MOVEMENT CLASSIFICATION AND ANALYSIS
FROM RGB – D VIDEO DATA

By

Patrick D. Ring, B.S.

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Computer Science
May 2019

Committee Members:

Vangelis Metsis, Chair

Jelena Tesic

Yijuan Lu

Yan Yan

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGEMENTS

This Thesis bears my name but in reality this is the culmination of time and effort of many individuals to whom I will be forever grateful. Without these people in my life I would not be where I am today. They aided and inspired me and I can only hope to do the same for others in the future. I enjoyed doing this research and providing value through this work but the real joy and value comes from those I have gotten to work with along the way.

First and foremost I would like to thank Dr. Vangelis Metsis for being my Thesis advisor and guiding through this work. Dr. Metsis is a great teacher and knows how to make learning fun and exciting. I want to thank Priyank Trivedi for providing me with many of the tools that I could use expand upon and just generally being great company to bounce ideas off of. Both Priyank and Dr. Metsis were integral to the foundation of my research and I very much enjoyed working with them. I'd like to thank all of those who volunteered to undergo data collection. I won't mention them by name as some preferred to remain anonymous but their contributions are invaluable not only for me but for future researchers.

I'd like to thank Dr. Tesic, Dr. Lu, and Dr. Yan for agreeing to join my committee. Thank you to all of my professors during my time at Texas State. Each one has given my

not only knowledge but also wisdom that I will carry with me forward. I hope you all have enjoyed having me as a student as much as I have enjoyed learning from you.

Most of all I would like to thank my wife LesLeigh Ring. She encouraged me to pursue a Master’s degree and has been my love and support through these past years. I live a truly amazing life and it is all because of her.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF EXAMPLES

Example **Page**

ABSTRACT

The aim of this thesis is to develop and evaluate methods of human movement classification using motion tracking data captured using a RGB-D sensor. As hardware solutions evolve and improve, so too must the software solutions evolve to creatively leverage and combine new technologies. Accurate human movement classification can facilitate a variety of practical applications, ranging from the health domain to sports, film, and even advanced surveillance. In this work, we focus on human movements related to physical therapy exercises. Motion tracking data was collected from subjects performing various physical exercises. The goal of our system is to automatically recognize the types of exercises performed by teach subject and the number of repetitions of each particular exercise. To achieve this goal, we use 3D skeleton tracking data points provided by the Microsoft Kinect sensor. After a set of transformation steps, we apply a Long Term Short Term Memory (LSTM) Deep Learning networks in tandem with the Dynamic Time Warping sequence matching algorithm to classify the types of exercises and number of repetitions performed by each subject.

1. INTRODUCTION

Human movement classification and analysis remains a challenging task for which there are many uses especially in the medical field. As with many other problems in computer vision, a task that is very simple for a human to perform can be exceedingly difficult for a machine.

In this thesis, I gather data of subjects performing exercises using a Microsoft Kinect. The data is classified and analyze using two distinct techniques. One is a common signal processing strategy used for comparison called dynamic time warping (DTW). The other is a deep learning classification strategy that uses a long short-term memory (LSTM) network.

The purpose of the analysis is to compare and contrast the two strategies and assess their usefulness in terms of applications in physical therapy. Each algorithm may excel in different ways to be used in different applications. Speed, accuracy, resource efficiency, and system requirements all play a role in fitting into a particular scenario.

Tracking and analyzing movements of patients is a difficult task even for an experienced physician. The goal of this research is to explore and compare effective methods for classifying and evaluating physical movements using sensor technologies and software analysis. Currently much of the assessment is made visually by a medical professional or reported verbally from the patient themselves. This process is subjective and can be inconsistent from patient to patient or therapist to therapist.

1.1 Motivation

Physical therapy in practice has a number of challenges that are increasingly being remedied through the use of technology. In a physical therapy setting patients are typically tasked with repeating certain movements or exercises in order to progress with recovery or adaptation. There exist a number of standards for movements though there is not one universally agreed upon set. Even within the same set the progress is evaluated by a physician or medical worker which leads to highly subjective interpretation. In cases such as these it is useful to have an automated system with which to compare data between sets and users. Automated systems open up the potential for scalability by allowing for tasks that require trained medical professionals to be performed by untrained individuals assisted by smart tools. Many physical therapy patients will have limited mobility would benefit greatly from digital systems that allow them to get advanced medical care without having to leave their homes. New and improved tools continue to be explored to give patients better diagnoses and to improve the treatment options.

