REAL TIME SYSTEM FOR HUMAN ACTION EVALUATION

by

Randy Tan

Bachelor of Applied Science, Ryerson University, May 2015

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada 2017

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Real Time System for Human Activity Analysis
Master of Applied Science 2017
Randy Tan
Electrical and Computer Engineering
Ryerson University

Abstract

This thesis presents a real-time human activity analysis system, where a user's activity can be quantitatively evaluated with respect to a ground truth recording. Multiple Kinects are used to solve the problem of self-occlusion while performing an activity. The Kinects are placed in locations with different perspectives to extract the optimal joint positions of a user using Singular Value Decomposition (SVD) and Sequential Quadratic Programming (SQP). The extracted joint positions are then fed through our Incremental Dynamic Time Warping (IDTW) algorithm so that an incomplete sequence of an user can be optimally compared against the complete sequence from an expert (ground truth). Furthermore, the user's performance is communicated through a novel visual feedback system, where colors on the skeleton present the user's level of performance. Experimental results demonstrate the impact of our system, where through elaborate user testing we show that our IDTW algorithm combined with visual feedback improves the user's performance quantitatively.

Acknowledgments

I would like to sincerely thank Prof. Ling Guan and Prof. Naimul Khan for all their help as my supervisors. Their direction, creative ideas, and high expectations has shaped this research to what it is.

I would also like to thank the lab member of the Ryerson Multimedia lab. The friendly environment has made the two years of my Masters a very enjoyable time. The constant discussions we have had provided much needed insight and inspiration for my research. I would like to thank Leon Zhang for introducing me to the research lab. I would also like to thank Dr. James Smith, Lei Gao, Chengwu Liang, Nour El Madany, and Gareth Higgins for their help and discussions.

Lastly, I would like to thank my friends and family for their help and support. Their support and encouragement gave me the drive to accomplish what I have done in this thesis. I would especially like to thank those who volunteered for the user studies and also my brother Ryan Tan who helped facilitate recording the user studies and also helped me as a lab assistant.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Human activity analysis is the process of automated categorization\evaluation of different actions. Within human activity analysis, the evaluation of human actions has a wide range of applications,including physical rehabilitation, assisted living, telemedicine, entertainment, and fitness. In this thesis, the focus will be on the evaluation of human activities where the system must judge the quality of an action so that a user can improve their own performance. Traditionally, the evaluation process is manual, where an expert provides qualitative feedback to a user [1, 2]. The problem with this approach is twofold: 1) A human expert will have to be available, and 2) the feedback is mostly qualitative, and can vary from expert to expert. A universal automated activity evaluation process can greatly enhance the quality of life for many people.

There are many common techniques found between different branches of human activity analysis and it is worth investigating them. One popular topic under human activity analysis is human activity recognition where the goal is to be able to classify and segment actions into different groups. In both action recognition and action evaluation, spacial-temporal data needs to be interpreted into a meaningful output. The only difference is that for action recognition, the output is put through a classifier while in this thesis, the output is transformed into a representation that is easy for viewers to understand. Another popular research topic is human pose recognition where the goal is to classify certain parts of the human body and also possibly fit it to a model. While action pose recognition can have its own applications, it is often used as a preliminary step in other human activity analysis systems including our own.

Various types of sensors have been used as input for human activity analysis. The largest body of work uses RGB monocular video as input. One of the biggest challenges with using RGB videos as input is the reliance on visual information only [3]. One of the earlier methods of collecting accurate body pose information was to use motion capture systems where either active sensors are attached to the body (such as inertial or magnetic sensors) or passive markers are attached to the body and are tracked through IR cameras [4, 5]. Unfortunately, most of these sensor systems are expensive, have elaborate setups, and are usually invasive. The introduction of RGB-D cameras such as Microsoft's Kinect has sparked much interest in the research community. Different applications in human activity analysis have seen significant successes due to the re-introduction of depth information to traditional RGB video and also the real-time skeleton stream without expensive and invasive motion capture suits [6]. While the Kinect does offer a cheap real-time method for body tracking, it still runs into issues with self-occlusion, fast movements, and bone-length variations. Several works look into fixing these issues with Kinect tracking for different purposes [7, 8, 9, 10].

1.2 Objective

In this thesis, a novel system is designed and implemented to quantitatively evaluates a user's performance of an activity with respect to a pre-recorded ground truth and return a grade in real-time via a colored skeleton. Continuous repetition and rehearsal is a fundamental step in learning. While it may be necessary for an expert to be present to teach new skills to a user, it may not always be feasible for them to remain present during the rehearsal, especially if a single expert is in charge of multiple users. This system is meant to be used in conjunction with an expert to supervise the user in the repetitive parts of practice when the expert may not be available. Since the expert will mainly offer qualitative feedback, our system will focus on giving complementary quantitative feedback. As long as the user has received sufficient instruction from the expert, he/she should be satisfied with knowing if they are improving over multiple sessions or if they are repeating their mistakes.

The system is targeted for at-home general use. Rather than targeting absolute performance, many of the choices for this system were targeted towards ease of use, accessibility, and real-time performance. The use of multiple Kinect allows for decent tracking on a large range of natural movements without being invasive. The system was designed to be setup invariant meaning that the Kinect placement does not have to be the same between sessions. The calibration and grading feedback were also designed to be simple enough for any user to intuitively understand.

1.3 Contribution

This work presents the following contributions:

- **Improvement of an intuitive grading and feedback system for human activities.** The algorithm for grading and feedback is based off [11, 12]. One improvement we made to this system was removing its dependance on the same camera setup between sessions. During calibration, our system records the direction that the user is facing with respect to the main Kinect so that the camera setup does not need to be identical between the expert session and the user session. The other improvement made was disabling the functionality of ignoring limbs that did not move in the expert session. In many activities, posture is an important part of learning an activity properly. The system should not be ignoring limbs just because they did not move.

- **Addition of multiple-Kinect algorithm to increase range of trackable natural movements.** One of the most serious weaknesses of vision-based sensors is their dependance on perspective. Self-occlusion limits the number of activities that can be recorded, especially if the activity involves turning the body or crossing arms. By using multiple Kinects from different perspectives, as long as each body part can be seen by at least one Kinect without occlusion, the proper skeletons can be recovered.

- **User Experimentation.** In [11], only two simple activities were used to demonstrate that the algorithm could differentiate between poor and correct performances. This thesis performs the same experiment using 4 activities, and it also includes another experiment that shows that using the system improves learning compared to watching the recording without visual feedback. In [7], the experiments used limb length as their measure of error. Since the algorithm uses the limb length as a constraint, it was more heavily favored compared to the single Kinects. They also compared their resulting voted skeleton to the skeletons of each individual Kinect. While this ensures that the individual Kinects and the voted skeleton are using identical input, the voted skeleton is again favored due to the fact that the individual Kinects are placed at the sides and are therefore not in optimal positions when isolated. The experiment in this work will compare a single front facing Kinect to the multiple Kinect skeleton using a motion capture suit as ground truth.

1.4 Overview of Thesis

The rest of this thesis is organized as follows:

Chapter 2 - Literature Review: This chapter will go over all related works. The first section will go over the use of Kinect sensors in previous research and also compare them to other sensors such as RGB cameras and motion capture suits. Research using RGB methods are popular due to the widespread availability of RGB cameras but segmentation of body parts remains a challenge. Motion capture suits offer high accuracy tracking but the systems are not practical in most situations due to their cost and invasive nature. Kinects and other RGB-D devices leverage depth information to easily segment different body parts while being cheap and non-invasive. The largest concern when using RGB-D devices is mistracking due to occlusions in which one of the solutions is the use of multiple sensors.

Chapter 3 - System Design: This chapter will study the design of the proposed system and describe its individual parts. This chapter will be broken up into two parts. The first part will talk about how the multi-Kinect algorithm votes for the optimal skeleton. Singular Value Decomposition is used to find a Rigid Body Transform between Kinect cameras and Sequential Quadratic Programming is used to find optimal body joint positions using the skeleton from each Kinect as input. The second part of the algorithm will discuss the grading algorithm. Dynamic Time Warping (DTW) is used to temporally align a user sequence to an expert sequence to compare individual body limbs. The resulting DTW costs are then mapped to different colors which are used to overlay a colored skeleton on the user's replay as real-time visual feedback.

Chapter 4 - Experiments and Results: This chapter will present the experiments used to validate the system. The first part of the experiment compares the system and a single front-facing Kinect using a motion capture suit as ground truth. The results show that the system can mitigate lost tracking due to self-occlusions. The second part of the experiment will show that the system can differentiate between poor and good performances. The last part of the experiment separates the users into two groups where each group performs two complicated activities multiple times and only gets to see the visual feedback for one of the activities. Results indicate that users on average showed greater improvement when they saw the visual feedback compared to only seeing the replay with the feedback disabled.

Chapter 2

Literature Review

2.1 Use of Different Sensors in Activity Analysis

Over the years, researchers have used many input devices for human activity analysis. The vast majority of research for human activity analysis has been with RGB input [13]. RGB inputs have the benefit of being the most prevalent form of media that is commercially available. With applications such as automated content-based annotation and security, it is important for human activity analysis systems to be able to use current forms of input. Unfortunately, RGB based systems face a lot of challenges as they do not contain depth information [14]. Methods using RGB have to rely on techniques that rely on the appearance of the user in question. For human pose estimation, a popular method using RGB inputs is pictorial structures [15]. With these techniques, the human body is modeled as a graph where all the body's joints are connected by limbs. The objective of pictorial structures is to find the location of body parts given the appearance of an image, the prior probability of an image given the user's posture, and the likelihood distribution of body part locations with respect to each other. Models like these are popular due to the fact that the likelihood distributions can be more accurately trained by imposing the kinematic constraint that the limbs are connected in a set configuration. With variations in the appearance of humans RGB images and wide variations of human pose, it is necessary to create models such as these to give pose estimation systems consistent conditions to rely on.

Within action recognition, there are several techniques using RGB inputs that do not necessarily track a kinematic model of the human, but instead treat a video sequence as a space-time volume where only the most distinguishing motions only need to be captured to classify actions. [16] created a new feature detector specifically for capturing unique patterns in space-time sequences. Based off of the Harris detector, the concept was adapted to include the time axis in which corners found in 3D volumes are found as interest points and their descriptors can be used as an input for action recognition. [17] represents as an entire sequence as a single image. One of their representations is called a Motion Energy Image (MEI) which is a binary image where a pixel is 1 if motion occured in that pixel and 0 otherwise. Their other representation is called a Motion History Image (MHI) which is essentially a weighted summation of the previous frames. In both of these methods, high accuracy recognition is obtained by looking for pattern unique in the space-time volumes of each action but they completely discard the kinematic information of the subjects. In both of these methods, it is not possible to extract the individual pose or motion information at a specific frame since the volumes are constructed from the concatenation of multiple frames.

There is a large amount of variation in the appearance of certain body parts due to clothing, lighting, variability in body shape, and the loss of depth information. On the other side of the spectrum, there are several motion capture systems that prioritize high accuracy with elaborate setups [5, 4]. The motion capture systems are separated into different types: optical sensors, inertial sensors, mechanical sensors, magnetic sensors, and accoustic sensors. Optical sensor based motion capture systems have sensors placed on the subject's body that emit or reflect IR light that can be captures by cameras. As opposed to markerless optical systems, optical sensor based systems offer higher precision although they still fail with occlusion. Inertial systems use acceleration and rotational velocity measurements to calculate a user's posture. Magnetic systems use electromagnetic fields to release individual pulses to measure each individual axis of the sensors attached to the subject. Mechanical suits measure the displacement of joints using potentiometers. Acoustic systems use a combination of ultrasonic transmitters and microphones on different parts of the body to find the distances between them. Most of these systems are fairly expensive and require the subject to wear invasive sensors. Research with these sensors are typically left for ground truth measurements or for non-consumer applications like robotics [18]. In [19], The author mentions that sensor based motion capture systems are impractical for at home systems and RGB systems have several limitations.

Microsoft released the Kinect in 2010 as a RGB-D device that offered a low cost solution to obtaining depth information. Microsoft themselves also released a skeleton recognition algorithm that was highly attractive due to its low computation time [20]. The way that they achieved such a small computation time is through the use of simple depth comparison features and Random Decision Forests (RDF). Each pixel is run through the RDF and classified into body parts. The RDF contains several trees where each node contains a simple feature and threshold that determines which branch to take. Each simple feature is a comparison of two depth pixels in the area. At the leaves of each tree is a trained probability distribution of what class each pixel could be. While each feature does not give a good indication of the class, summing the distributions from multiple trees gives a sufficient classification of body parts.

While the Kinect offers a cost effective method of depth sensing, errors caused by occlusion limit the number of natural movements that can be recorded. Several methods have already been proposed to correct failed tracking including [7, 8, 9, 10]. Method [7] by Yeung et al. uses the skeletons from duplex Kinects facing the user at different perspectives to synthesize a completely new skeleton. Their skeleton optimization scheme has been adapted into the algorithm as it focuses on correcting occlusions and is computationally efficient. The method first uses Singular Value Decomposition (SVD) to find the perspective transform between the Kinects [21]. It then uses Sequential Quadratic Programming (SQP) iteratively solve the optimal joint positions [22]. The constraint for the SQP is that the length of the user's limbs must stay the same and the objective function is to minimize the weighted distance between the optimal joint position and joint positions reported by the two Kinects. The weights are determined by the probability of each Kinect reporting the joint position properly.

There are other, more accurate methods such as [8] and even commercial systems such as ipisoft (http://ipisoft.com/), but the accuracy comes at the price of high computational costs. In [8], the kinects are only used for their depth frames. The depth frames from multiple perspectives are stitched together to eliminate the effect of occlusions. The system then fits the depth information to a Shape Completion and Animation of PEople (SCAPE) model. The advantages of using the SCAPE model is that it handles nonrigid deformations like the movement of skin and muscles and can match a sparse set of tracked points onto the model. The downside of this method is that the computations takes about 10 seconds per frame on average and is therefore not viable for real time applications.

