The Kinect has also been used to evaluate salsa dancers performance with comparison to a professional dancer in real-time[1][32]. A score was achieved by adding three different metrics, one from the correlation coefficient of quaternions, another using joint velocities and finally a "3D flow Error," calculated from frame vectors. Quaternoin correlation was then used to estimate the time-shift between two dancing sequences.
The experimental results were encouraging with most of the scores consistent with real-life rankings with the exception of a few poor results due to bad skeleton calibration and tracking. This work draws strong parallels in what I am trying to achieve and demonstrated that the Kinect was a viable option for my work.

Work on disc throwing performance[40] also showed some promise with limited success with the lower ability groups improving their movement. This approach simply used joint angles to measure 5 different phrases of the throw and compare that to joint angle rules. I found this to be overly simplistic in comparison to the project I am trying to achieve although it did indicate that joint angles may be a useful technique for judging movements.

Xia took each joint position and represented this using a histogram. [20] The joint positions were converted from Cartesian co-ordinates to spherical co-ordinates and splitting the sphere

into several grids. A sequence of poses was recorded and represented as histograms before using Hidden Markov Models for classification. Some of the concepts used in this work were adapted from Li [38] in particular the 'bag of 3D points' used to characterise poses.

Previous implementations have used joint weighting which have been shown to increase recognition performance with weighted dynamic warping[31] and decision forests [19]. Join weighting is also used with a newly proposed skeletal representation 'Sequence of most informative joints (SMIJ)' where at each time instance the most informative skeletal joints are selected depending on the action being taken [26]. In this work each sequence was divided into small temporal sequence (frames), taking into account joints with maximum variance and represented these 'most informative joints' with a histogram for action recognition.

Raptis proposed a different skeleton representation which was designed for 'recognition robustness under noisy data'[28] alongside a cascaded correlation-based classifier and a distance metric produced by dynamic time warping that will match the performed gesture to the closest matching gesture.[18] PCA is applied on only the torso joint and used to estimate the orientation of the skeleton which allows specific relevant features to be extracted.

The most recent research in real-time action recognition methods have been built on random decision forests.[19] [25] Miranda's 2012 paper[24] and Raptis's paper [28] both use SVM's to identify key poses before passing them to the forest to classify. A comparative summary of 'model-based method for body pose estimation and tracking'[21] show SVM to be a popular and effective technique used in 2013.

auto seg

Using RBB video I have encountered very few methods for automatic action quality assessment although a few approaches have been proposed based on skeleton tracking. For example Bianco and Tisato recognize Karate moves based on skeletal joints.[4]

For each strike a triplet of joints are manually selected and can then be represented by the angles of the joints. K-means clustering is used to obtain a set of key poses and Dynamic Time Warping attempts to align sequences of poses. The distance measure from DTW is then used as a performance evaluation. However the quality of this paper did not seem to me to meet the same standards as many of the others reviewed and the poses used were very static and very different to each other. Given this I would expect a good classification performance since there should not be too many common features. Since boxing has a much more 'closed' guard and the punches share many similarities I am not convinced that k-means clustering will produce good results.

2.5 Existing Products

UFC Personal Trainer The closest commercial product is 'UFC Personal Trainer' a very broad commercial Xbox game aimed at introducing people to UFC. After evaluating this product

it became clear to me that it's main focus is exercise regimes rather than technical fighting. Therefore it does not offer the preciseness or technical fighting focus that I require. In this game you technically wrong punches will still register and several movements are unrealistic.

Kinect sports boxing game

This game has very poor punch discrimination and does not require any sort of real boxing ability. The goal here is usually to punch towards the Kinect controller as quickly as possible. It fails to recognise properly thrown hooks and uppercuts and translate those into the game.

Fighers Uncaged

This game has generally poor reviews from reviewers, with most complaints involving the reliability of the Kinect to accurately measure fighting moves. [15] "When the fights actually start, pulling off moves becomes a series of desperate flails, trying to get the game to recognize your actions."[13] and "The game fails to register most of the movements", "The idea was great for Kinect, but something went horribly wrong along the way. Kinect is supposed to register your every kick and punch, but it only catches about the half of them.", "the Kinect control is lazily implemented."[22]

Literature & Product Conclusion

From my research and product comparisons I concluded that the Kinect was a viable option for my project and worth pursing. Crucially however there has been no research specifically into the area of boxing with the Kinect which brings it's own unique challenges. Boxers are trained to be fast, well guarded and to give very little away in their movements, especially punches. Therefore many of the punches and poses are very similar since the goal is to be naturally evasive. This will make segmenting punches and giving useful quality metrics challenging in it's own way unlike say a discus throw. A boxers stance is often quite 'closed' so I anticipate challenges tracking obscured joints and the overall accuracy of inferred joint positions. However I could see from the above studies that hip movement could be effectively tracked which was my main concern when evaluating the Kinect since all boxing moves rely on the hip rotation.

It is clear that this will not be an easy project. Many years of Microsoft research have gone into the current Kinect but surprisingly games for their flagship console fail to provide tracking accuracy that satisfies consumers. Furthermore no commercial products exist that incorporates proper boxing technique into a game, suggesting there may be limitations with the hardware that prevent this or that it it simple difficult to do. However the research in this area has shown sufficient promise for me to combine my boxing and computer science knowledge to work on this problem.

2.6 Dimensionality

Curse of Dimensionality

The curse of dimensionality, first discussed by Richard E Bellman in his book Dynamic Programming[27] is the term for a set of problems that occur when using high dimensionality data. As dimensionality increases as does the search space, resulting in the available data become sparse. In order to obtain accurate, reliable and statistically sound results the total amount of data required can grow exponentially in relation to the dimensionality. Likewise the organisation and searching of non-reduced data becomes difficult, with space and search time dependent on data volume. Searching and organising data also relies on the ability to group instances into groups that share similar characteristics, unfortunately if the data appears to be dissimilar due to sparseness it can prevent grouping strategies.

Algorithms that can successfully deal with high dimensionality data typically will have high time complexity and **will not always** produce more accurate results than algorithms that work on the lower dimensionality data. Therefore it is sensible to look at some dimensionality reduction techniques which **might**produce better results.

Furthermore since the long-term goal is to give feedback with live data, speed is of the essence. Therefore dimensionality reduction is an important component required to decrease the processing time on any input.

2.6.1 Dimensionality Reduction

Dimensionality Reduction is the process of reducing the number of variables under consideration for any given problem. For example each frame from the Kinect is represented by 80 data points, 60 of which are the x,y,z co-ordinates of the 20 skeleton joints. Considering the Kinect is capable of 30 FPS that is 1800 data points per second of movement. With long sequences of recordings this could become a huge amount of data to process, most of which could be represented in a reduced dimensionality.