1.2 Challenges

There is an extensive array of tools that can potentially aid in movement analysis. Many of these such as motion capture require large equipment that users wear obtrusive gear in order to capture the data. They can also require the need of specialized technicians to operate and process. The more that these procedures require specific technologies and advanced operational expertise, the more limited these resources will be. The more simplified the system is the more it can be made available. These requirements have encouraged us to explore the Kinect. The Kinect is small, lightweight, portable, can be connected to almost any computer. It is simple to operate and allows the potential for remote capture and analysis. The budget requirements for a Kinect are far more affordable than a full scale motion capture lab without sacrificing much in the level of detail [21].

The Kinect itself does have some challenges of its own when it comes to collecting data. The data gathered comes from a camera and IR array that is then interpreted to a 3D space. The subject is not viewed from all angles simultaneously so there are some scenarios where the joints of the subject are estimated indirectly. These include when joints of the subject are blocked from view and when joints leave cross the boundaries of the Kinect field of view and come back. Certain variables will lower the accuracy of these estimations such as the depth of the subject not varying greatly from their surroundings or having vaguely human shaped objects in the field of view. For the sake of these experiments we asked our subjects not to stand right up against a wall. We do however have some odd objects have in the field of view and for one exercise have subjects interacting with a chair. The last major drawback to using the Kinect is the requirement of the local machine. For recording detailed images at a maximum frame rate does require a modern cpu. Recording only skeleton data does not have this limitation and can be done with a minimal setup.

1.3 Applications

3D motion capture technologies have been around for decades now and are continually advancing in their capabilities. Many healthcare facilities that could benefit from having systems like these use outdated computer systems in general. Not only are budgetary constraints a limiting factor in this but healthcare standards require rigorous safety testing to be approved. The Kinect is readily available and affordable for individual patients. It is already a widely adopted technology with low potential risk and high potential reward for use in medicine.

The Kinect is also more advantageous to the patients. Not requiring the patient to put on any specialized gear or clothing is a major positive to those with limited mobility which the target users are more likely to be. As is the overall size and weight of the system. The hardware is only 3 pounds, easy to install, and does not require any bulky equipment to be worn.

I collected data from ten subjects performing exercises in multiple varied sequences for analysis. The data collection application I developed and optimized for recording multiple data streams through the Kinect. It is written in C# and leverages Microsoft's libraries and Kinect services. The application is simple enough to use that an operator does not require much if any training to record data. It is a Windows application with minimal system requirements. It should run on most modern desktops and laptops.

Once the data is collected there is some extensive preprocessing that is done in order to make the sequences more amenable to analysis. Not least of which is normalization of the skeleton data. This data gives the most accurate representation of the movements and is much easier to analyze from the machine's perspective. I have built on and improved existing skeleton normalization methods. All of this preprocessing is done in matlab. The data can all be represented in the form of matrices making matlab algorithms optimized exactly for manipulating this data.

The data analysis is also done in matlab for many of the same advantages. There are two different algorithms that were chosen to classify the exercises. The dynamic time warping (DTW) algorithm is commonly used to compare signals that are equivalent to each other but take different amounts of time to. For example you may have a song that you want to compare to another song that is identical except played at double speed. A straight forward comparison of the audio signals will not give a positive match because the signals do not line up very well. DTW allows the signal to be aligned along the same time scale giving a more accurate match for comparison. In the same way we can compare people performing exercises even if they do not perform them at the same speed or cadence. DTW is a dynamic programming technique that requires comparing each point in time of one sequence M to each point in time in a second sequence N. This makes the time to run just one comparison *O(NxM).* One positive aspect of using DTW is that it gives a distinct distance measure between the two exercises compared. So not only do we know the classification but also how closely that classification matches the standard. We can use that distance as a measure of how well the subject performed the exercise. It could be used to create a standard that is transferable across different medical offices and different patients.

The second comparison algorithm used is Long Short Term Memory (LSTM). This utilizes a neural network that is trained on a set of exercises that are already classified. This training assigned weights to hidden variables that are used to later classify new sequences. Each round of training refines the hidden weights. We want to define a large number of hidden variables since we know that there are many possible dependencies between each step in an exercise. This process is useful when we know there are relationships but they are hard to define explicitly. We use a neural network that mimics the brain's power to learn these relationships. The LSTM implementation does not however give the same kind of distance measure that DTW does. LSTM also makes it more challenging to count individual repetitions since it does not automatically detect the start and finish of each exercise. DTW by its very nature gives a count of repetitions. Training can take a long time to reach acceptable levels of accuracy but once the network is trained it can be used to classify other sequences quickly.

Matlab was selected for analysis because of some of the new features developed for their Deep Learning Toolbox. The features optimize hardware utilization for LSTM and streamline the programming process. The examples provided by matlab combined with the examples built in this these will hopefully make programming with neural networks more accessible.