Kaewplee et al. improves single Kinect skeletal tracking mainly relying on the skeleton frame but using the depth frame for hints [9]. Their paper focused muay thai maneuvers in the experimentation but the algorithm corrects for fast movements and partial occlusions where the depth frame can at least see part of the occluded body part. In their algorithm they try to correct any joints that are reported as not tracking or moved too fast. Each joint being corrected is moved to the previous good position. The algorithm also checks the depth frame for the *boundaries* of the subject defined ad the left most, right most, and closest (to the Kinect) pixels in the subject's depth frame silhouette. For each of those three pixels, the closest joint to each of those pixels is forced to that coordinate and the skeleton is realligned. While their method handles mistracking due to very high speed movements well, their method can only handle partial occlusions which is the weakness of using only a single perspective. Unfortunately, their paper does not mention the execution time of their algorithm per frame and only mentions that the input videos were recorded at 30 fps.

There is a newer method of using multiple Kinects to improve skeleton tracking [10]. In their method, the Kalman filter is used to try to predict the location of joints in the current frame based off the locations in the previous points along with their trajectory. After predicting the current location, the algorithm then weighs the accuracy of the joint positions given by each Kinect according to how close it was to the prediction and then averages all the positions. This algorithm has an advantage over [7] in that this method takes temporal information into account.

2.2 Human Activity Analysis

The use of computers to understand and analyze human activities has been a large part of academic research for many years. Most of this research has been focused on action recognition where the goal is to be able to segment and classify actions into groups. The challenge for action recognition is to be able to define each class such that all instances of the action is recognized despite inter-class variations while excluding all instances of other actions despite their similarities. In this thesis, we focus on the evaluation of human actions in which we have a single desired ground truth and the challenge is to interpret the input data such that such that the spacio-temporal properties are preserved as much as possible. That being said, it is worthwhile to explore different methods of action recognition to understand how each of them affect the spacio-temporal properties of the data.

Action recognition and evaluation can both be broken up into two steps: first the input data is transformed into descriptors and then the descriptors are run through a classifier. Descriptors contain kinematic information like a subject's posture and motion information and transforms the data to segment the classes easier. One popular method in feature processing is to pull in as much information as possible and then use dimensionality reduction methods to turn the data into a compact form that represents the variability best. [23] extracts data representing current pose by comparing all current unique joint pairs, motion information by comparing each current joint with the previous frame's joints, and initial offset information by comparing the current joints to the joints in the initial frame. The features combined result in a 2970 dimension vector, but Principal Components Analysis (PCA) is used to reduce the dimensionality of the data while preserving the variance in the data. Dimensionality reduction techniques like PCA are popular due to the ability to initially take in a large amount of data that can discriminate between classes and then reduce it to a compact form for fast processing.

[24] converts an entire skeleton into a histogram of 3D joints with respect to the waist. Each of these histograms is fed through 's Linear Discriminant Analysis (LDA) to reduce the dimensions to the number of classes -1. LDA is a strong dimensionality reduction tool since it prioritizes separating classes rather than preserving variance but unfortunately it can only be used if the number of classes is known and there is no control over the number of resulting dimensions.

The classification comes in the form of a Hidden Marcov Model (HMM). HMMs are mathematical models that can identify sequences. The model describes a system that is assumed to include states that can not be observed and the probability of changing states only depends on the current state. Each state then has their own probabilities of having certain observed features. HMMs have a *forward algorithm* that can determine the probability of a sequence of observations belong to a certain model. By modeling each class as a HMM, the forward algorithm can be used to determine which class best describes a sequence. HMMs benefit from their sequential nature in that they are invariant to the time it takes to complete an action.

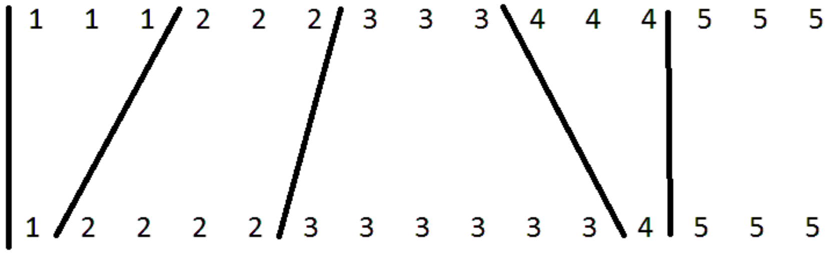


Figure 2.1: **Sequence matching using DTW.** Two sequences are composed of values from $1-5$ but at different rates. DTW locally warps the time axis to find the best match between both the sequences

Dynamic time Warping (DTW) is a pattern matching algorithm originally used for speech processing [25]. DTW's main strength is its invariance to speed. DTW allows matching of two time series of varying rates by locally warping the time axis. This local warping allows a sequence at a slower rate to be compared to a similar sequence at a faster rate without being punished. DTW has been used successfully in many applications such as hand writing recognition, gesture recognition, and information mining [26, 27]. The algorithm maps a query sequence to the template sequence as shown in 2.1. The algorithm starts at the beginning of both sequences and finds the path that minimizes the cumulative distance between warped sequences A and B, $D(A,B)$ shown as:

$$D(A,B) = \frac{1}{N}\sum_{t=1}^{T} d(P_t) \tag{2.1}$$

In which $P_t \in P$ is the warping path and each individual step $P_t = \{A(i), B(j)\}$ contains the i^{th} sample from A and the j^{th} sample from B. $d(P_t)$ represents a distance function that is appropriate for the application. N is a normalization factor such that Seeing as both sequences are assumed to be similar sequences with variances in temporal rate, a few restrictions are placed on the warping path to ensure that the final mapping preserves the sequential properties of both sequences.

- **Monotonicity condition**: Both sequences should not be able to move backwards in time: $P_t = \{A(i), B(j)\} \rightarrow P_{t+1} = \{A(i'), B(j')\}, i \leq i', j \leq j'$
- **Continuity condition** : Both sequences should not skip samples: $P_t = \{A(i), B(j)\} \rightarrow P_{t+1} = \{A(i'), B(j')\}, i' - i \leq 1, j' - j \leq 1$
- **Boarder condition** : The warping path must begin at $P_1 = \{A(1), B(1)\}$ and end at $P_T = \{A(N), B(M)\}$ where N and M is the length of sequences A and

B

DTW started as a speech processing technique and was popularized due to it's property to find similarity despite differences in rate. While DTW has been used before in human action recognition, it has some weaknesses in this field. When used to classify signals, it becomes computationally expensive since the technique that measures similarity per sample and therefore has quadratic time complexity [28]. Quantization makes DTW harder to classify and it must compare itself to multiple samples per class in order to overcome inter-class variations [29]. Other popular methods such as [23, 24] can afford to create compact feature descriptors that still retain reach class' discriminative properties and also compare directly to a class rather than to individual sequences. While these methods are preferred in action recognition, DTW becomes more advantageous in action evaluation. First of all, in action evaluation there is only a single ground truth and not multiple classes to compare to so DTW is no longer hindered by its inability to compare to multiple sequences at the same time. Secondly, the purpose of action evaluation is to explicitly punish deviations from the ground truth and therefore quantization and dimensionality reduction could potentially hinder action evaluation as opposed to directly comparing the raw data.

Another interesting type of feature uses angles from triplets of joints [30]. In their paper, 35 angles are taken from a skeleton where each angle is formed from 3 specific joints. One benefit that this type of feature has is that the angle is only rooted to the skeleton and therefore these features are view-invariant. Another benefit that this feature type has is that each angle is derived from 3 joints which means that each angle holds information on how all the joints are interacting with each other instead of containing information for only a single joint. Lastly, since these features are angle-based, they are invariant to the size of the subject's limbs.

There have been previous works that look into evaluating human actions in order to improve the performance of select activities [31, 32, 12]. [31] used kinects to observe a vehicle driver's posture as a input for driver assistance systems. They argue that a driver's posture could be indicative of distracted driving. Their researched looked into possible features that could indicate movements associated with safe or unsafe driving. The inputs used included both the skeletons and the depth frames. The features they extracted included the mean, standard deviation, and motion of the head and arms of the driver. While the method explored different types of features, the study was conducted as a proof of concept and results on classification are not shown. [32] use classification based methods to evaluate a user's performance in golf [32]. In their work, they set the golf swings into 4 qualitative categories from perfect to poor in order to take advantage of popular classification methods such as Support Vector Machines (SVM) and Gausian Mixture Models (GMM). [12] created a system to teach ballet using Kinect. The interesting part of ballet is that there are basic postures and moves that are common between different routines. Their paper takes advantage of this fact by using a Spherical Self Organizing Map (SSOM) to learn the postures and how they transition to each other, Similar to our system, they propose using DTW to align a user sequence to a ground truth. By using a SSOM, their system has the ability to focus more on key postures within a specific type of activity and weigh them more when evaluating the performance.

[33] created a system with Kinect that would offer quantitative advise based off of their performance. The paper's goals are quite similar to this thesis in that they wished to design a system that could assist users in learning different activities. Their system used angles from joint triplets as features. In order to evaluate the performance of activities, hard coded thresholds for specific angles were set for each exercise. The feedback for their system was a qualitative audio message that played when certain thresholds were met or failed. In their user study, they concluded that while their users felt more engaged with the qualitative audio feedback, the quantitative results were poorer when the feedback system was enabled. One flaw of the system could be that it was targeted towards replacing a coach in evaluating performance and giving feedback. The system analyzed quantitative information such as joint angles and then tried to translate it to qualitative feedback which requires high level semantics. A system for the evaluation of activities would benefit users more if the quantitative data is given back to the user as is and used as a tool to supplement qualitative advise given by a coach.

Chapter 3

Proposed System

In the proposed system, multiple Kinects are used to allow for cheap and robust tracking for at-home activity evaluation. While RGB cameras are the most common sensor, most RGB-based systems can not handle high accuracy tracking in non-laboratory enviroments. Also, the high costs and invasive nature of sensor based motion capture systems are not suitable for consumer use. Kinect-based systems are ideal for commercial use as the sensors were originally intended for consumer use. While the Kinect does suffer from occlusion problems, the use of multiple Kinects can remedy this problem and are still far cheaper than most motion capture system. Many motion capture systems cost several thousand dollars while Kinects cost approximately a hundred dollars. The most important factors that go into the design of the proposed system is its usability for the average consumer. Rather than targeting the highest possible tracking accuracy, this system is optimized for real-time performance while having sufficient accuracy for later evaluation. The feedback system is designed to be intuitive and easily understood. A color coded skeleton is overlaid on the user to indicate performance as it transfers information to the spatial portion of working memory better than printed text such as a numeric score [34].

The proposed system requires at least two Kinects connected to the computer to start up successfully. A flow diagram of the system with two Kinects is shown in figure 3.1. While all work in this thesis is done based on this system, the generalization to multiple Kinects is straightforward. At start-up, the system will choose which of the dual Kinects will be the main camera and the secondary camera in which all joint positions will be transformed to the main camera's perspective and the final voted skeleton will be displayed in the main camera's video feed. After start-up, a calibration must be performed by the user which allows the system to find the rigid body transform between the perspective of both the Kinects. The calibration will save the transform from the secondary Kinect to the primary Kinect, the length of all the user's limbs, and also the initial direction that the user was facing. After calibration, the user will have the option to either record an expert session or go through a play session using a previously recorded expert. In both sessions, the two Kinects start to record in real time. While both video feeds update every time a new frame from their respective Kinects updates, the optimized skeleton is only voted for every time the main camera has a new frame. Whenever the optimized skeleton is being voted for, the system assumes that both cameras are updating at 30 fps and therefore synched. Once the optimized skeleton is voted for, the resulting joint positions are converted to normalized unit vectors

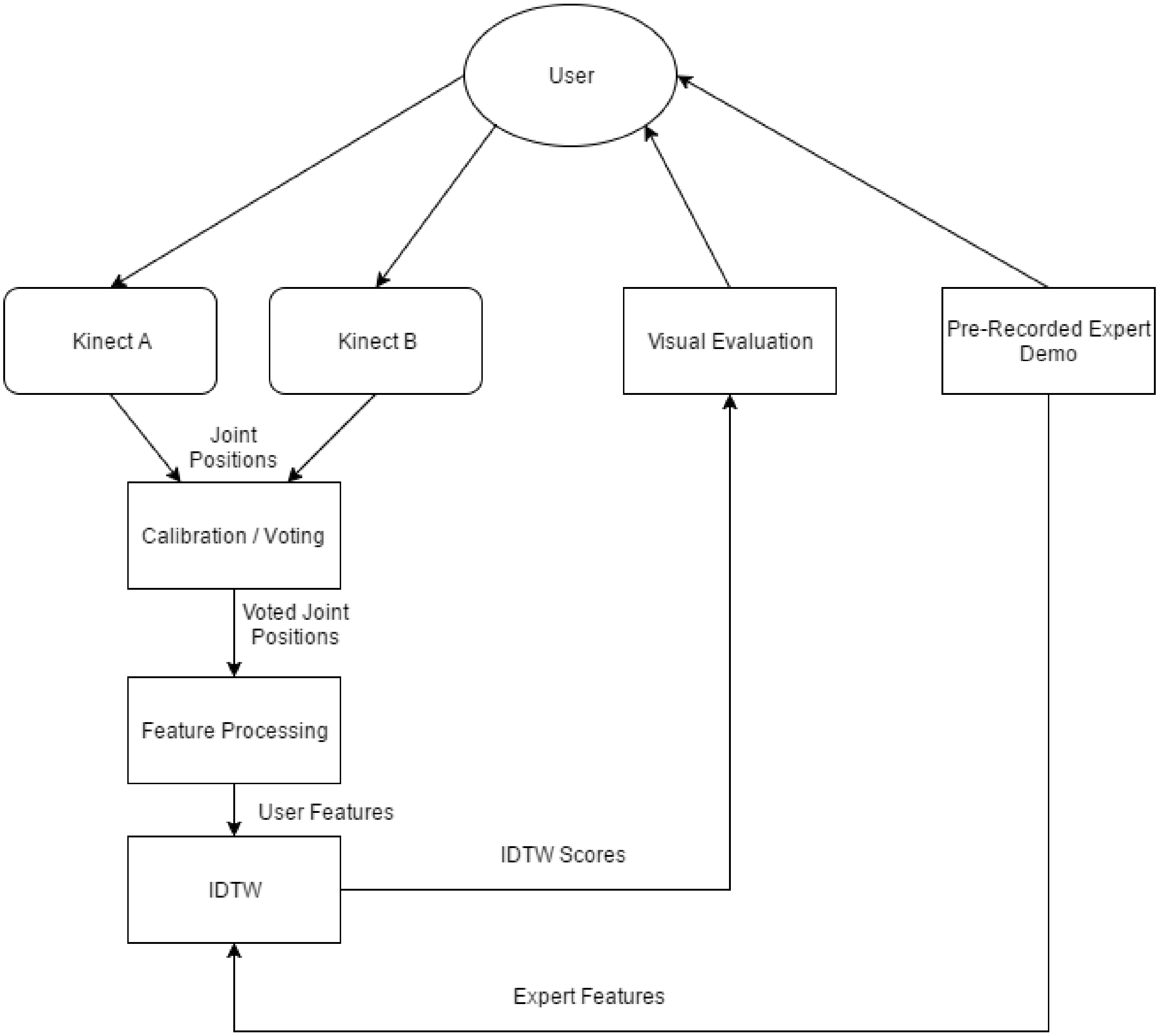


Figure 3.1: **Flow Diagram of the Proposed Pystem**

with respect to their parent joint in the feature processing step. Figure 3.2 shows the parent-child relationship of all joints. Within the play session, each joint is graded separately and the scores are converted to a color scale in which the voted skeleton is then displayed using the main camera's video feed.