2.6.2 Principal Component Analysis (PCA)

Principal Component Analysis is a statistical procedure that transforms a set of observations of potentially correlated variables into a set of linearly uncorrelated variables called principal components. The number of principal components should always be less than or equal to the number of original values with the first principal component having the largest possible variance. Each following component will attempt to represent as much variance in the data as possible. In my case I will be looking to reduce my 60 data points per frame into a low dimensionality set that will help me to uniquely identify punches.

Eigenvectors & Eigenvalues

The eigenvector of a matrix M is a vector V such that $M \times V$ yields a constant multiple of V, the multiplier of which is denoted by λ.

$$Av = \lambda v$$

λ is known as the eigenvalue of M which corresponds to v. Any multiple of an eigenvector will have the same eigenvalue as as original eigenvector.

The principal components produced by PCA correspond to the principal eigenvectors of the covariance matrix which are calculated by eigenvalue decomposition. The principal eigenvectors are those with the highest eigenvalues and so are easily deduced.

Covariance Matrix & Optimisations

If the data is represented in a column vector each with a finite variance then the covariance matrix $\sum$ is the matrix whose (i,j) entry is the covariance

$$\sum_{\mathbf{ij}} = COV(X_i, X_j) = \mathbf{E}[(X_i - \mu_i)(X_j - \mu_j)]$$

$$\mathbf{X} = \begin{bmatrix} X_i \\ . \\ . \\ X_n \end{bmatrix}$$

Where:

$$\mu_i = E(X_i)$$

and

$$\mu_j = E(X_j)$$

The is equivalent to

$$\sum = E[(X - E[X])(X - E[X])^T]$$

with $\frac{1}{n-1}$ added to make the estimate unbiased.

The number of data points in each frame is 60 and the Kinect is capable of 30 FPS which over 120 seconds produces 216,000 data points to represent that sequence. A matrix of this size is costly and slow to find eigenvectors for. Let us say that 60n is the number of data points for frame and m is the number of frames. Take X as size (d x 60n) and X^T as (60n x d) meaning $X^T X$ is of size (60n x 60n), while XX^T is of size (d x d). It is however known that the eigenvalues for $X^T X$ are the same as for XX^T which is useful since this is of a much smaller size and hence less costly.[34] The eigenvectors (v_i) of $X^T X$ are related to the eigenvectors (U_i) of XX^T by the equation

$$V_i = X^T U_i$$

with V_i needing to be normalised afterwards. This method can reduce the eigenvalues and eigenvectors produced from a maximum possible N^2 for $X^T X$ to M values for $X^T X$.

The M eigenvalues of XX^T will be identical to the first M largest eigenvalues from $X^T X$. This is an important performance step since the goal of my system is to run in real time and this will reduce the time taken.

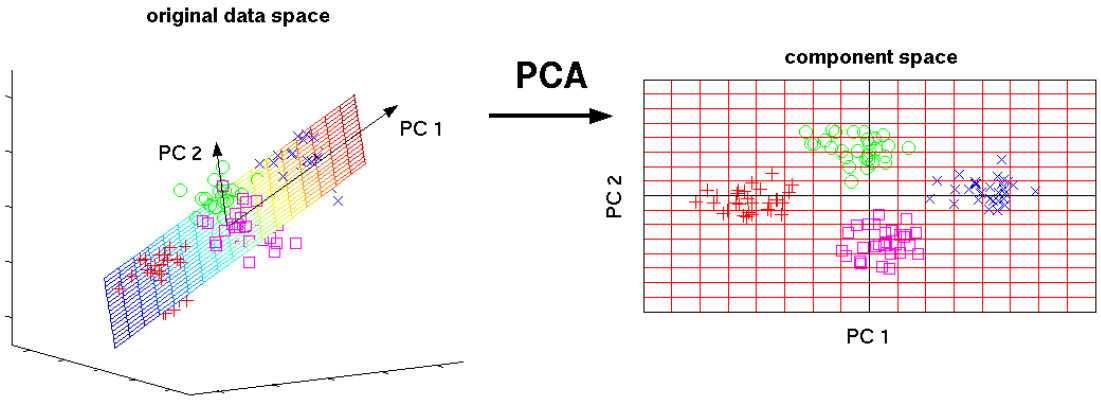


FIGURE 2.6. PCA example showing the 1st and 2nd largest eigenvalues and the component space

2.7 Manifold Learning

Manifold learning is a Non-Linear Dimensionality Reduction process that transforms high dimensionality data that typically requires multiple dimension to represent it which is difficult to interpret. To simplify the data it is possible to assume that the date lies on an embedded non-linear manifold within the high-dimensionality space. Assuming the manifold has low enough dimensions it can then be visualised in this lower-dimensionality space.

2.7.1 Diffusion Maps

Diffusion maps are a non-linear and relatively new technique developed in 2006 by Ronald R. Coifman and Stephane Lafon.[10] The aim of a diffusion map is to provide a framework for finding meaningful geometric descriptions of data sets. Diffusion maps are capable of turning high dimensionality data into low dimensional structure. Unlike other dimensionality reduction techniques like PCA, diffusion maps attempt to discover the underlying manifold, a lower dimensional constrained surface containing the data. Diffusion maps are based on defining a Markov chain on the graph of the data. By performing this for a set number of time steps a measure for the proximity of data points is obtained, which is used to define a diffusion distance. Pairwise diffusion distance are retained as well as possible in the dimensionally reduced data.

2.7.2 Locally Linear Embedding (LLE)

In LLE a data manifold is constructed by finding a set of the nearest neighbours for each point[30]. Together they are used to compute a set of weights that describes each point as a linear combination of its neighbours. Finally, it uses an eigenvector-based technique to find the low-dimensional embedding of points, such that each point is still described with the same

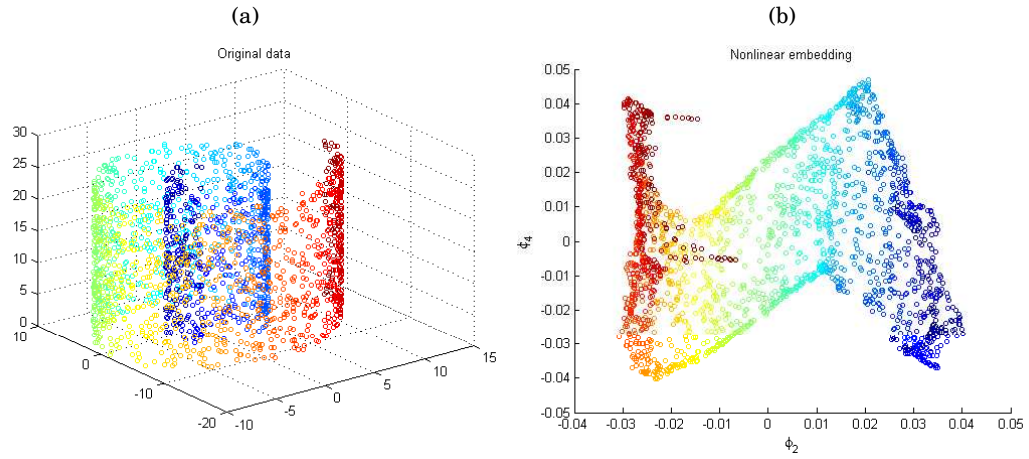


FIGURE 2.7. (a) Original example data (b) Non-linear embedding

linear combination of its neighbours. Furthermore, the preservation of local properties allows for successful embedding of non convex manifolds. **LLE tends to handle non-uniform sample densities poorly because there is no fixed unit to prevent the weights from drifting as various regions differ in sample densities.**