DTW and LSTM have different advantages with each being better for a given task depending on the circumstances. DTW does not scale will with a large number of classes and LSTM has challenges with reliably counting individual exercises. Ultimately a hybrid implementation can take advantage of the best qualities of each. The LSTM can be used to narrow the classification on a particular sequence and DTW used to count repetitions using only one movement.

2. RELATED WORK

The problem of analyzing movements has been an ongoing area of research and there are existing solutions that attempt to do many of the tasks that I am going through in this thesis. In one attempt the data gathering process is equivalent but the normalization and analysis uses mostly rudimentary methods to compare data. They found better accuracy in analyzing just the joint angles rather than their absolute positions. The results of their experiments were not very favorable for precise body movement analysis where these methods excel is in counting exercise repetitions [3]. “Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis” (Pfister, A., West, A. M., Bronner, S., & Noah, J. A. 2014) compared the data collected from the Kinect with data from 3D motion capture technologies and both found that the Kinect is less accurate especially when joints are hidden behind other body parts [2]. However there are multiple suggested methods to work around this obstacle some of which I have used in my implementation.

The obstacles presented by certain Kinect inaccuracies has encouraged research to compensate. One difficulty with comparing the 3D motion of two different people is that they never perfectly line up in the 3D space. Subjects are different distances from the Kinect sensors and at different angles in relation to the sensor. To account for misaligned subjects I make extensive use of the skeleton normalization presented by “Kinect skeleton coordinate calibration for remote physical training” (Wei, T & Qiao, Y & Lee, B. 2014) which combines skeleton rotation and translation [3]. Another problem is the inferred data from the Kinect. When portion of the body is hidden from view the Kinect makes its own estimation about the location of the hidden body parts. For example when a person puts their arm behind their back while facing the Kinect, the arm's location is estimated. When the arm comes back on screen it may seem to jump if the estimation was wrong. A method for correcting for inaccurate estimations in Kinect data was found to be quite successful that uses a Gaussian filter passed over the joint movements to smooth out irregularities in the sequence [6]. I used the smae approach with a different filter passed over the data set.

Dynamic time warping as an algorithm has been extensively studied in the use of signal processing. Much of this work has been done with speech and audio signals which have some challenges when mapping to multidimensional image and body data [4]. The basic concepts of using dynamic time warping involve expanding and contracting time sequences compared to each other to find a best match.

Similarly LSTM networks have been around for years and been explored in different avenues, often with speech and audio signals [8]. Studies have been done to perform sequence classification on Kinect skeleton data [9]. The availability of Kinect 3D skeleton data is a relatively recent phenomenon so there still many unexplored possibilities. With the skeleton positions we train a neural network on sequence data gathered from test subjects performing exercises over many iterations until it can reliably classify exercises. In addition to the use of Kinect sensor data, there are studies that use different movement sensors like accelerometers to classify exercises using neural networks with comparable results to our own [24][25]. A different study includes comparing using and LSTM and DTW for gesture classification using input from accelerometers and gyrometers. In this study LSTM accuracy was much worse at 75% as compared to DTW at 95% [28].

Neural networks have been studied in comparison to many classification techniques for movements. One study compared many classification techniques to neural networks using Kinect skeleton data. These other classification were K nearest neighbors, Support Vector Machines, and Bayes classifiers. Each method has its own merits with some being superior at classification, some being better at assessing the quality of the movement, and others better for performance [30][31]. This experiment was used for classifying still poses rather than a continuous sequence of movements [10]. Another study created a system for classifying a sequence of movements of subject playing video games showing that the Kinect has a comparable level of accuracy to human viewers when it comes to counting and classifying movements [16]. What the studies note as a major motivation for using the Kinect is the ease of use and the simplicity of the skeleton data [21][23][26]. Having the skeleton data as inputs allows for much more speed and precision as opposed to image data alone [11]. Without processing the image and just using skeleton data allows for some applications to do real-time classification when it would otherwise be difficult to do so without specialized hardware [22]. One study was able to achieve near real-time classification using Kinect data and neural networks having only a one second delay [27]. The Kinect depth data also adds a layer of accuracy that is difficult to extract from images only. One study was able to accurately measure minute changes in breathing and heart rate of subjects thanks to these additional sensors [12]. Another major advantage of the Kinect is the lack of requirement for subject to wear anything obtrusive. Sensor technologies are still being studied with the same goal of therapeutic enhancement but typically require such equipment [13].