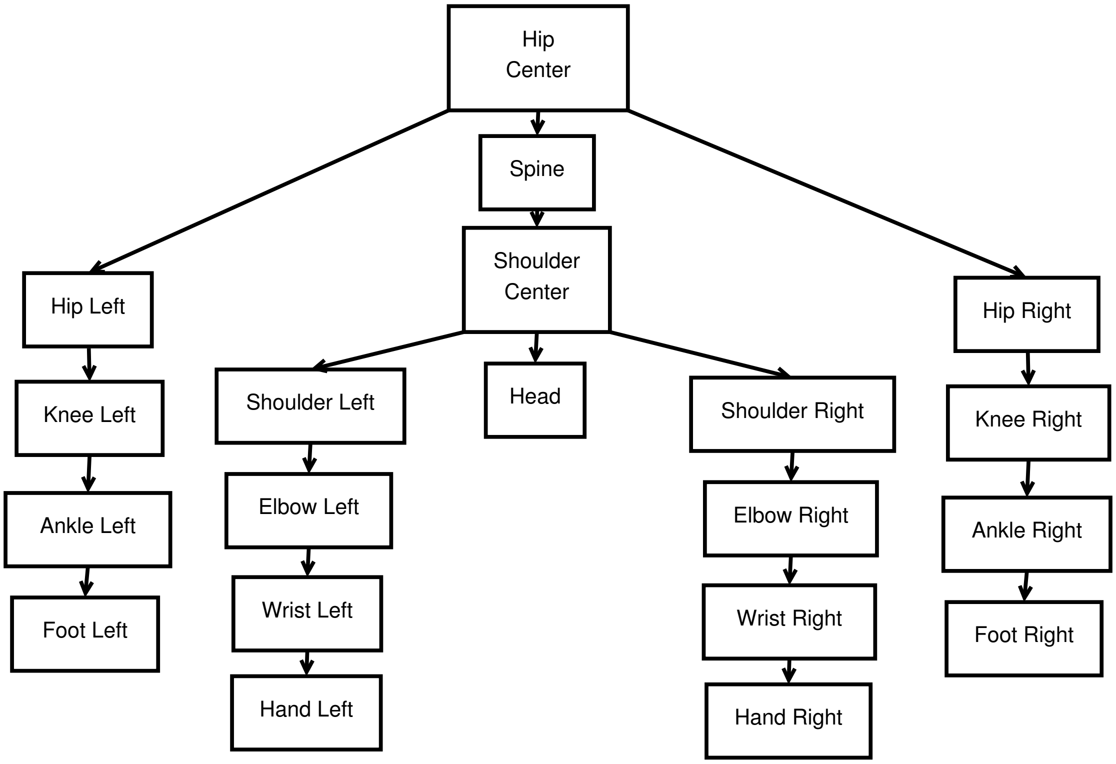
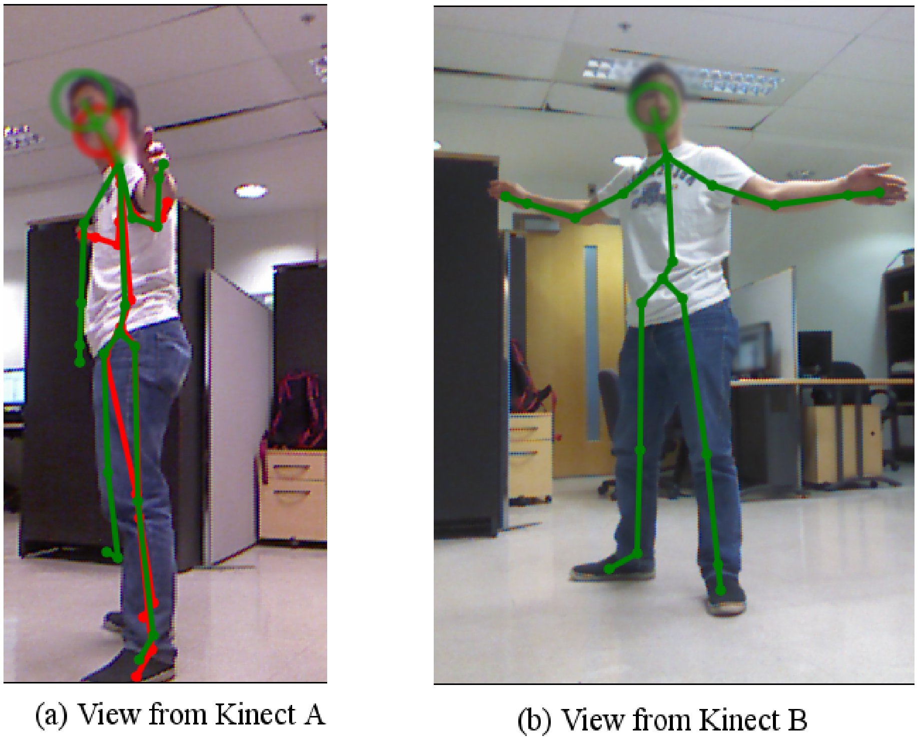


Figure 3.2: **Tree Representation of the Parent-Child Relationship Between Joints**



(a) View from Kinect A

(b) View from Kinect B

Figure 3.3: **Voting system with two Kinects**. The green skeletons are the reported joint positions from both Kinects. The red skeleton is the final voted skeleton from Kinect A's perspective.

3.1 Skeleton Voting Using Sequential Quadratic Programming

Figure 3.3 shows how Kinects from multiple perspectives can correct mistracking due to occlusion. In the example, a subject was asked to stand in a position with arms and legs out with each Kinect approximately 45° from the front of the user with one Kinect to the left of the user and one to the right so that both Kinects get a good view of the user and are approximately orthogonal. Calibration was done when all joints were visible. The subject then rotated so that the view from Kinect A was occluded and the view from Kinect B was fine. The reported skeletons are displayed in green while the voted skeleton is displayed in red. In this situation, Kinect A does not track the subject's right arm and leg properly and the reported joint positions are left at the last known good position. Since Kinect B is reporting proper positions, the final voted skeleton is able to reasonably place the right arm and leg from the viewpoint of Kinect A.

As explained earlier, Multiple cameras are used to solve for self occlusions. In order to use multiple Kinects within the same voting system, all of the reported joint positions must be translated to the same coordinate system. Since our system will eventually show visual feedback as a colored skeleton, all joint positions are translated to the coordinate system of the Kinect sensor that is displaying the final skeleton. In the current duplex setup, only one sensor's coordinates are being transformed to the other, but if more sensors are used then each sensor except for the first one would need their own transform for their coordinates. The calibration is initiated by the users as a button on the UI so that the users can initialize the calibration after they can see that they are being tracked properly by both Kinects and can re-calibrate if they are unsatisfied with the result. When the calibration button is clicked, the joint positions from the current frame of each Kinect is fed through a Rigid Body Transform using SVD to extract a rotation and translation to the main camera's position [35].

The Rigid Body transform between the two Kinects consists of a rotation and then a translation [35]. In order to find the transformation, a set of joint positions are simultaneously taken from both Kinects when the user clicks the button. Within a single frame, Kinect A and B track skeletons S_A and S_B. The joint positions P_{A_i} and P_{B_i} come from the i-th joint of skeletons S_A and S_B in a single frame. These joint positions should have a rotation R and translation t such that $P_{B_i} = RP_{A_i} + t$. The first step in finding the rotation matrix is to calculate the correlation matrix H through:

$$H = \sum_{i \in S_A, S_B} (P_{A_i} - \bar{P}_A)(P_{B_i} - \bar{P}_B)^T \tag{3.1}$$

where $\bar{P}$ is the centroid of the skeletons calculated by:

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i \tag{3.2}$$

Once the correlation matrix has been found, the Singular Value Decomposition yields $\mathrm{SVD}(H) = U\Lambda V^T$ and the rotation matrix R can be found by:

$$R = VU^T \tag{3.3}$$

After the rotation matrix is calculated, the translation vector can be found by using the definition of the rigid body transform and substituting the centroids as points:

$$\begin{aligned} \bar{P}_B &= R\bar{P}_A + t \\ t &= \bar{P}_B - R\bar{P}_A \end{aligned} \tag{3.4}$$

The resulting rotation and translation matrices can be applied to any joint position from Kinect A to translate its perspective to Kinect B's.

The skeleton voting system proposed by Yeung et al. uses Sequential Quadratic Programming (SQP) as an optimization technique for finding joint positions. Occlusion and mistracking with multiple Kinects can cause joints to be reported as successfully tracked, but be far from each other. In these situations, guessing incorrectly or simply taking an average of both positions would not yield the optimal result. Rather than relying purely on the reported joint positions which can be inconsistent, matching the limb lengths taken from the calibration step is used as a hard constraint for the optimization while getting the final voted joint position to be as close as possible to the reported positions. Mathematically, the problem we wish to optimize is:

$$\min \sum_{i \in S_A, S_B} w_{A_i} ||P_{V_i} - P_{A_i}||^2 + w_{B_i} ||P_{V_i} - P_{B_i}||^2,$$
$$\text{s.t.} \sum_{i,j \in S_A, S_B} ||P_{V_i} - P_{V_j}||^2 - l_{i,j}^2 = 0 \tag{3.5}$$

where:

- w_{A_i} and w_{B_i} are weights assigned to the distance between the final voted joint position and the reported positions from Kinects A and B respectively and will be explained later within this section
- P_{V_i} and P_{V_j} are the final voted joint positions of the current joint and the parent joint (refer to figure 3.2) and
- l_{ij} is the limb length that was recorded in the calibration stage

The hard condition has been improved in this work from [7] for more efficient calculation. The original condition was

$$\sum_{i,j \in S_A, S_B} (||P_{V_i} - P_{V_j}|| - l_{i,j})^2 = 0$$

in which both statements are only true as long as the voted limb length is the same as the limb length recorded from the calibration stage. Since the calculation of 3D distances requires squaring and square rooting, our condition removes redundant rooting by keeping all distances squared.

Figure 3.4 illustrates how this optimization equation finds the optimally voted

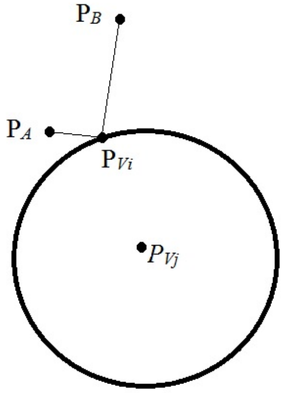


Figure 3.4: **Voting for one incorrect joint**. P_A and P_B are the reported joint positions from Kinect A and B where P_A is fairly close to the actual joint position, but P_B is mistracking. P_{V_j} represents the position of the parent joint and the circle represents the radius of the limb length that is set at the calibration stage. P_{V_i} is the position of the final voted joint.

joint. In this situation, both Kinects are reporting their joint positions as "tracked", but have their positions far away from each other. P_B is mistracked by a far margin while P_A is fairly close to the actual joint position. Taking a simple averaging of P_A and P_B would bring the voted joint position further away from the the actual joint position than if only P_A was used, but only relying on the "better" joint position will not always guarantee good results. By constraining the final joint position to the limb lengths found in the calibration stage, we can use information reported by both Kinects and find an optimally voted solution that at least preserves limb length.

The weights w_{A_i} and w_{B_i} need to numerically represent how reliable the reported joint positions from Kinect A and B are. The two situations in which a reported position is not as reliable is when the joint position is reported as "inferred" instead of "tracked" and when occlusion causes mistracking [7]. Whenever occlusion causes mistracking, the mistracked joint typically snaps to another part of the skeleton. The most common example is when arm joints accidentally get tracked onto a user's waist. Therefore, we can assume that if one kinect is mistracking, it's reported position will be a lot closer to another joint poisiton on the body as compared to a properly tracked joint. Calculating the weights starts with equation 3.7.

$$d_{A_i} = \min_{k \in S_A, k \neq i,j} ||P_{A_i} - P_{A_k}||^2 \qquad d_{B_i} = \min_{k \in S_B, k \neq i,j} ||P_{B_i} - P_{B_k}||^2 \tag{3.6}$$

where P_{B_k} is the position of every other joint in the skeleton except for joint i and it's parent j. The distances from the closest joints are incorporated into an initial weighting by:

$$\bar{w}_{A_i} = \frac{d_{A_i}}{d_{A_i} + d_{B_i}} \qquad \bar{w}_{B_i} = \frac{d_{B_i}}{d_{A_i} + d_{B_i}} \tag{3.7}$$

In order to incorporate the Kinect's tracking state into the weighting, variables h_i^A and h_i^B are used in creating the final weights. Their value will be assigned a value h when the joint is properly tracked and $(1-h)$ whenever the joint position is being inferred where $h \in (0.5, 1)$ and can be tuned. All examples in this paper use h=0.9. By using a larger h, the weighting relies more on the joint being tracked.