2.7.3 Laplacian Eigenmaps

Laplacian Eigenmaps find a low-dimensional data representation by preserving local properties of the manifold.[3] Similar to LLE, a graph is built from neighbourhood information, with the distance between a point and it's K nearest neighbour is minimised. A weighting system is used such that in the low-dimensionality space the distance from a point to it's nearest neighbour is more significant to the cost function than other nearby points. **put simply the closer a neighbour is to the selected data point the heavier its weighting.** The goal overall is to minimise the cost function based on the graph information to ensure that points close together in the high dimensionality data remain so after the reduction, preserving local distances.

2.7.4 Local Tangent Space Alignment (LTSA)

LTSA is based on the intuition that when a manifold is correctly unfolded, all of the tangent hyperplanes to the manifold will become aligned.[41] In other words there will exist a linear mapping from high dimensionality data to a local tangent space which will have a linear mapping to a low-dimensionality data point. It begins by computing the k-nearest neighbours of every

point. It computes the tangent space at every point by computing the d-first principal components in each local neighbourhood. It then optimizes to find an embedding that aligns the tangent spaces.

2.7.5 Curvilinear Component Analysis (CCA)

CCA is an learning algorithm that starts with larger distances before iterating to smaller ones.[12] It looks to output a configuration of points that preserves the original distances as much as possible while focusing on the smaller distances.

The large distance information will be overwritten by the smaller distance information unless a conflict occurs. The stress function of CCA is related to the sum of Bregman divergences which aim to generalise squared euclidean distances so they all share the same properties. If compromises between the larger and smaller distance information but be made the stress function determines this.

2.8 Segmentation Methods

Automatic segmentation is a crucial step in this project since on a larger scale much greater data sets will need to be used and collated from multiple sources. If these can be processed and automatically segmented this will remove any manual work required and make this a truly useful system. I will look at several methods that might aid me in finding meaningful characteristics to separate each punch.

2.8.1 Dynamic Time Warping (DTW)

Dynamic time warping is a time series comparison method that takes two temporal sequences, often which vary in time or speed and tries to calculate an optimal match. The two sequences are aligned and a similarity score is produced. This technique can be used on any data that can be represented in a linear fashion and has been successfully applied to fields such as signature recognition, voice recognition, partial shape matching and gait analysis. For example in gait analysis, similarities could be detected in the walking pattern of two people even though their speed and acceleration may be difference. If a punch sequence is similar enough to others of the same type and different enough from other types of punch this could prove a useful method of recognising the start and end of a punch as well as the type of punch.

2.8.2 Fourier Transformation (FT)

The Fourier Transform is a mathematical transformation that can decompose a time series from the **spatial domain (normal image space) | should this be time domain?** and transform it into the frequency domain. Any wave-like functions can be represented as a combination of simple

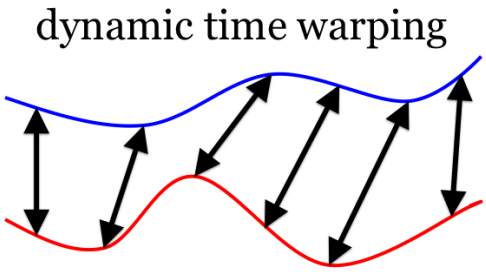


FIGURE 2.8. Dynamic time Warping

sine waves by breaking down the original series into the sum of a set of oscillation functions, sines and cosines. The series is split into a magnitude spectrum which will comprise of complex exponentials and a phase spectrum which will encode angles in radians. The Fourier series of a function $f(x)$ is given by:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n cos(nx) + \sum_{n=1}^{\infty} b_n sin(nx)$$

Where:

$$a_0 = \frac{1}{\pi}\int_{-\infty}^{\infty} f(x)dx$$

$$a_n = \frac{1}{\pi}\int_{-\infty}^{\infty} f(x)cos(nx)dx$$

$$a_n = \frac{1}{\pi}\int_{-\infty}^{\infty} f(x)sin(nx)dx$$

Here the cosine coefficient is a_n and the sine coefficient is b_n.

A property worth noting is that a shift in the time domain corresponds to a linear change in phase but does not alter the magnitude spectrum. Since my punches sequences over time are a continuous sinusoidal-like function I can use the discrete-time Fourier transform (DFTD) to obtain the Fourier coefficients which I hope will be unique enough to tell me something useful about the punch sequence.

2.9 Classification Methods

2.9.1 Support Vector Machines

A Support Vector Machine is a kernel-based method that is effective for highly dimensional datasets. The SVM uses a kernel to calculate the scalar product of two feature vectors in a high dimensional feature space. The decision function uses the hyper-planes, defined by the Support

vectors, to classify the data. Only significant samples are taken for use as support vectors so that a high variance will make less of a different to the accuracy of the model.

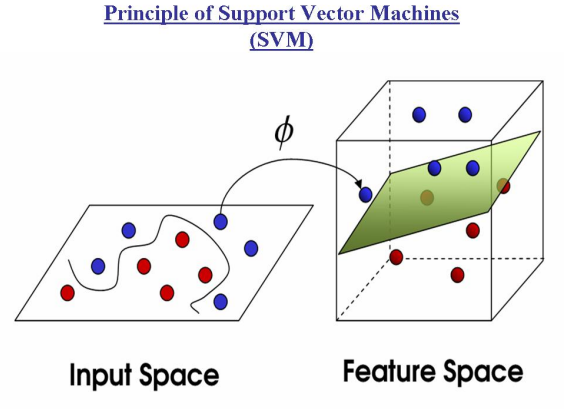


FIGURE 2.9. Kinect Device.