A large portion of the research done with the Kinect and on image processing in general is gesture recognition. This thesis can be considered a subset of this research. One study “Easy gesture recognition for Kinect, Advances in Engineering Software” (Rodrigo Ibañez, Álvaro Soria, Alfredo Teyseyre, Marcelo Campo 2014) while not explicitly targeted for physical therapy did use the Kinect to achieve an over 99% classification rate using Dynamic Time Warping and Hidden Markov Models. The classifications were for gestures recorded by the Kinect of which there were only seven. Hidden Markov Models use statistical probabilities to predict sequence states based on previous inputs and their predictability. Baysian classifiers are an extension of this process into multiple levels [14].

One of the aims of this Thesis is to put together s data set of 3D skeleton, RGB, and Depth data that can be made publicly available and used by future researchers. I am quite thankful for the efforts of others to make their own data sets publically available such as the gesture recognition library for the Kinect [15].

As the technology evolves research is being conducted to compare individual versions of the Kinect for their usefulness. This Thesis uses the Kinect 2 for all of its tests but the older Kinect is still heavily used. Many studies compare the Kinect 1 and the Kinect 2 and most often find the Kinect 2 to be more accurate [17]. The Kinect 1 has the advantage of being more thoroughly explored and lower in cost however [18][19][20].

This study differs from the existing work in a number of ways. Building on previous research I implement an improved skeleton normalization method that uses more sequence input to make its calculation and also accounts for the different sizes of subjects. The DTW methods that I implement use a dozen classes which can be quite slow so some unique optimizations are employed. The boundary limits are defined by the minimum and maximum length of time it takes to perform exercises. The distance given by DTW to show how closely the exercises line up is also given some boundary limitations based on previously compared exercises and their distance measures. The LSTM implementation used is new in that it has two layers of classification. The first layer is used to classify which exercise is being used at a particular period of time. The second layer takes the output from the first layer and classifies the exercise as either being in the first half of the movement or in the second half. From the second layer of classification we can count and measure the individual repetitions. I also conducted additional experiments using the LSTM for the first layer classification and DTW for the second layer. The LSTM implementation uses new features in the matlab Deep Learning Toolbox that have yet to be fully explored. Sequence to sequence classification takes a sequence as input and classifies an entire sequence as output. Past LSTM implementations using the Kinect and in matlab are optimized to classifying a single frame as output [10]. These new methods from the Deep Learning Toolbox are optimized for output of an entire sequence.

3. METHODOLOGY

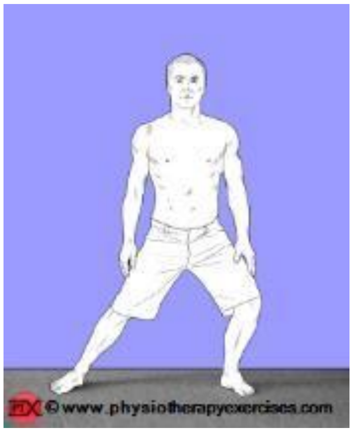
3.1 Overview

In this thesis I explore the efficacy and utility of two different strategies for analyzing Kinect RGB-D video data. Dynamic time warping (DTW) and a long short-term memory (LSTM) network are used to classify a series of movements performed by a human in front of the Kinect sensor into classes of exercises.

The approach includes data collection of 10 subjects performing sequences of instruction led exercises. Once the data is collected there is a series of preprocessing steps performed on the data to make it usable for comparison by DTW and LSTM. Then the data is analyzed using the two strategies outlined above.

For this study I selected twelve exercises (See Figure 1) that are used in physical therapy with a standard set of instructions. Modifications were made to those where an object is placed in between the Kinect sensor and the subject by removing the object from the exercise. These exercises vary in similarity with some involving the exact same body parts moving in a different way and others involving completely different joints altogether. These exercises are number 1 through 12.

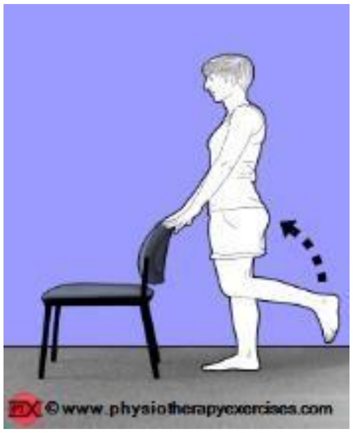
1. Adductor stretch



Client`s aim
To stretch tight tissue over your inner thigh and knee.

Client`s instructions
Position yourself in standing with feet wide apart. Shift weight to one side by bending your knee on the same side and maintain the position. Change your position so that you receive maximal stretch over your inner thigh as instructed by your physiotherapist.