The final weights are calculated by:

$$w_{A_i} = \frac{(\bar{w}_{A_i} h_{A_i})^4}{(\bar{w}_{A_i} h_{A_i})^4 + (\bar{w}_{B_i} h_{B_i})^4} \qquad w_{B_i} = \frac{(\bar{w}_{B_i} h_{B_i})^4}{(\bar{w}_{B_i} h_{A_i})^4 + (\bar{w}_{B_i} h_{B_i})^4} \tag{3.8}$$

The final objective function which needs to be solved is:

$$\begin{aligned} \ell(P_{V_i}, \lambda) = \sum_{i \in S_A, S_B} & w_{A_i} ||P_{V_i} - P_{A_i}||^2 + w_{B_i} ||P_{V_i} - P_{B_i}||^2 \\ & + \lambda \sum_{i,j \in S_A, S_B} ||P_{V_i} - P_{V_j}||^2 - l_{i,j}^2 \end{aligned} \tag{3.9}$$

where lambda is the Lagrange multiplier. Note that since the constraint is an equality, one can choose to either add or subtract the constraint as long as the operation remains consistent. SQP uses Newton's method to solve equation 3.9 iteratively through a line search. Within each iteration of the line search, the following linear system is solved:

$$\begin{pmatrix} \nabla^2 \ell(P_{V_i}^k, \lambda^k) & \nabla g(P_{V_i}^k) \\ \nabla g(P_{V_i}^k)^T & 0 \end{pmatrix} \begin{pmatrix} \tau_p \\ \tau_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla \ell(P_{V_i}^k, \lambda^k) \\ g(P_{V_i}) \end{pmatrix} \tag{3.10}$$

where:

- $P_{V_i}^k$ and λ^k are the values of P_{V_i} and λ at iteration k
- τ_p and τ_λ are the steps towards the next iteration: $\{P_{V_i}^{k+1}, \lambda^{k+1}\} \leftarrow \{P_{V_i}^k + \tau_p, \lambda^k + \tau_\lambda\}$

- ∇ and∇^2 are the Jacobian and Hessian of a matrix and
- $g(P_{V_i})$ is the constraint set in equation 3.5

In our adaptation of the algorithm, we set the initial guess for each joint as the position of the Kinect with the higher weighting as opposed to the position of the previous optimized skeleton. In our tests, we found that in situation when the algorithm makes a mistake and places a joint in the incorrect position, the SQP would get stuck at the absolute maximum and point in the opposite direction of the optimal position. Unfortunately, this change results in a large amount of jitter of the hand positions since the difference in hand positions of the individual Kinects is frequently larger than the size of the hand itself. Our algorithm also takes advantage of the efficient numerical scheme as outlined in [7]. We end the line search when
$\max\begin{pmatrix}\tau_p & \tau_\lambda\end{pmatrix}^T \leqslant 0.1^4$ or when 50 iterations have been reached.

3.2 The IDTW Algorithm and Grading

A similar grading scheme is used as proposed in [11, 12]. IDTW is an efficient means of comparing an incomplete sequence to a fully completed ground truth. When going over the algorithm of DTW per frame, it is important to realize that the distance costs are the same and only costs for the current frame needs to be calculated. Also, while in conventional DTW both sequences are complete, in IDTW, the query sequence is still incomplete. Therefore, it is necessary to lower the restriction on the path so that it goes through all frames of the incomplete query sequence, but only goes through the first set of frames in the ground truth sequence that best matches.

One major difference between [11, 12] and this thesis is that the minimum movement requirement for a limb to be graded has been removed. While this feature made the algorithm more computationally efficient, it is important to recognize that certain exercises require proper posture meaning that limbs can not be disregarded from grading just because it doesn't move. Moving from the skeleton voting to the grading, only the final voted skeletons from both the recorded expert and the user are considered for grading. Grading using IDTW is done on a per-joint basis in order to have the real time feedback be done on a per-joint basis. Each joint's position is normalized by having the coordinates from the parent subtracted and the result is divided by limb length. Each normalized joint coordinate can be calculated by $J_i = \frac{P_{V_i} - P_{V_j}}{||P_{V_i} - P_{V_j}||}$. This normalization allows the system to accommodate for people with different limb sizes and ratios [36]. Since the hip center can be considered as the root of the skeleton and therefore always at $(0,0,0)$, it is not considered in the IDTW calculations.

For DTW calculations, time sequences of the expert E with M frames and the user U with N frames are compared in a grid. For each cell (U_a,E_b) in the grid, the distance between the normalized joints J_i are compared for the sequence U in time index a and E at time index b. For classic DTW calculations, an optimal warping path between the two sequences is calculated:

$$\mathbf{D_i}(\mathbf{U},\mathbf{E}) = \frac{1}{N}\sum_{t=1}^{T} ||{J_i}^{U_t} - {J_i}^{E_t}||, \qquad i \in S_U, S_E. \tag{3.11}$$

where T is the total grid cells taken in the warping path, t corresponds to each grid coordinate (a,b) in that path, and S_U and S_E are the set of all joints in the skeletons U and E. In the classical DTW approach, the minimum path is required to reach from the bottom left of the grid to the top right. Since our application is real-time, the full sequence for the user is not complete and therefore the requirement for the sequence to reach the top right needs to be relaxed. In other words, we represent the IDTW equation as:

$$\mathbf{D'}_i(\mathbf{U},\mathbf{E}) = \min_{c=1,\ldots,M} \mathbf{D_i}(\mathbf{U},\mathbf{E}^c), \tag{3.12}$$

where $\mathbf{D_i}(\mathbf{U},\mathbf{E}^c)$ is the same as in equation 3.11 except the requirement to end at the top right cell of the grid has been relaxed to end anywhere on the right-most column of the grid that achieves the minimum DTW cost. For each iteration of IDTW, the grid calculations are saved so that only the right-most column needs to be calculated when new frames come in. To further increase computational efficiency, a sakoe-chiba band is implemented [25]. Under the assumption that both the expert and user recordings will mostly only contain the desired action, global constraints are placed on the warping path: $||a-b|| \leq r$ where r is a predefined radius. In this work, the radius is set to one quarter of the expert sequence length.

By placing these global constraints, pathological warping is avoided and computation time is also saved by not calculating grid cells outside of the band. Finally, to calculate the score of each limb in real time, we use:

$$Z_i = e^{-\nu * (\mathbf{D'}_i \mathbf{D'}_j)/2}, \tag{3.13}$$

where the score of each limb Z_i corresponds to a joint i and its parent j and ν is a parameter to control the score's sensitivity to mistakes. Finally, the value Z_i is projected to a color map. The color map used in this work is blue-aqua-green-yellow-red, i.e., the color stays closer to blue if the user is doing well, and shifts towards red as the performance worsens.

Chapter 4

Experiments and Results

In this chapter the proposed system in this thesis is validated through user experiments. The goal of these user experiments is to not only verify that the algorithms can track and evaluate users, but to also show that using the system can increase performance. The user study is separated into three sections. The first part of the experiment shows the system's improved tracking compared to a single front facing Kinect. The second part of the experiment shows that our system can differentiate between good and poor performances pinpointing the source of errors. The third part of the experiment shows that our real-time feedback system can allow users to learn an activity much more efficiently than simply watching and imitating a video. In all of our tests, we had 2 lab members act as "experts" and 8 participants act as "users" with a mix of different gender, age, and demography. In all 3 parts of the experiment, four simple exercises (bar curl, horse stance (from karate), marching, vertical press) and two complicated tai chi exercises (brush

Knee, parting horse's mane) are used. While the numeric scores can give a decent indication of overall performance, it does not give a clear picture of when and where mistakes were made in the recordings. To get a better idea of the performance of each limb at any given point in time during the exercises, readers are recommended to watch video demonstrations of the exercises used in the experiment at: https://youtu.be/yYIDoiGzfEo and https://youtu.be/rS6Gd5M9o90.

4.1 Multi-Kinect Performance Test

In the first part of the experiment, only the experts are used for comparing Kinects to motion capture suits for all 6 exercises. The first expert performs the bar curl, marching, and brush knee while the other expert performs horse stance, vertical press, and horse's mane. The experts perform each exercise with the dual Kinect and single Kinect separately. The graphs shown in each figure show the distances between the joint positions reported by the motion capture suit and the Kinects per frame. Figure 4.1 shows the legend for all of the graphs. The joint positions have been converted to a unit vector relative to their parent joint since that is what is being fed into the DTW algorithm. The exercises were recorded multiple times to make sure the mistracking consistently occurred for the single Kinect but only one session is shown. The motion capture suits places joints in slightly different locations than the Kinect so there will always be a static amount of distance between them. Areas of interest in the graphs will therefore be the spikes in the graphs. For each graph, all spikes in the single Kinect session will be marked on the top graph and the equivalent frame (the closest peak) for the dual Kinect session will be shown on the bottom graph. For every marked frame, the 3D skeletons are shown from a perspective that shows the errors best. The single Kinect skeletons will be shown on the left side and the equivalent dual Kinect skeleton will be shown on the right side. Blue skeletons belong to the motion capture suit ground truth while red skeletons belong to the Kinect.

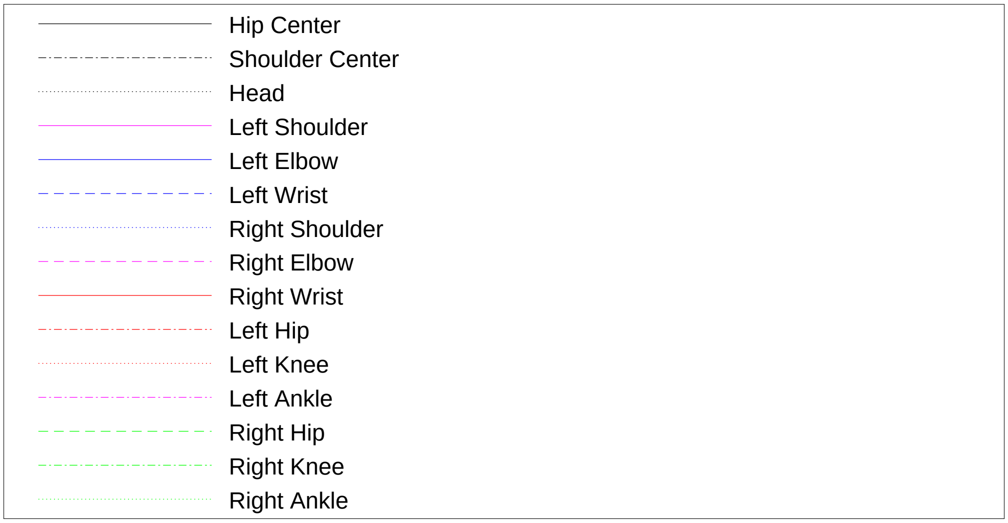


Figure 4.1: **Legend for all tracking graphs.** Hands, feet, and the center spine are not included in any of the graphs due to the fact that the motion capture suit places those joint too far away from the Kinect relative to their limb lengths causing them to give false errors.

In the graph for bar curl exercise shown in figure 4.2, there is not any mis-tracking due to occlusion. In our tests, it has been observed that movements in the direction towards the camera are not tracked very well and slightly delayed. The spikes in distance for this specific exercise correspond to when the subject starts raising and lowering the bar as shown in 4.3. Since there are two repetitions of the