2.9.2 Neural Networks

Neural networks are models based on the parallel processing of information similar to that of the brain. A NN can be configured for different applications such as clustering, pattern recognition, Dynamic Time series and curve fitting. They are able to derive meaning from complex data that human beings would be unable to notice and that other techniques will fail on. Crucially the networks are capable of approximating non-linear functions **unlike DTW. Neural networks learn by examples in the form of training data and are adaptive based on a system of weightings calculated by % error. Other characteristics are their ability to self organise and the ability to work in real-time if sufficient parallelism is supported.**

Below is a model of a simple Neuron with many inputs and a single output. This is based on the biological models of a single neuron where a certain injection of current will decide if the neuron should fire. The 'Integrate & Fire' model is as follows:

$$C_m \frac{dv}{dt} = -G_l(V - E_l) + I_e$$

With time constraint:

$$\tau_m = \frac{C_m}{G_l}$$

And membrane resistance:

$$R_m = \frac{1}{G_l}$$

IF:

$$V > V_t h'$$

THEN: Generate a spike.

$$V = V_reset$$

In this scenario the neuron has two modes of operation, testing and training. In the training mode the neuron can be trained to fire depending on a particular input pattern. In the testing mode when a recognised input pattern is detected the associated output from the training becomes the output. If the pattern is not recognised in the list of taught input patterns then it tries to identify that which is closest to it (which it does recognise) and produce an output matching that.

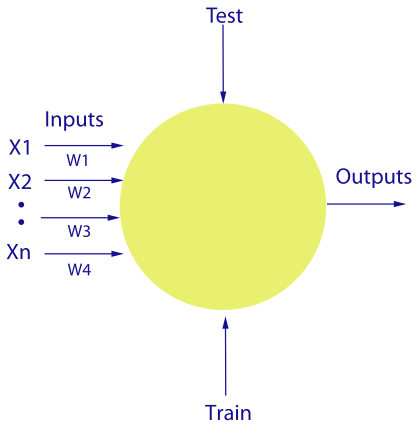


FIGURE 2.10. Kinect Device.

Back Propagation (BP)

Neural networks (NN) work by connecting a series of single neurons into a learning network. Back Propagation is a learning algorithm whereby the network behaves similarly to above but with the added complexity that each input is given a weighting. This weighting is calculated iteratively by working out the error for each neuron like so: $Error_a = Ouput_a(1-Output_a)(Target_a-Output_a)$ which put simply is the difference between the correct output and the actual output for neuron A.

Since BP networks learn by example they must be given labelled training data which is used to correctly calculate the weights given to each input to give the required output. Since BPNN are ideal for pattern recognition and mapping tasks it could be suitable for use on punch sequences.

2.9.3 Decision Trees & Random Forest

A predictive tree like model which maps observations about an object to reach a conclusion about its target value. In this model the 'leaves' represent class labels while the branches themselves represent conjunctions of features that will lead to the class labels. Put simply each condition in an internal node while each outcome is an external node. Information gain is the difference between the initial entropy and the new entropy after following a branch along the decision tree. Since the goal of any machine learning technique is to achieve a low value of entropy to make

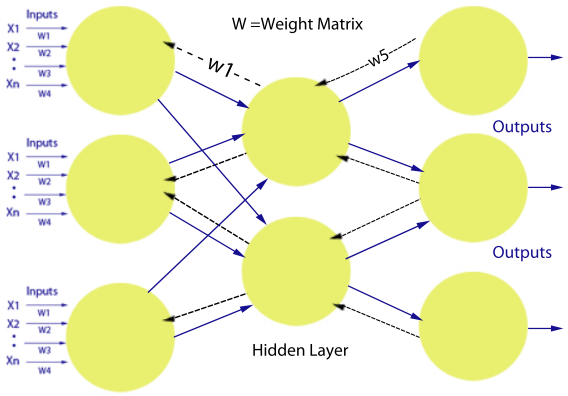


FIGURE 2.11. Kinect Device.

accurate predictions, the decision tree is constructed such that each branch has the maximum possible information gain. The data is split by each feature that has the maximum information gain recursively for each branch.

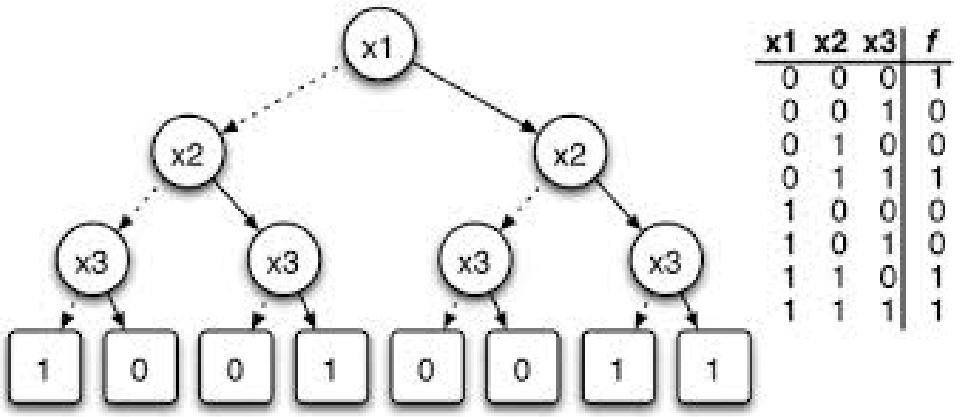
Bagging is a method of assigning a measure of accuracy to sample estimates. We sample from an approximating distribution and try to approximately calculate the properties of the estimator based on this. We measure a statistic from a sample of the population and then use this to say something about the whole population. For example, we might say that for our set of data a subset is used to determine a class. If any sample in the entire population follows the same tree rules then it will be labelled as in that class. We have used bootstrapped samples in our classifiers as they have been constructed using random sampling with replacement. This method is designed to improve the accuracy of machine learning algorithms, helping to reduce variance and hence over fitting.

Random forest is an averaging algorithm which produces a diverse set of classifiers by introducing randomness in the classifier construction. Each tree is built from a bootstrapped sample from the training set. During construction the node splits are chosen based on the best split among a random subset of features, not the best split among all the features. Due to the randomness of this algorithm we expect it to have a slight bias compared to other decision tree models, however due to the averaging we expect a lower variance.

These Ensemble methods combine the predictions of multiple trees to come to a consensus.

2.10 Punch Quality Algorithm

I will be measuring quality in reference to a 'ground truth' produced by a local professional. My aim is to create rules that are capable of detecting basic mistakes and provide feedback. I will start by targeting the characteristics as mentioned in the earlier background chapter, for example throwing a jab with the elbow sticking out instead of tucked is a classic beginners mistake.



x1	x2	x3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

FIGURE 2.12. Kinect Device.

CHAPTER 3

IMPLEMENTATION

Each individual implementation stage will be discussed, collection of data, dimensionality reduction, punch segmentation, punch classification and quality assessment.