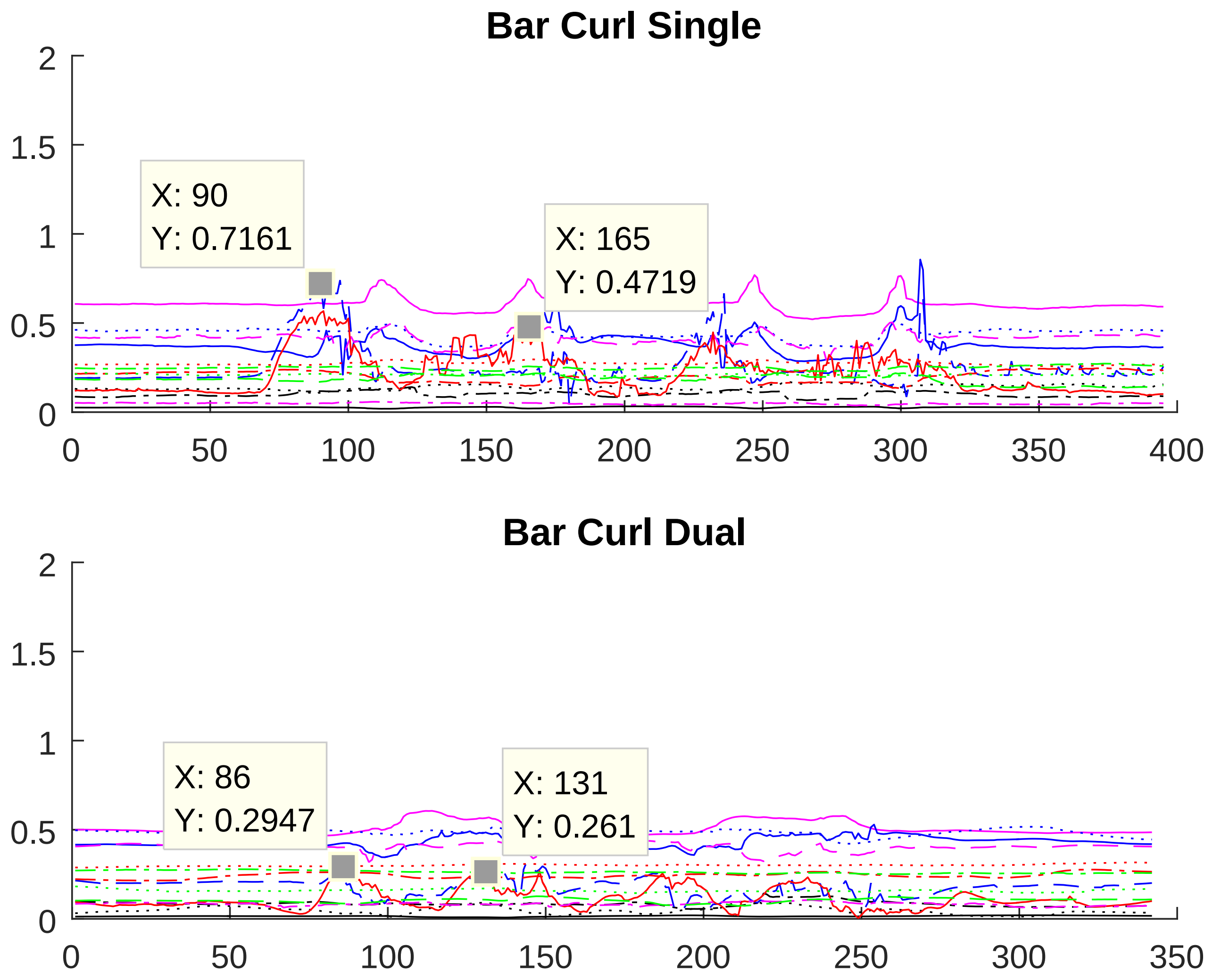
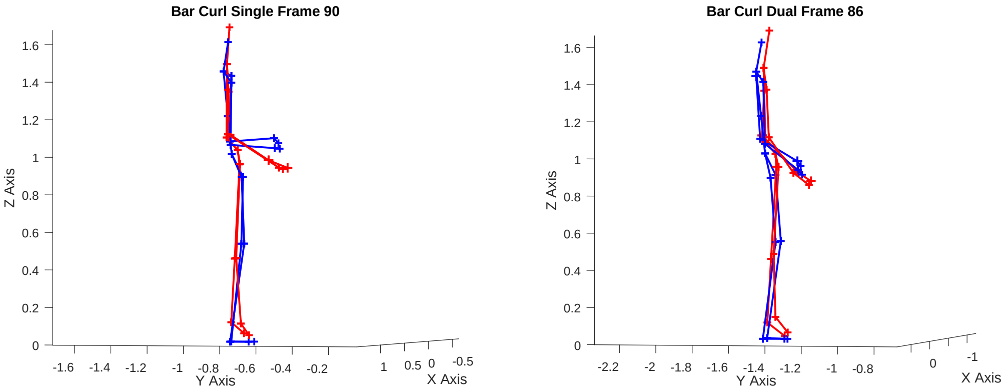
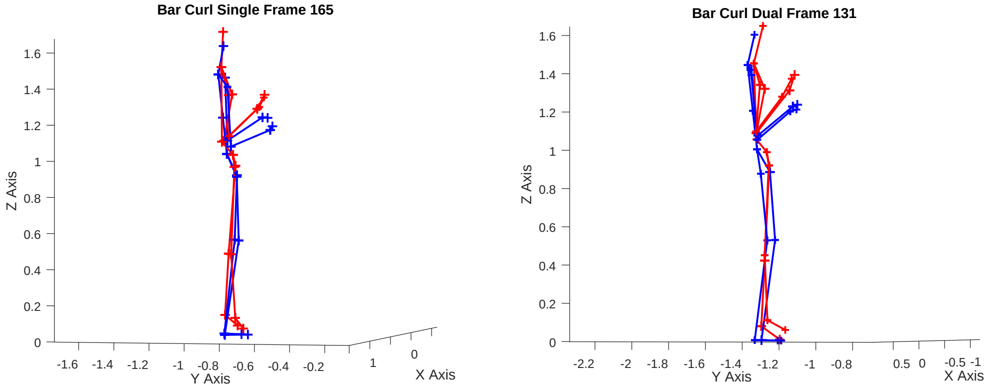


Figure 4.2: **Tracking test for bar curl.** Top graph shows the distances per joint from the single Kinect to the motion capture suit, and the bottom graph shows the dual Kinect



(a) Single Kinect: delay in raising arms

(b) Dual Kinects: delay in raising arms



(c) Single Kinect: delay in lowering arms

(d) Dual Kinects: delay in lowering arms

Figure 4.3: **Peak frames for bar curl.** Kinects are in red and the motion capture suit is in blue.

bar curl in the recordings, there are four spikes in the graph. Also, since the dual Kinect session have the kinects at a slight angle, the amount of delay is less than in the single kinect session. Since the full system uses DTW to temporally align sequences, delays such as these are not a large concern.

In the graph for the brush knee tai chi exercise shown in figure 4.4, a mistracking error occurs due to occlusion. Towards the end of the exercise, the subject makes a 90° turn clockwise and the single Kinect skeleton's right arm mistracks on to the subject's left arm as shown in figure 4.5. In the dual Kinect session, when the subject does the clockwise turn, the Kinect placed on the right side of the subject can still track the movements of the user and therefore the final voted skeleton still follows the movement of the subject.

In the horse's mane tai chi exercise shown in figure 4.6, two spikes occur in the single Kinect session. As shown in figure 4.7, the first spike occurs when the subject lifts their right hand forward and the single Kinect skeleton is delayed at the start due to the motion towards the camera. The dual Kinect session's delay is slightly less due to the Kinects having an angled perspective. The second spike in the single Kinect session occurs due to occlusion. Towards the end of the exercise, the subject makes a 90° turn counterclockwise and the single Kinect skeleton's left arm mistracks to the subject's right arm. In this situation, the subject's left shoulder is also occluded, but the Kinect reports the last known position instead of snapping onto the right arm. In the dual Kinect session, the Kinect on the left side of the subject can still see the left arm and therefore the final voted skeleton still follows the movement of the subject.

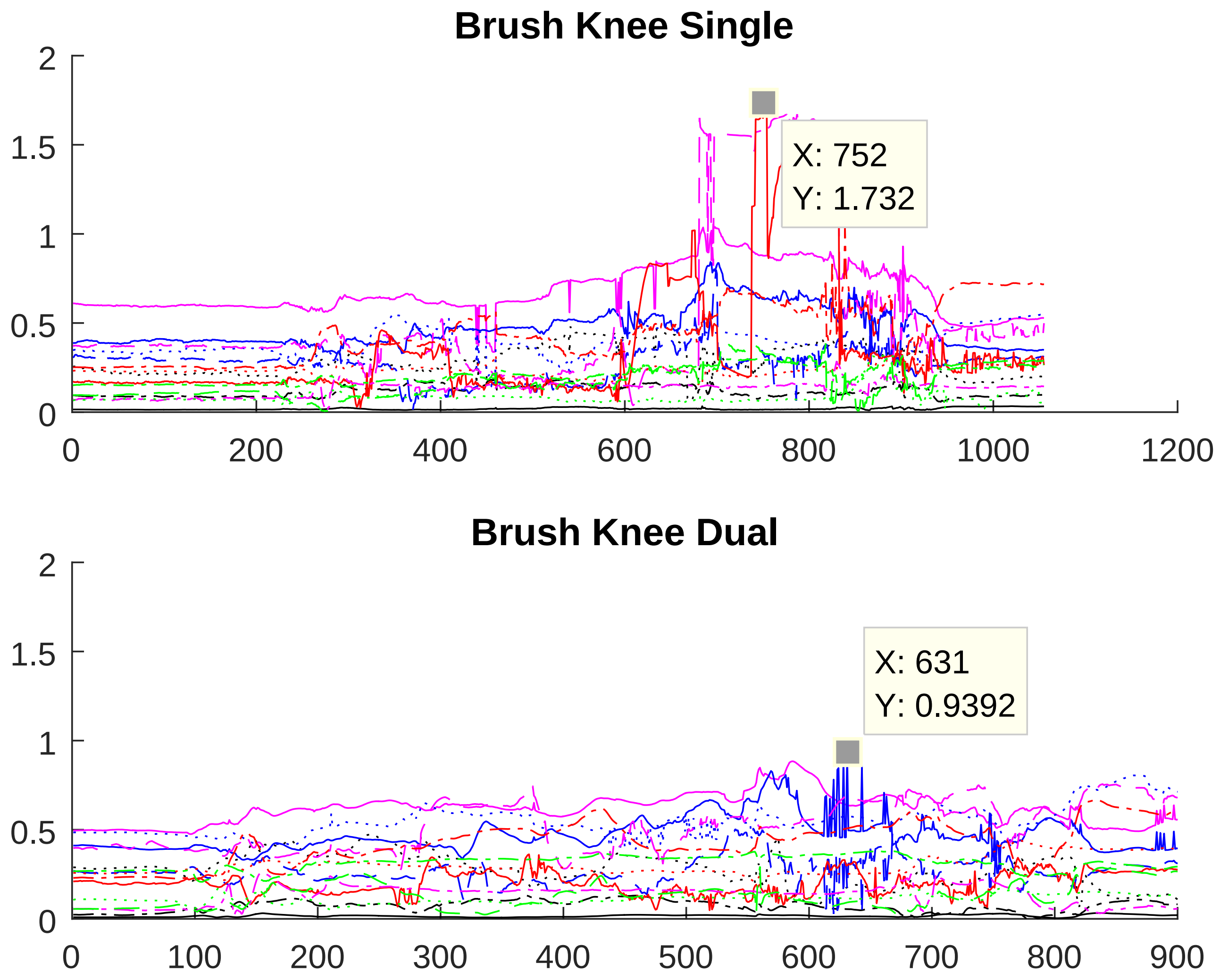
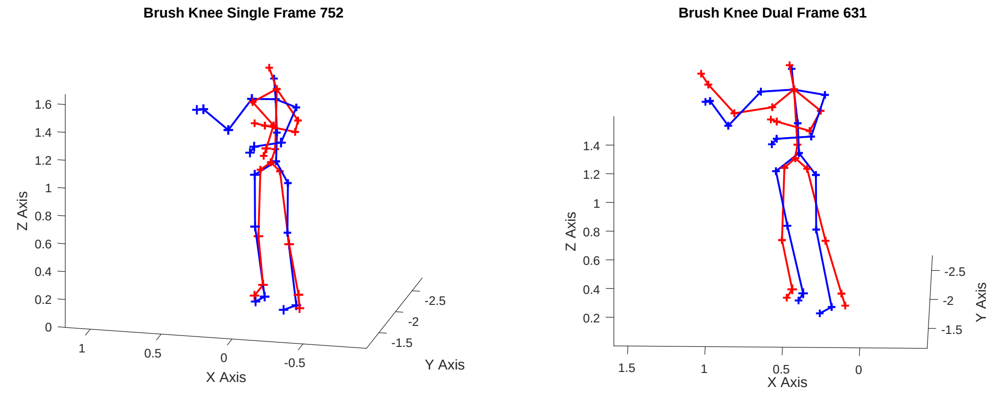


Figure 4.4: **Tracking test for brush knee.** Top graph shows the distances per joint from the single Kinect to the motion capture suit, and the bottom graph shows the dual Kinect



(a) Single Kinect: right arm occluded by left arm

(b) Dual Kinects: no error

Figure 4.5: **Peak frames for brush knee.** Kinects are in red and the motion capture suit is in blue.

In the horse stance exercise shown in figure 4.8, there is little movement in the recording and both sessions tracked the subject successfully.

In the march exercise shown in figure 4.9, there are two types of distance spikes that get repeated as there are multiple repetitions of the march. As shown in figure 4.10 the first spike occurs when the knees get raised. When each leg gets raised, the motion capture suit follows the small swaying the upper body does while the Kinects in both sessions do not. Since the swaying is relative to the hips, the distance for both the left hip and right hip oscillate. The second spike occurs when the subject raise their arms towards the camera and the Kinect skeleton lags.

In the vertical press exercise shown in figure 4.11, the only spikes in distance occur when the subject is raising and dropping arms and the Kinect lags behind the motion capture suit which is shown in figure 4.12.

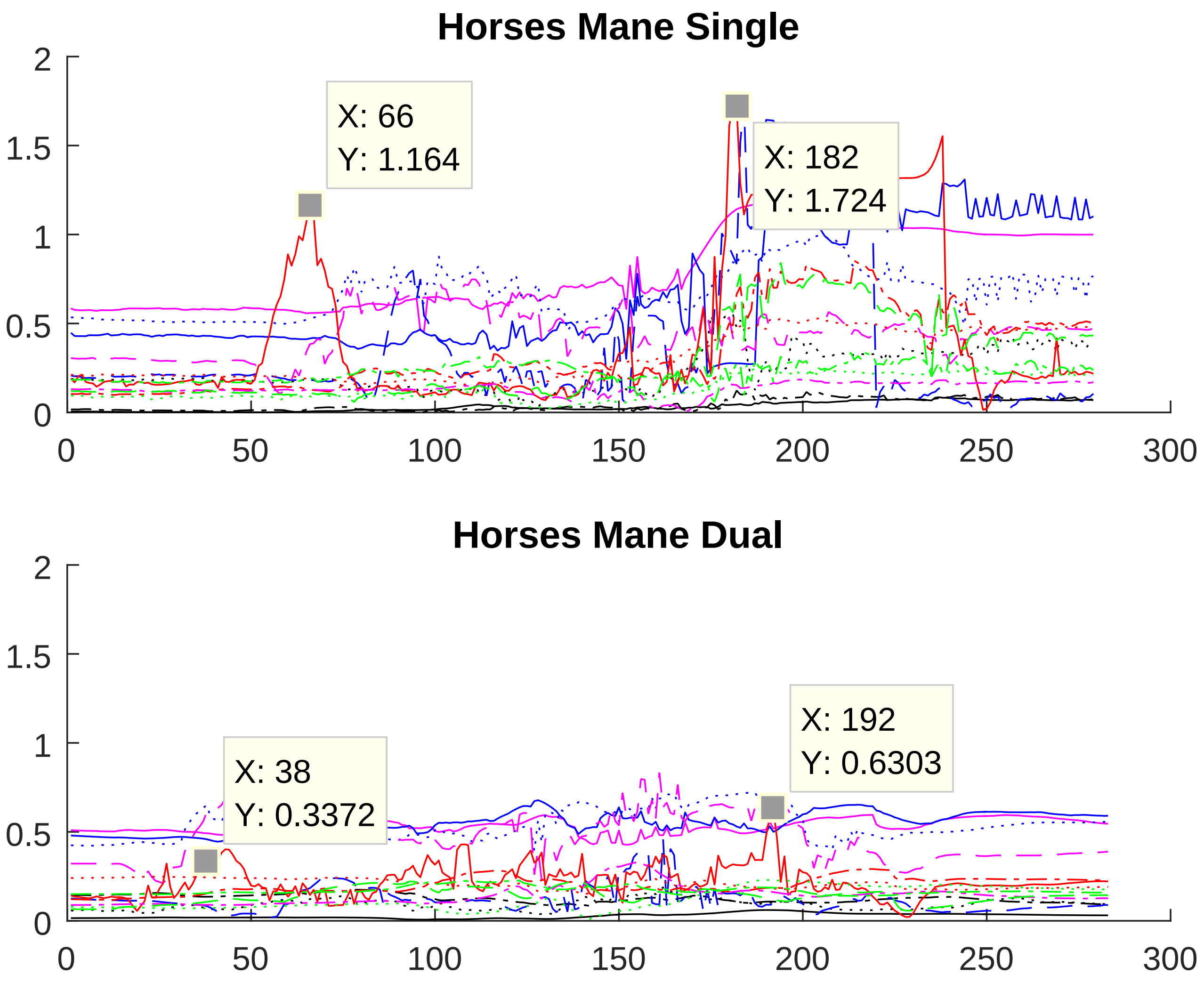
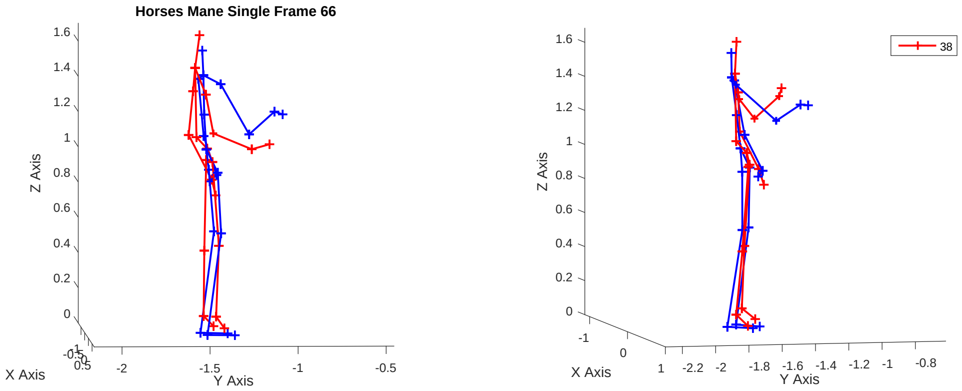
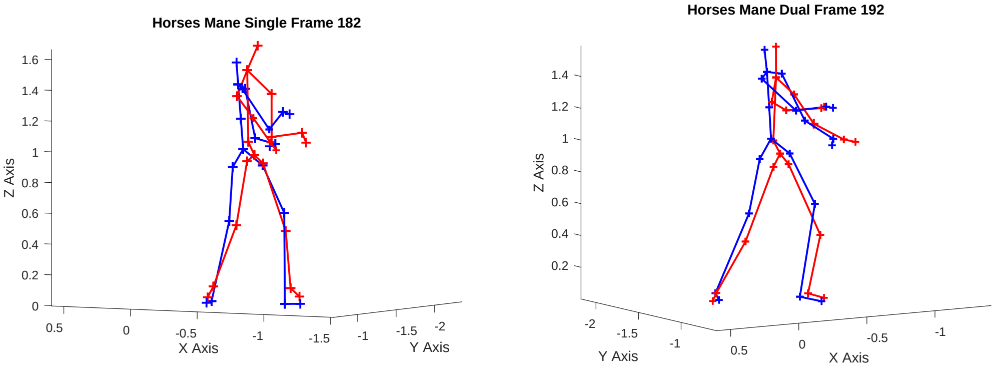


Figure 4.6: **Tracking test for horse's mane.** Top graph shows the distances per joint from the single Kinect to the motion capture suit, and the bottom graph shows the dual Kinect



(a) Single Kinect: delay in raising right arm

(b) Dual Kinects: no errors



(c) Single Kinect: left arm occluded by body

(d) Dual Kinects: no errors

Figure 4.7: **Peak Frames for horse's mane.** Kinects are in red and the motion capture suit is in blue.

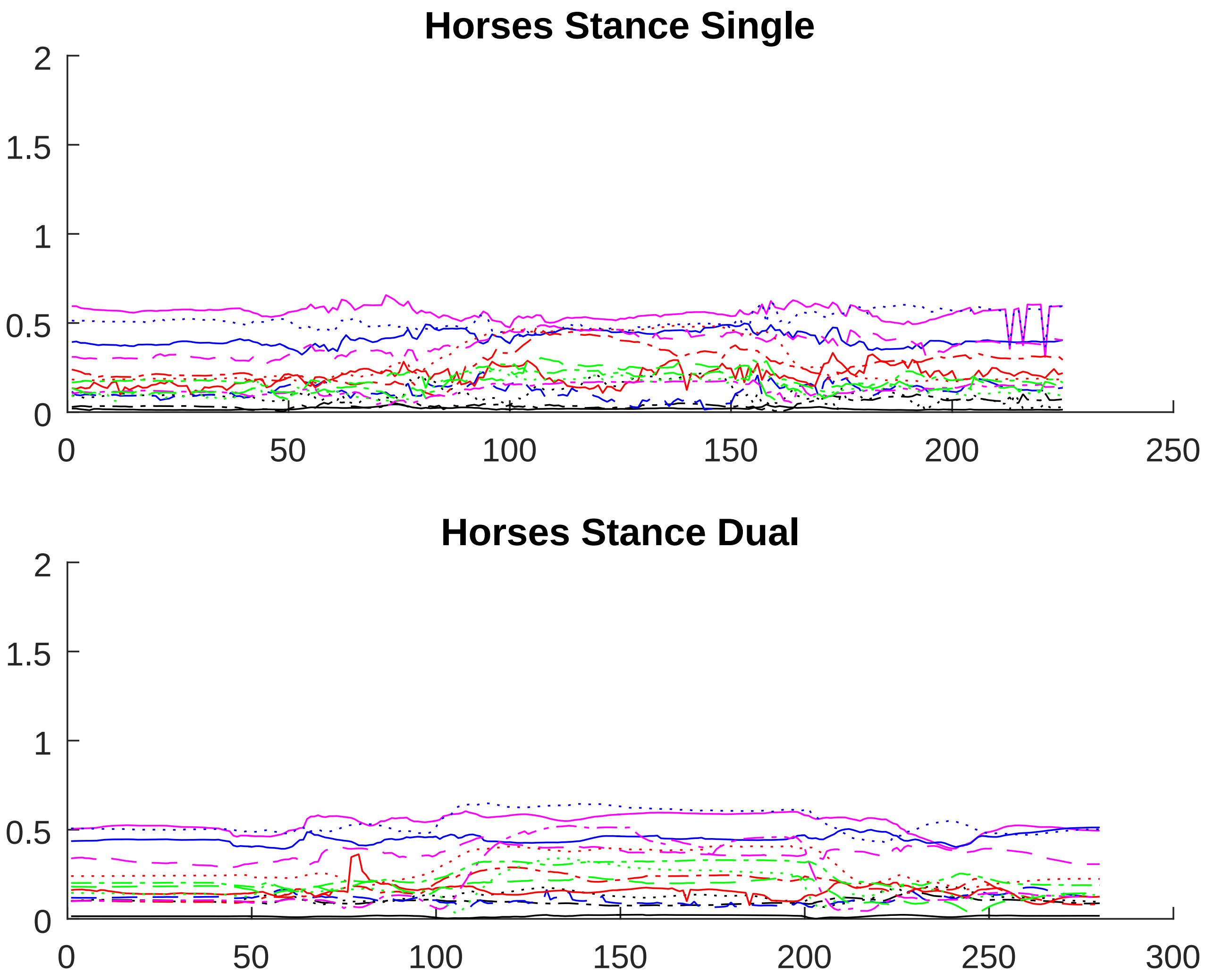


Figure 4.8: **Tracking test for horse stance.** Top graph shows the distances per joint from the single Kinect to the motion capture suit, and the bottom graph shows the dual Kinect

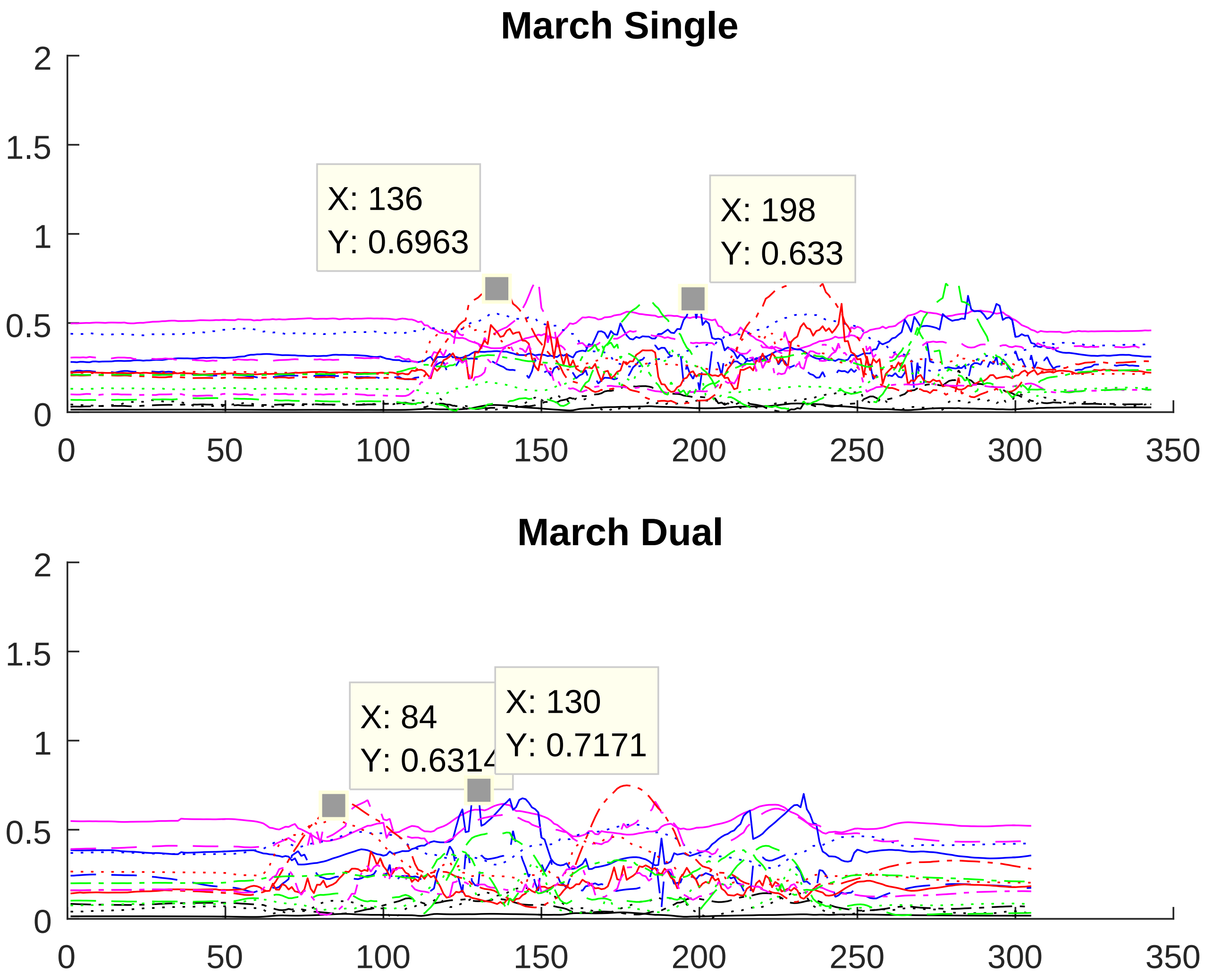
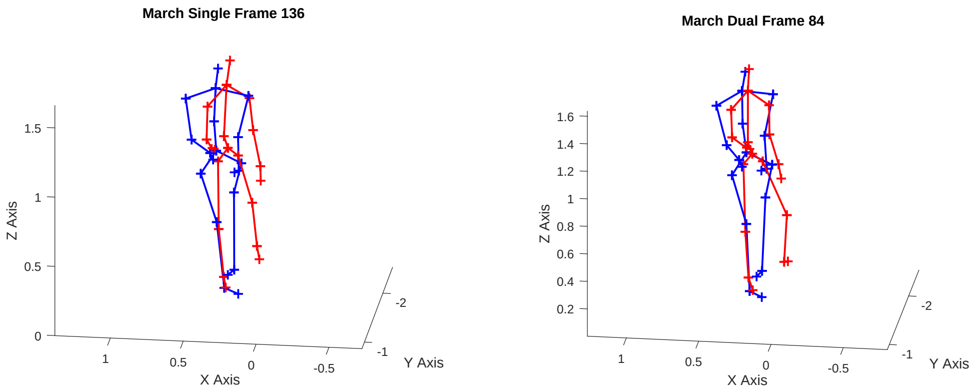
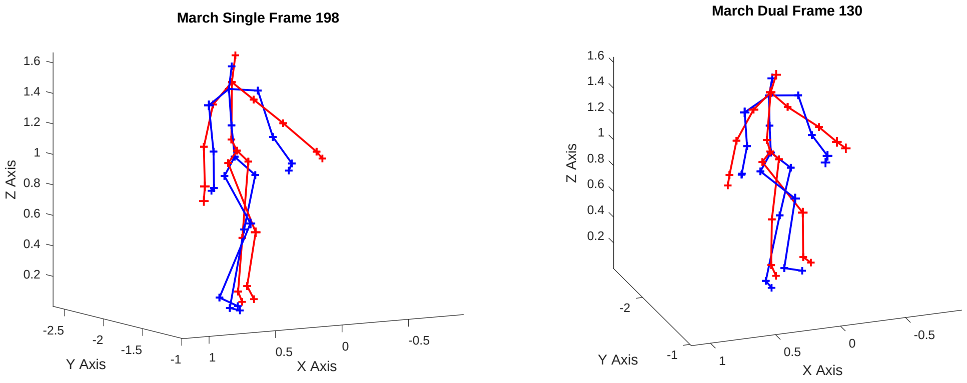


Figure 4.9: **Tracking test for March.** Top graph shows the distances per joint from the single Kinect to the motion capture suit, and the bottom graph shows the dual Kinect



(a) Single Kinect: Kinect does not follow body swaying

(b) Dual Kinect: Kinect does not follow body swaying



(c) Single Kinect: delay in raising arms

(d) Dual Kinects: delay in raising arms

Figure 4.10: **Peak Frames for march.** Kinects are in red and the motion capture suit is in blue.