My first step was to organise the data in such a way that it could be usefully visualised so that I comprehend the nature of the data that needed to be processed. I began by recording a sequence of 5 jabs, being careful to evenly time the punches before manually isolating the left wrist joint over time. As my jab was thrown 5 times with approximately equal spacing I was hoping that I would be able to produce a sinusoidal-like sequence with five full 'cycles' demonstrating the movement of the left hand over time.

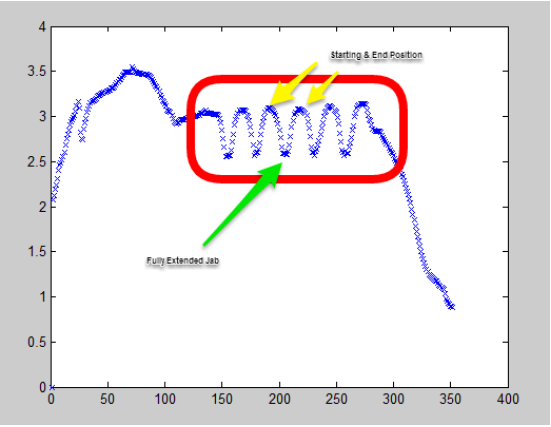


FIGURE 3.1. This figure shows the left wrist joints movement in the Z direction over time. The x-axis represents the kinect frames, they y-axis the distance of the fist from the kinect.

SO MANY REASONS WHY JUST ONE HAND IS SHIT POSE, whole body, not robust etc...

I needed to find a sensible way to visualise the entire skeleton over time in a meaningful way. I naively began by plotting the entire data set which produced a fairly jagged signal with two troughs followed by a massive peak before levelling of again. At this point I realised attempting to plot the skeleton data in this way would not be meaningful but as a test I differentiated the raw data to produce velocity as opposed to distance, I was hoping that this might show something more meaningful or smooth out the data. Realising that for now I needed to look at individual joints I took the central hip joint and plotted it's movement in the Z direction over time as well as plotting the differential of that to give me the velocity. As you can see from Figure 3.1, the velocity data was much more useful and produced a reasonably smooth periodic sequence. Since the initial recording was 10 jabs one after the other I expected a periodic, cyclical signal to be produced.

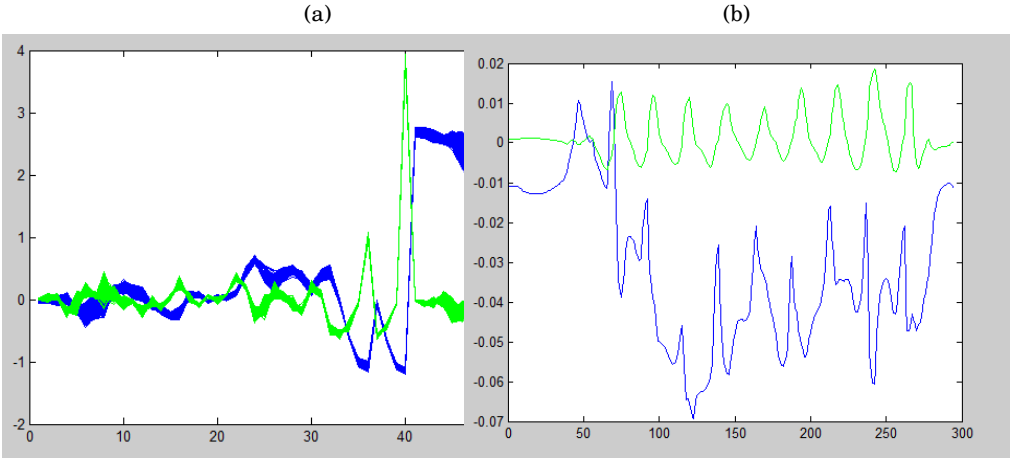


FIGURE 3.2. (a) Kinect Skeleton vs Time. (b) Hip Joint vs Time, Blue = Raw data, Green = Differentiated data.

Since PCA was a familiar method I implemented it in Matlab with 3 principal components and plotted the first project coefficient for the first principal component with both the distance and velocity data. From this we can see that the distance data was slightly jagged with a local maxima which might obstruct attempts to automatically segment punches. The velocity data produced a smoother series but introduced unwanted noise towards the beginning and end of a sequence.

Next I plotted all three projection coefficients for the three principal components for both distance and velocity. We can see for the first projection coefficient both distance and velocity are similar with distance being slightly more jagged at the peaks of the punching period and velocity being more jagged at the start and end. For the second they are both fairly similar but with distance having less peaks and troughs and with only one dramatic spike at 250 as opposed to two at 50 and 275. For the third the depth comes out ahead with a smoother signal which

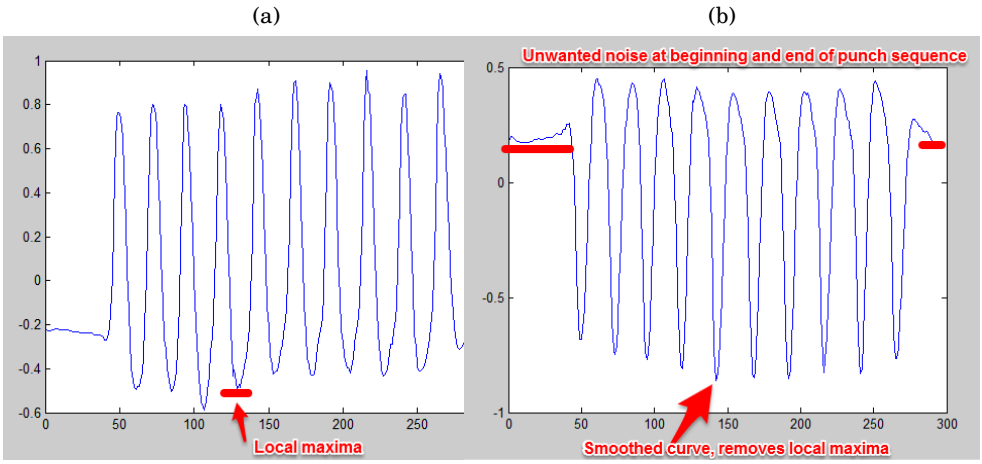


FIGURE 3.3. (a) Kinect Skeleton vs Time. (b) Hip Joint vs Time, Blue = Raw data, Green = Differentiated data.

again has less 'spikes' (which will make it hard to segment.) Although by eye it looks like the depth data under PCA produces a more useful periodic type wave neither are convincing and we should do some comparisons.

Dimensionality Reduction Comparison

Each dimensionality reduction method was evaluated on it's ability to produce useful, smooth and sinusoidal output to aid automatic segmentation. Automatic segmentation is a crucial step in the project since on a larger scale much greater data sets will need to be used and collated from multiple sources. If these can be processed and automatically segmented this will remove any manual work required and make this a truly useful system. I used the Matlab Dimensionality Toolbox and the 'intrinsic_dim' function that performs an estimation of the intrinsic dimensionality of a dataset X given a specific method. This was calculated six different methods, maximum-likelihood estimation (MLE), correlation dimension estimation (CorrDim), nearest neighbourhood, packing numbers and Geodesic minimum spanning tree (GMST) and eigenvalue estimation. These results were then combined and averaged to produce a dimension **d** to be used as an argument for LLE, LLE with d, Laplacian, LTSA, CCA and PCA (see background). The lower dimensionality data was then plotted and analysed.

We can see from the first principal component plotted from multiple dimensionality reduction method that PCA does in fact give the most useful results, followed closely by CCA. If we then look at the second **principal components** we can see that PCA performs second to LTSA but since LTSA performs poorly for the first component it makes PCA the obvious choice.

Despite PCA being an older, simple technique than more modern techniques such as diffusion maps it actually produced the most uniform, cyclical signal for the first component making it an obvious choice.

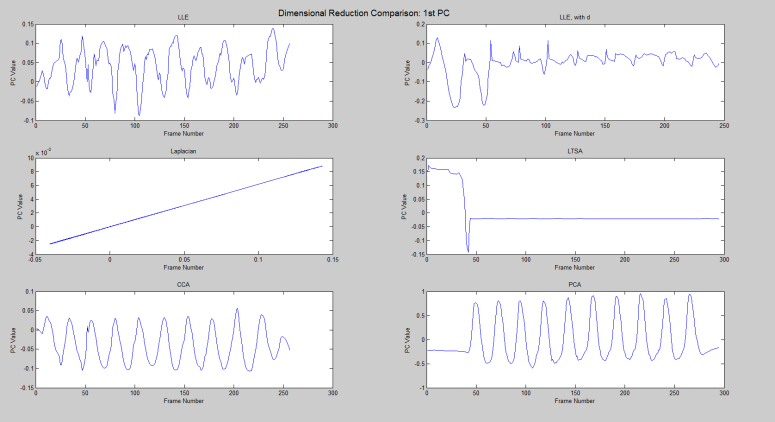


FIGURE 3.5. 1st PC comparison

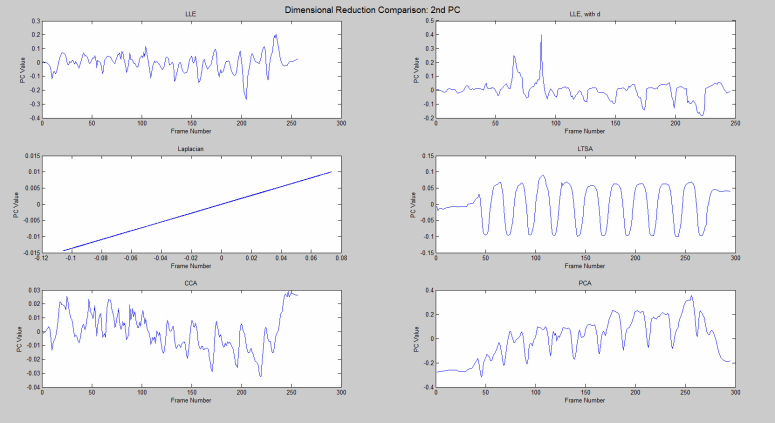
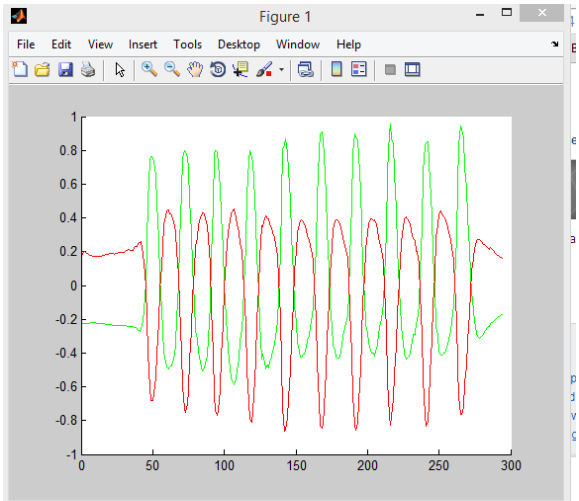
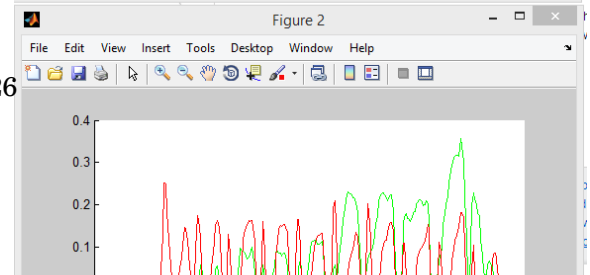


FIGURE 3.6. 2nd PC comparison

Now lets see the first Principal Component (from PCA) for each punch. TALK about normalisation of data. Talk about getting better data.





3.0.1 Punch Segmentation Algorithm

After failing to find a way to automatically segment punches using existing methods It became apparent that a custom algorithm will be needed for this task, which can be broken down into several steps.