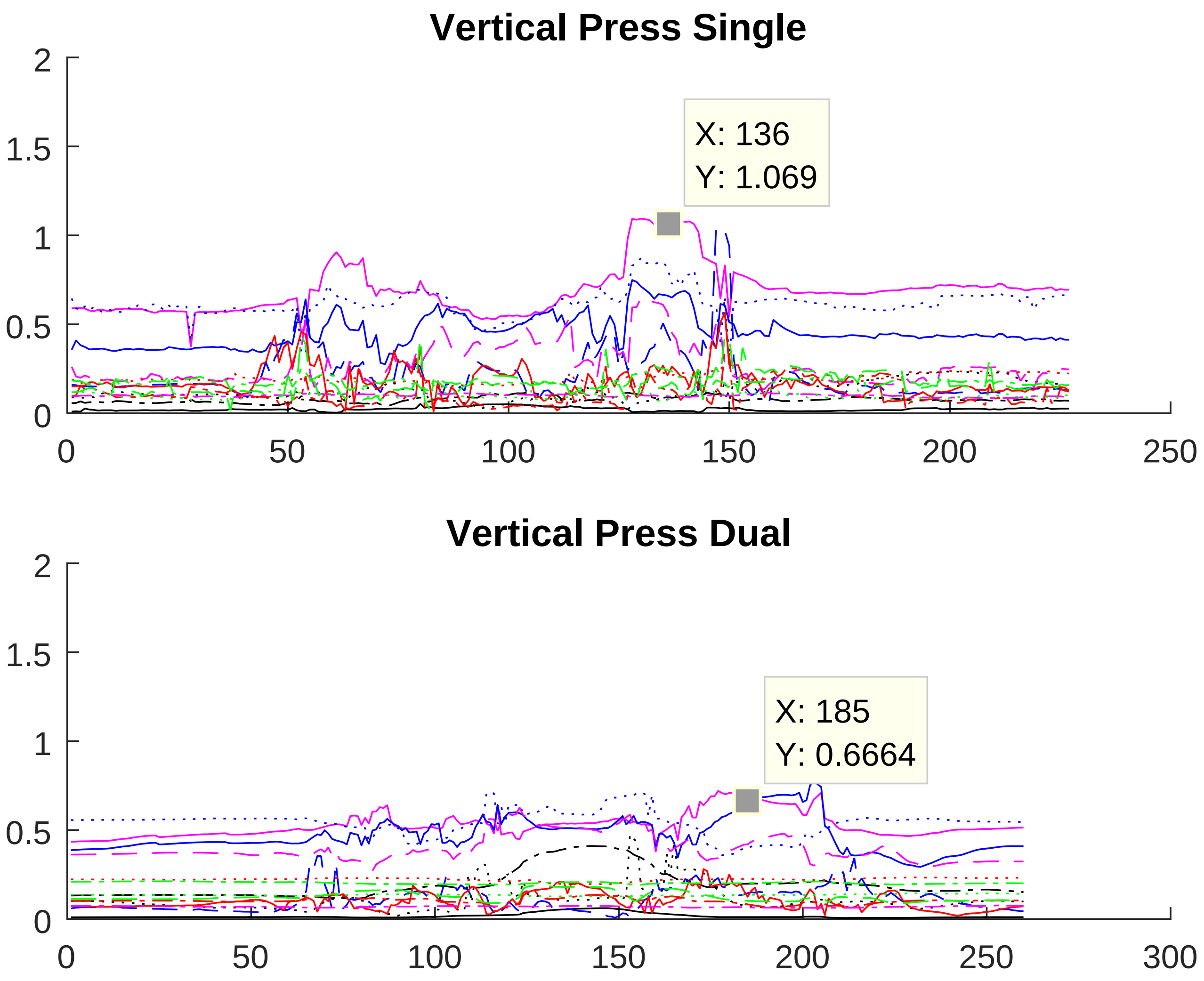
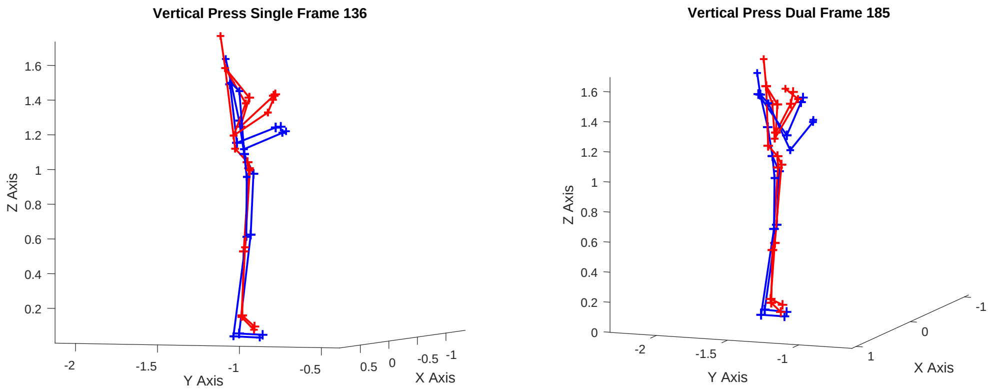


Figure 4.11: **Tracking test for vertical press.** Top graph shows the distances per joint from the single Kinect to the motion capture suit, and the bottom graph shows the dual Kinect


Vertical Press Single Frame 136

Vertical Press Dual Frame 185

(a) Single Kinect: delay in raising arms

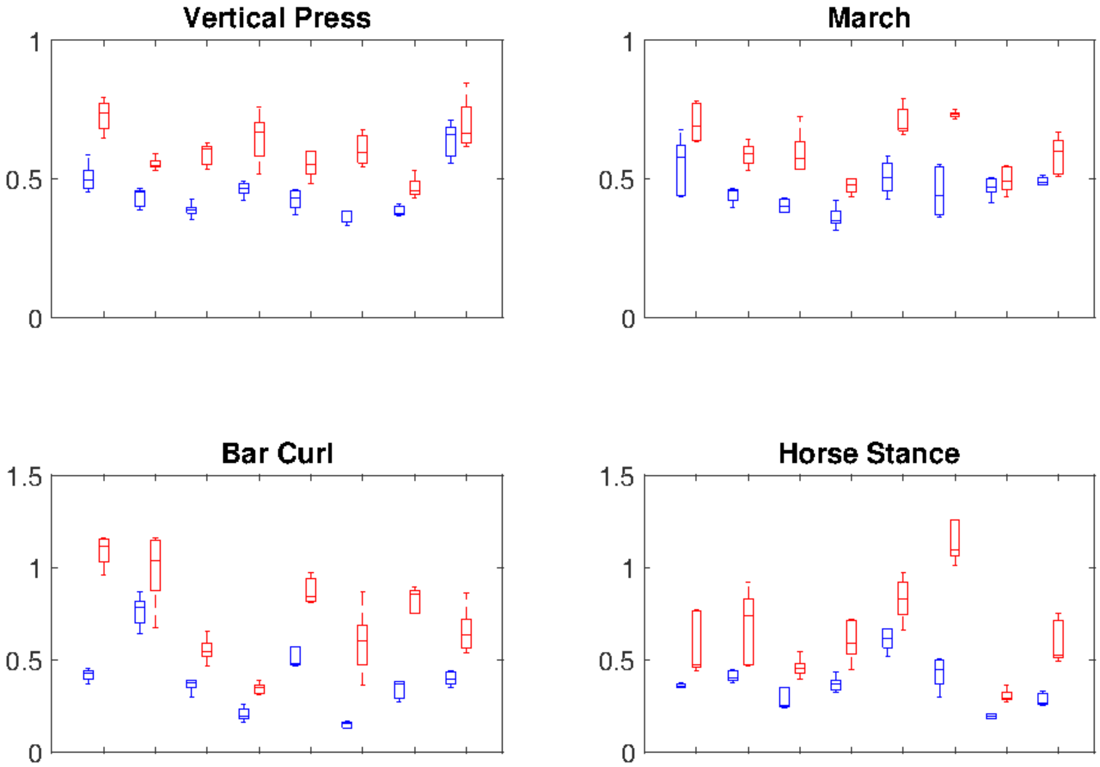
(b) Dual Kinect: delay in raising arms

Figure 4.12: **Peak frames for vertical press.** The peaks occur due to the Kinect reacting slower to the arms being raised.

4.2 Grading Test

The second part of the experiment includes the four simpler exercises. For each of the exercises, the “expert” in the first part of the experiment made the recording to remain consistent. All eight “users” practiced each of the exercises until they were comfortable performing them. They were then asked to perform each of the exercises five times to the best of their ability and then five times with a specific mistake. The exercises and mistakes were: a) vertical press with inclined back, b) march with not bending the legs 90°, c) bar curl with putting the whole arm into motion instead of only using biceps and d) horse stance with not spreading the legs far out enough and compensating by pointing the knees outwards.

The results of the first test in Figure 4.13 show the effectiveness of IDTW scoring. For each individual user, the incorrect performances received higher IDTW costs on average compared to their proper performances. It is also important to


Vertical Press

March

Bar Curl

Horse Stance

Figure 4.13: **Results of the grading test.** Each user's good performances are in blue, and their incorrect performances are in red. The results are shown in a boxplot where the top and bottom markings are the max and min data points while the middle box with the line through it represents the first quartile, median and third quartile.

note that the IDTW costs were fairly dependent on the user and the type of exercise. While it may not be possible to set a universal threshold between good and poor performances, it is possible to easily tune the system per individual and exercise. While all joint lengths were normalized to compensate for users of varying sizes, factors such as different limb length ratios could cause even a standing posture to have slightly different IDTW costs between users. Another observation is that since the IDTW costs are time averaged, the initial coloring of the skeleton is heavily affected by minor changes in posture between the user and expert but quickly changes as the exercise actually starts.

4.3 User Study

The objective of the third part of the experiment is to quantify a user's performance over consecutive sessions with/without the colored skeleton visualization while following a difficult routine alongside an expert. The two tai chi exercises were used in this part of the experiment. Tai Chi was chosen as it was complex enough to not get correct on the first try while not being strenuous to perform. The *experts* that recorded the ground truth were the same as in the first part of the experiment to remain consistent. In order for the users to learn the exercises quicker, the expert recordings break the exercises up into individual steps with pauses in between. The eight "users" are divided into two groups: the first group performs Horse's Mane with the visual feedback system enabled while performing Brush Knee with the visual feedback system disabled and vice versa for the second group. Each user performed the same exercise for ten sessions. The users' cumulative DTW costs were recorded at the end of each session.

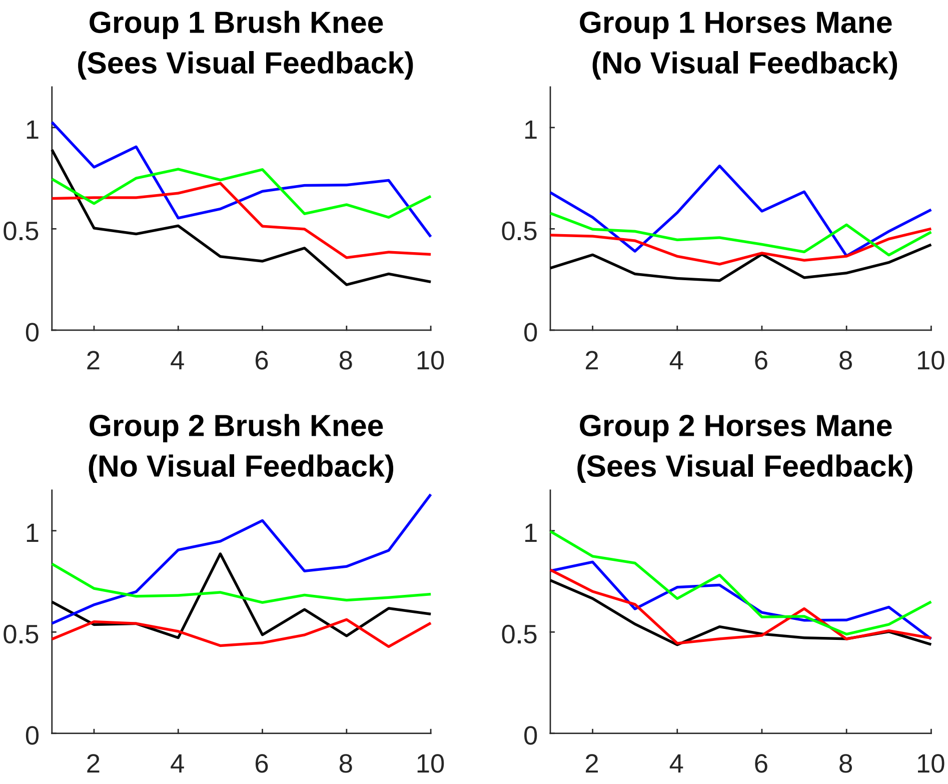


Figure 4.14: **Results of the user study.** The graphs shows each group's individual DTW costs over 10 sessions for each of the two exercises. Each line represents an individual user and group 1 has different people from those in group 2.

As seen in Figure 4.14, when each group saw their feedback, their graphs showed a downward trend in their IDTW costs. When the users had their colored skeletons disabled, their IDTW costs were more erratic overall and some users even got worse over the ten sessions. While it can be argued that certain users performed better than others and that both of the exercises were not equal in difficulty, erratic scores between sessions only occurred when users did not see their visual feedback. These results show that the visual feedback system can indeed help a user quickly improve over time with easy-to-interpret feedback, which isn't possible with simply imitating a video.

Chapter 5

Conclusion

5.1 Summary of Thesis

In this thesis, an improved real-time human activity analysis system is proposed. Many previous works state how skeletal tracking with Kinects are unreliable as occlusion can cause mistracking. The system improves its tracking through the use of dual Kinect voting. IDTW is used to calculate the distance cost between a partial user sequence and a complete expert sequence. Visual feedback allows users to easily understand where they can improve their performance in real-time. The experiments conducted in this thesis show that the system has potential benefits in expert-guided activities. The system itself can be split into two parts; the dual Kinect algorithm and the grading and visual feedback system.

The dual Kinect algorithm leverages the use of multiple viewpoints to alleviate the mistracking due to occlusions. The usage of Kinects and other RGB-D devices in activity analysis benefit from easier segmentation and classification of body segments which is not easily achieved by regular RGB methods while not being expensive and invasive as motion capture suits. While the depth frames give much more information that color frames, occlusion is still a huge problem for any camera-based method. Many methods exist that try to correct Microsoft's skeleton acquisition when it mistracks. While there are many methods that completely rebuild skeletons using depth frames from multiple angles that have high tracking accuracy including commercial products, these methods tend to have high processing times that prevent their use in real-time applications. The dual Kinect algorithm by [7] only targets the prevention of mistracking due to occlusion but has quick processing time due to only using skeletal data. While the method uses two Kinects, it is possible to further extend it to use more. The method uses SVD to find the RBT between all the Kinects. Once the transforms have been found, all Kinects transform their skeletons to the same coordinate system. SQP is used to optimize skeletal joint positions using the constraint that a user's limbs must remain the same size while the direction is weighted by how correct each reported joint is. The weights are determined by if the Kinect reports the joint as being tracked and if the joint is close to another one (which is a sign of possible self-occlusion).

The grading and visual feedback system was designed to give users intuitive feedback in real time. The use of IDTW as seen in [11, 12] allows a completed expert sequence to be matched to an incomplete user sequence in real time. The DTW algorithm can find optimal temporal alignments for sequences that are not rate dependent. As most calculations are redundant with previous frames, IDTW saves computation time by saving all previous calculations and only calculating the path for the most current frame. In order to match sequences where the subject has different body sizes, the joint features have been converted to a unit vector relative to its parent joint. Also, as the system is targeted for at-home use, the system counter-rotates each skeleton with the left to right hip vector from the calibration stage to compensate for different Kinect positions between uses. Since numeric scores per joint that refresh per frame would be too complex for users to comprehend, an intuitive feedback system is used. The scores per pair of joints are averaged to get scores per limb and those scores are mapped to a color scale. The colors are overlaid on the user's replay as a colored skeleton in real time.

In our first part of the experiments, all exercises used in the later parts of the experiment are compared to a motion capture suit to ensure they do not mistrack due to occlusion. When compared to a single front-facing Kinect, the dual Kinect algorithm manages to at least follow all of the motions while the single Kinect consistently mistracked. It is observed that the dual Kinect algorithm increases the range of natural movements that can be tracked. In the second part of the experiment, it is demonstrated that for four different exercises, the system can differentiate between good and poor performances and can localize the source of error. Through two Tai-Chi exercises and splitting the users into two groups with/without the visual feedback system, it is observed that the group with the visual feedback can improve their performance noticeably over 10 sessions, whereas the performance of group without any visual feedback is erratic. This low-cost system can give users a better learning experience compared to simply imitating a video.

5.2 Future Works

This thesis targeted a specification for satisfactory accuracy to give meaningful feedback to users while targeting real-time performance and intuitive human computer interaction. Several proposed future works could improve the accuracy of the system.

The first possible change to the system would be to add more conditions to the weighting in the SQP. The current conditions include if the Kinect is reporting the joint as tracked and also the distance to the closest non-connected joint. The second condition is based off the assumption that if a joint is occluded but still reported as tracked, the joint most probably snapped onto the part of the body that occluded the joint. Unfortunately, the unintended consequence of is that if any body joints are actually close to other body parts (like if you were clapping or crossing your arms), the correct joint position would be given less weight.

Another option would be to change the multiple Kinect algorithm completely. [10] that was published after our system was implemented works fairly similar to the current algorithm in that the final joint position is a weighted average of all reported positions based off the confidence on their accuracy. One advantage that the Kalman filter has is that it incorporates temporal information into the weighting while the current method only deals with single frame information.

Another possible future work would be for the system to adapt the UI improvements of [12]. In their work they suggest the correct joint position if the user's score drops below a certain threshold. While our system can quantify if a user is improving or getting worse, the system currently does not have a way of showing the user where the correct position is. Also, in their work they give higher weights to specific postures within the routine. Currently, the DTW score is time averaged over the entire user recording. By weighting certain frames higher, the system can ensure the certain postures within the activities are met. One example would be the vertical press where we would want to ensure the user extends their arms straight above their heads before dropping their arms. Unfortunately, since our method is meant for general applications and not a specific activity, we can not use a SSOM to learn the postures.

The features being used in the DTW algorithm can also be changed. Currently, the joint position relative to their parent joint is the feature being used. Many other works leverage motion features and joint angle features [23, 30]. While exploring the different combinations of features, it may be possible to find complementary interactions between them. While using more features may potentially increase the accuracy of the evaluation, it is important to prioritize the computational efficiency so that it still runs in real-time.

Bibliography

[1] S. Hagler, H. B. Jimison, R. Bajcsy, and M. Pavel, "Quantification of human movement for assessment in automated exercise coaching," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 5836–5839.

[2] A. Yurtman and B. Barshan, "Detection and Evaluation of Physical Therapy Exercises by Dynamic Time Warping Using Wearable Motion Sensor Units," in *Information Sciences and Systems 2013*, ser. Lecture Notes in Electrical Engineering. Springer, Cham, 2013, pp. 305–314, dOI: 10.1007/978-3-319-01604-7_30. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-01604-7_30

[3] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–126, Nov. 2006. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1077314206001263

[4] Matthew Field, Zengxi Pan, David Stirling, and Fazel Naghdy, "Human motion capture sensors and analysis in robotics," *Industrial Robot: An International Journal*, vol. 38, no. 2, pp. 163–171, Mar. 2011. [Online]. Available: http://www.emeraldinsight.com.ezproxy.lib.ryerson.ca/doi/full/10.1108/01439911111106372

[5] G. Shi, Y. Wang, and S. Li, "Human Motion Capture System and its Sensor Analysis," *Sensors & Transducers; Toronto*, vol. 172, no. 6, pp. 206–212, Jun. 2014. [Online]. Available: https://search-proquest-com.ezproxy.lib.ryerson.ca/docview/1545208935/abstract/DF472229C0A4403FPQ/1

[6] Z. Cai, J. Han, L. Liu, and L. Shao, "RGB-D datasets using microsoft kinect or similar sensors: a survey," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4313–4355, Feb. 2017. [Online]. Available: https://link-springer-com.ezproxy.lib.ryerson.ca/article/10.1007/s11042-016-3374-6

[7] K.-Y. Yeung, T.-H. Kwok, and C. C. Wang, "Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications," *Journal of Computing and Information Science in Engineering*, vol. 13, no. 4, p. 041007, 2013.

[8] Z. Gao, Y. Yu, Y. Zhou, and S. Du, "Leveraging Two Kinect Sensors for Accurate Full-Body Motion Capture," *Sensors*, vol. 15, no. 9, pp. 24 297–24 317, Sep. 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/9/24297/

[9] K. Kaewplee, N. Khamsemanan, and C. Nattee, “A rule-based approach for improving Kinect Skeletal Tracking system with an application on standard Muay Thai maneuvers,” in *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*. IEEE, 2014, pp. 281–285. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7044763

[10] S. Moon, Y. Park, D. W. Ko, and I. H. Suh, “Multiple Kinect Sensor Fusion for Human Skeleton Tracking Using Kalman Filtering,” *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 65, Mar. 2016. [Online]. Available: http://journals.sagepub.com/doi/10.5772/62415

[11] N. Khan, S. Lin, L. Guan, and B. Guo, “A Visual Evaluation Framework for In-Home Physical Rehabilitation,” in *2014 IEEE International Symposium on Multimedia (ISM)*, Dec. 2014, pp. 237–240.

[12] P. Muneesawang, N. M. Khan, M. Kyan, R. B. Elder, N. Dong, G. Sun, H. Li, L. Zhong, and L. Guan, “A Machine Intelligence Approach to Virtual Ballet Training,” *IEEE MultiMedia*, vol. 22, no. 4, pp. 80–92, Oct. 2015.

[13] J. Aggarwal and M. Ryoo, “Human activity analysis: A review,” *ACM Computing Surveys*, vol. 43, no. 3, pp. 1–43, Apr. 2011. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1922649.1922653

[14] T. Pfister, “Advancing human pose and gesture recognition,” dissertation, University of Oxford, 2015. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/publications/2015/Pfister15/pfister15.pdf

[15] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005. [Online]. Available: http://www.springerlink.com/index/m441rlt8j0063k7k.pdf

[16] I. Laptev and T. Lindeberg, “Space-time interest points,” in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct. 2003, pp. 432–439 vol.1.

[17] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 3, pp. 257–267, 2001. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=910878

[18] J. Koenemann, F. Burget, and M. Bennewitz, “Real-time imitation of human whole-body motions by humanoids,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2806–2812.

[19] M. Ye, C. Yang, V. Stankovic, L. Stankovic, and A. Kerr, “A Depth Camera Motion Analysis Framework for Tele-rehabilitation: Motion Capture and Person-Centric Kinematics Analysis,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 5, pp. 877–887, Aug. 2016.

[20] 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," in *IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1297–1304.

[21] H. Yanai, K. Takeuchi, and Y. Takane, "Singular Value Decomposition (SVD)," in *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*, ser. Statistics for Social and Behavioral Sciences. Springer New York, 2011, pp. 125–149, dOI: 10.1007/978-1-4419-9887-3_5.

[22] M. S. Gockenbach, "Introduction to sequential quadratic programming," *Course material, Michigan Technological University*, vol. 106, 2003. [Online]. Available: http://www.math.mtu.edu/~msgocken/ma5630spring2003/lectures/sqp1/sqp1.pdf

[23] X. Yang and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 14–19. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6239232

[24] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer Vision and*

Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. IEEE, 2012, pp. 20–27. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6239233

[25] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[26] Y. L. Hsu, C. L. Chu, Y. J. Tsai, and J. S. Wang, "An Inertial Pen With Dynamic Time Warping Recognizer for Handwriting and Gesture Recognition," *IEEE Sensors Journal*, vol. 15, no. 1, pp. 154–163, Jan. 2015.

[27] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping," *ACM Trans. Knowl. Discov. Data*, vol. 7, no. 3, pp. 10:1–10:31, Sep. 2013. [Online]. Available: http://doi.acm.org/10.1145/2500489

[28] N. Begum, L. Ulanova, H. A. Dau, J. Wang, and E. Keogh, "A General Framework for Density Based Time Series Clustering Exploiting a Novel Admissible Pruning Strategy," *arXiv preprint arXiv:1612.00637*, 2016. [Online]. Available: https://arxiv.org/abs/1612.00637

[29] J. F. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders, "Sign Language Recognition by Combining Statistical DTW and Independent Classifica-

tion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2040–2046, Nov. 2008.

[30] S. Nowozin and J. Shotton, "Action points: A representation for low-latency online human action recognition," 7 J J Thomson Ave, CB30FB Cambridge, UK, Tech. Rep., July 2012. [Online]. Available: https://www.microsoft.com/en-us/research/publication/action-points-a-representation-for-low-latency-online-human-action-recognition/

[31] A. Kondyli, V. P. Sisiopiku, L. Zhao, and A. Barmpoutis, "Computer Assisted Analysis of Drivers' Body Activity Using a Range Camera," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 18–28, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7166427

[32] L. Zhang, J.-C. Hsieh, T.-T. Ting, Y.-C. Huang, Y.-C. Ho, and L.-K. Ku, "A kinect based golf swing score and grade system using gmm and svm," in *Image and Signal Processing (CISP), 2012 5th International Congress on*. IEEE, 2012, pp. 711–715. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6469827

[33] C. L. V. Lisboa, L. Nedel, and A. Maciel, "A Study for Postural Evaluation and Movement Analysis of Individuals," in *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, Jun. 2016, pp. 122–126.

[34] C. D. Wickens and J. G. Hollands, *Engineering psychology and human performance*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2000.

[35] D. W. Eggert, “Estimating 3-D rigid body transformations: a comparison of four major algorithms,” *Machine Vision and Applications*, vol. 9, no. 5/06, pp. 272–290, 1997.

[36] W. Shen, K. Deng, T. Leyvand, B. Guo, and Z. Tu, “Exemplar-Based Human Action Pose Correction,” *IEEE Transactions on Cybernetics*, vol. PP, no. 99, 2013.