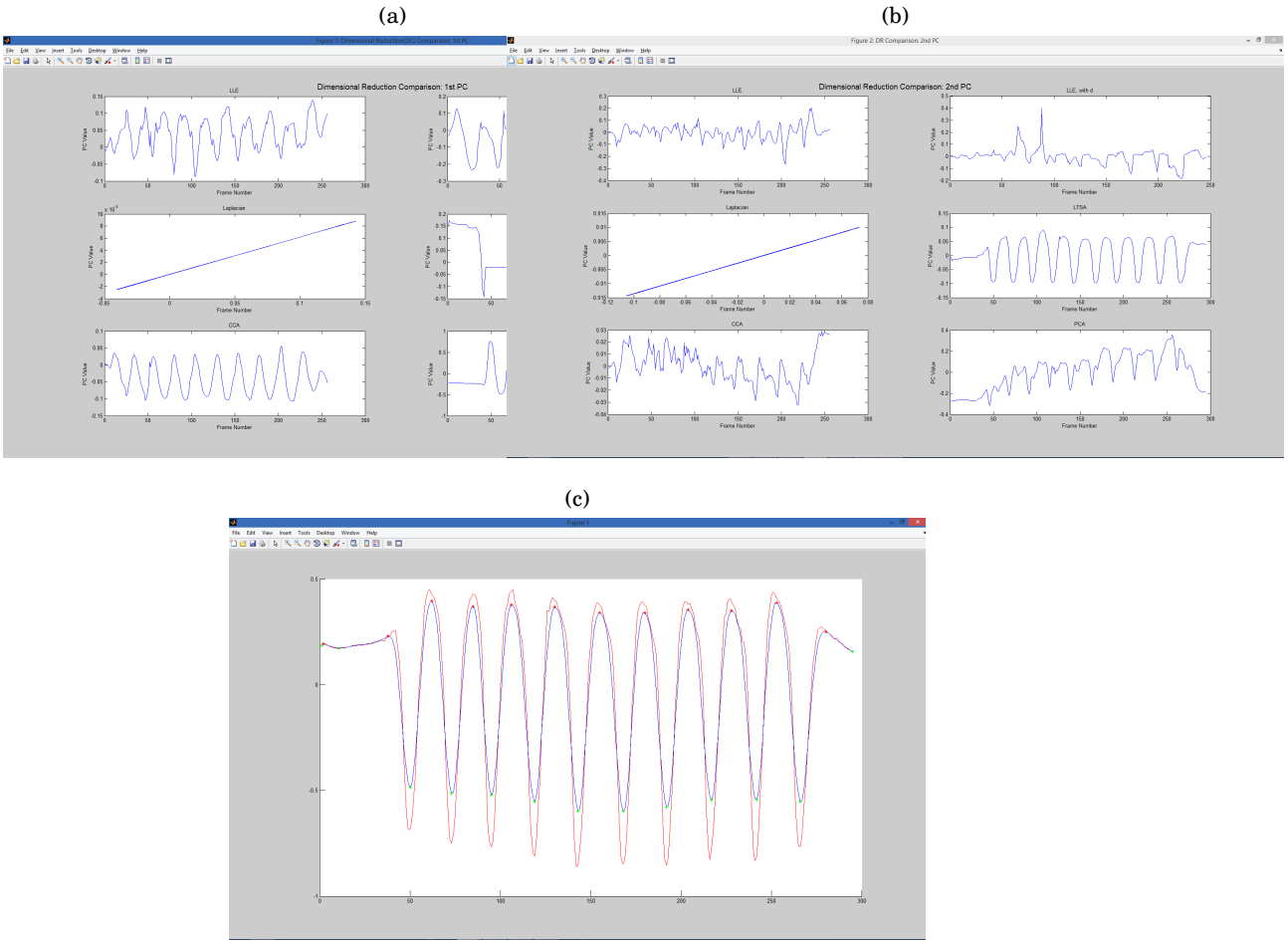


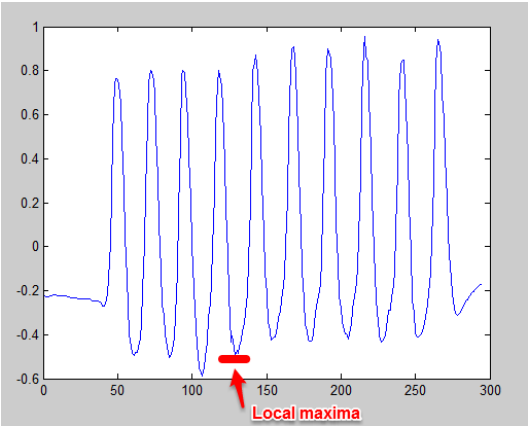
FIGURE 3.7. (a) Comparison of Dimensionality Reduction Techniques. PLotting first and second principal components.

1. Perform PCA on raw data.
2. Smooth the principal coefficients.
3. Find the local maxima and minima for the punch sequence.
4. Remove erroneous MinimaMaxima using heuristic rules.
5. Take a number of evenly spaced samples from between two maximum points, the length of one punch.
6. Fetch all PCA components that correspond to the time the samples were taken.
7. Train classifier on new data.

Each frame representing 20 joint positions is reduced from 60 dimensions to 3 principal components per frame. Next the principal components are smoothed to remove any **wrong minima/maxima** which would prevent the automated segmentation of punches. All remaining maxima are then passed through a set of heuristic rules that checks the location of a maxima point relative to it's neighbourhood points to determine if it is legitimate. The simplest form of this is using the Pythagorean theorem to calculate the distance between a maxima and it's neighbours, if the distance is too small it is clear that one of the points is erroneous and that only one of these will be necessary for segmentation. Likewise if a neighbour is too far away it becomes obvious that one of the maxima is incorrect and needs to be removed.image showing points really far down compared to baseline Thresholding is also used for each punch, with all maxima below a certain threshold removed since the cyclical signature of the punch guarantees the end of the punch will be approximately close to the end of the last punch.

I decided the best way to do this was to smooth the signal and use local maxima/minima as a means of successfully segmenting the beginning and end of each punch.

(a)



(b)

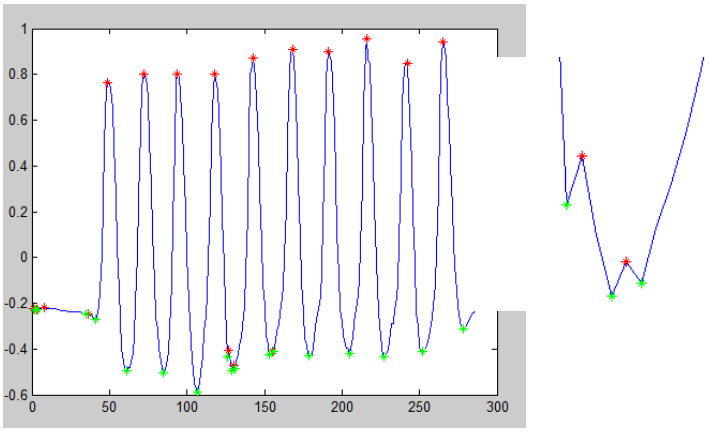


FIGURE 3.8. (a) Periodic punch signal over time (frame number).(b) Periodic signal with local maxima/minima labelled. (c) Close up of local maxima/minima.

Chapter 4

Results, Conclusions, and future work

Begins

4.1 Section

Begins a section.

4.1.1 Results

Begins a subsection. Avg Classification score random forest: 78.931Avg Classification score decision tree: 90.882

APPENDIX A

Begins an appendix

BIBLIOGRAPHY

[1] D. S. ALEXIADIS, P. KELLY, P. DARAS, N. E. O'CONNOR, T. BOUBEKEUR, AND M. B. MOUSSA, *Evaluating a dancer's performance using kinect-based skeleton tracking*, Proc. 19th ACM Int. Conf. Multimed. - MM '11, (2011), p. 659.

[2] D. S. BBC, *Usa boxing goes high tech in training in colorado springs*. http://gazette.com/usa-boxing-goes-high-tech-in-training-in-colorado-springs/article/1500163, May 2013.

[3] M. BELKIN AND P. NIYOGI, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Comput., (2003).

[4] S. BIANCO AND F. TISATO, *Karate moves recognition from skeletal motion*, 2012.

[5] E. B. C. KESKIN AND L. AKARUN, *A unified framework for concurrent usage of hand gesture, shape and pose*, CVPR - 2012.

[6] D. CATUHE, *Programming with the kinect for windows software development kit*.

[7] O. ÇELIKTUTAN, C. B. AKGÜL, C. WOLF, AND B. SANKUR, *Graph-Based Analysis of Physical Exercise Actions*, (2013).

[8] C. CHANG, B. LANGE, AND M. ZHANG, *Towards pervasive physical rehabilitation using Microsoft Kinect*, Pervasive . . . , (2012).

[9] S.-F. C. CHANG, YAO-JEN AND J.-D. HUANG., *"a kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities."*, ICIAR - 2011.

[10] R. R. COIFMAN AND S. LAFON, *Diffusion maps*, Appl. Comput. Harmon. Anal., 21 (2006), pp. 5–30.

[11] E. . P. S. R. COUNCIL, *New technology will help improve athletes' performance*. http://www.epsrc.ac.uk/newsevents/news/2009/Pages/improveathletesperf.aspx, 2009.

[12] P. DEMARTINES AND J. HERAULT, *Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets.*, IEEE Trans. Neural Netw., 8 (1997), pp. 148–54.

[13] J. DeVries, *Fighters uncaged review | ign*.
http://www.ign.com/articles/2010/11/09/fighters-uncaged-review, 2010.

[14] E. F. dynamics, *"elliott fight dynamic gloves"*.
http://elliottfightdynamics.com/, 2014.

[15] Gamesradar, *Fighters uncaged review | gamesradar*.
http://www.gamesradar.com/fighters-uncaged-review/, 2011.

[16] Gazette, *Usa boxing goes high tech in training in colorado springs*.
http://www.bbc.co.uk/news/technology-18735629, May 2013.

[17] T. Guardian, *London 2012 olympics: How athletes use technology to win medals*.
http://www.theguardian.com/sport/2012/jul/04/london-2012-olympic-games-sport-technolog
2012.

[18] K. Kaewplee, N. Khamsemanan, and C. Nattee, *Muay Thai Posture Classification using Skeletal Data from Kinect and k-Nearest Neighbors*.

[19] C.-C. C. L. Xia and J. K. Aggarwal., *A decision forest based feature selection framework for action recognition from rgb-depth cameras.*, ICIAR - 2013.

[20] ——, *View invariant human action recognition using histograms of 3d joints.*, IEEE CVPR - 2012.

[21] M. B. Manjuatha, B. P. Pradeep, and S. Y. Santhosh, *Survey on Skeleton Gesture Recognition Provided by Kinect*, (2014), pp. 8475–8483.

[22] Metacritic, *Fighters uncaged critic reviews for xbox 360*.
http://www.metacritic.com/game/xbox-360/fighters-uncaged/critic-reviews,
Jan. 2011.

[23] Microsoft, *Infrared stream - msdn*.
http://msdn.microsoft.com/en-us/library/jj663793.aspx, 2012.

[24] L. Miranda, T. Vieira, D. Martinez, T. Lewiner, A. W. Vieira, and M. F. M. Campos, *Real-Time Gesture Recognition from Depth Data through Key Poses Learning and Decision Forests*, 2012 25th SIBGRAPI Conf. Graph. Patterns Images, (2012), pp. 268–275.

[25] L. Miranda, T. Vieira, D. Martínez, T. Lewiner, A. W. Vieira, and M. F. M. Campos, *Online gesture recognition from pose kernel learning and decision forests*, Pattern Recognit. Lett., 39 (2014), pp. 65–73.

[26] C. R. K. G. V. R. B. R. Ofli, Ferda, *Sequence of the most informative joints (smij): A new representation for human skeletal action recognition*, 2012.

[27] R. E. B. P. U. PRESS, *"dynamic programming"*.

[28] M. RAPTIS, D. KIROVSKI, AND H. HOPPE, *Real-time classification of dance gestures from skeleton animation*, Proc. 2011 ACM SIGGRAPH/Eurographics Symp. Comput. Animat. - SCA '11, (2011), p. 147.

[29] K. RECTOR, C. L. BENNETT, AND J. A. KIENTZ, *Eyes-Free Yoga : An Exergame Using Depth Cameras for Blind & Low Vision Exercise*, Int. ACM SIGACCESS Conf., (2013).

[30] S. T. ROWEIS AND L. K. SAUL, *Nonlinear dimensionality reduction by locally linear embedding.*, Science, 290 (2000), pp. 2323–6.

[31] T. T. T. S. CELEBI, A. S. AYDIN AND T. ARICI, *Gesture recognition using skeleton data with weighted dynamic time warping*, VISAPP - 2013.

[32] R. T. M. G. P. K. D. M. P. D. A. D. S. ESSID, D. ALEXIADIS AND N. E. O'CONNOR, *An advanced virtual dance performance evaluator*, ICASSP - 2012.

[33] J. SHOTTON, A. FITZGIBBON, M. COOK, T. SHARP, M. FINOCCHIO, R. MOORE, A. KIPMAN, AND A. BLAKE, *Real-time human pose recognition in parts from single depth images*, Cvpr 2011, (2011), pp. 1297–1304.

[34] M. TURK AND A. PENTLAND, *Eigenfaces for recognition*. `http://www.cse.unr.edu/~bebis/MathMethods/PCA/case_study_pca1.pdf`, 1991.

[35] R. C. UK, *Cutting edge 2012: The research behind sport*. `http://www.rcuk.ac.uk/media/CuttingEdge2012/#bike`, 2012.

[36] ——, *Memorandum from research councils uk (rcuk) in response to the house of lords inquiry into sports and exercise science and medicine.* `http://www.rcuk.ac.uk/RCUK-prod/assets/documents/submissions/SportsExercise.pdf`, 2012.

[37] N. L. M. B. T. R. V. VENKATARAMAN, P. TURAGA AND S. L. WOLF., *Attractor-shape for dynamical analysis of human movement: Applications in stroke rehabilitation and action recognition*, HAU3D'13 - 2012.

[38] Z. Y. Z. W. Q. LI AND Z. C. LIU, *Action recognition based on a bag of 3d points.*, CVPR4HB10 - 2010.

[39] WIKIPEDIA, *Great britain at the 2012 summer olympics*. `http://en.wikipedia.org/wiki/Great_Britain_at_the_2012_Summer_Olympics`, 2014.

[40] K. YAMAOKA, M. UEHARA, T. SHIMA, AND Y. TAMURA, *Feedback of Flying Disc Throw with Kinect and its Evaluation*, Procedia Comput. Sci., 22 (2013), pp. 912–920.

[41] Z. ZHANG AND H. ZHA, *Nonlinear Dimension Reduction via Local*, in Intell. Data Eng. Autom. Learn., 2003, pp. 477–481.