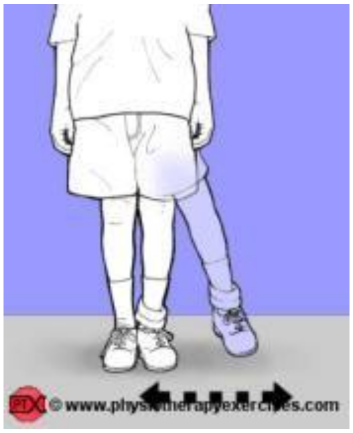
2. Hamstring strengthening in standing



Client`s aim
To strengthen the muscles in the back of your knee.

Client`s instructions
Position yourself standing holding onto the back of a chair. Start with your knee straight. Take your heel towards your bottom. Finish with your knee bent. Ensure that you keep your thigh straight.

3. Hip abductor strengthening in standing using sandbag weights

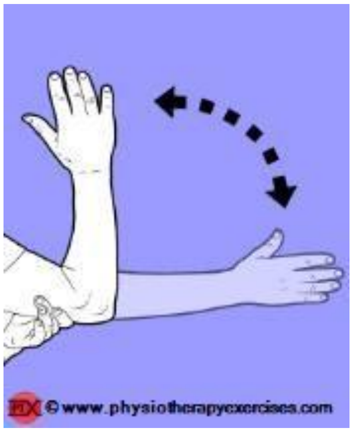


Client`s aim
To strengthen the muscles at the side of your hip.

Client`s instructions
Position yourself standing with a weight around your ankle. Start with your leg beside your body. Finish with your leg away from your body.

Figure 1. Exercise Diagrams

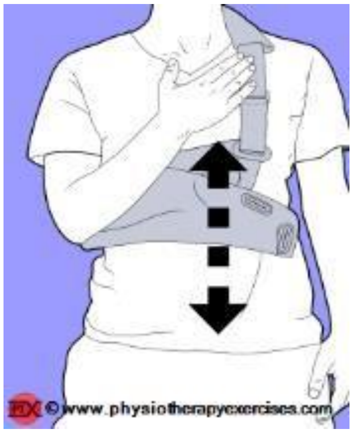
4. Elbow flexion



Client`s aim
To improve your ability to bend your elbow.

Client`s instructions
Position yourself with your elbow straight. Bend your elbow so that your palm moves towards your shoulder.

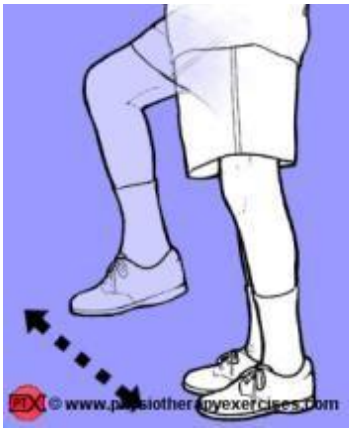
5. Elbow flexion and extension skimming body



Client`s aim
To maintain or improve range of motion of your elbow with your arm in a sling.

Client`s instructions
Position yourself standing with your arm outside of the sling. Keep your hand against you stomach and slowly straighten your elbow as much as possible. Keep your hand against your stomach and bend your elbow. Ensure that your hand skims your body and that your shoulder remains still.

6. Hip flexor strengthening in standing

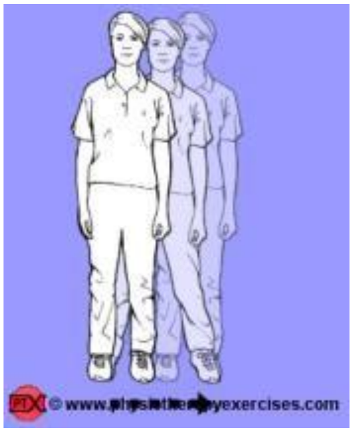


Client`s aim
To strengthen the muscles at the front of your hip.

Client`s instructions
Position yourself standing with your feet together. Start with your hip straight. Lift your hip and knee in front of you.

Figure 1. Continued

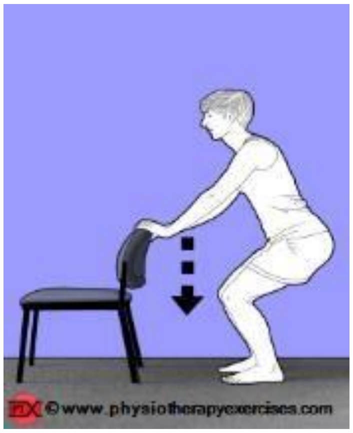
7. Stepping sideways



Client`s aim
To improve your ability to walk.

Client`s instructions
Position yourself standing with your feet together. Practice stepping sideways. Ensure that your knees are kept straight and your feet point forwards.

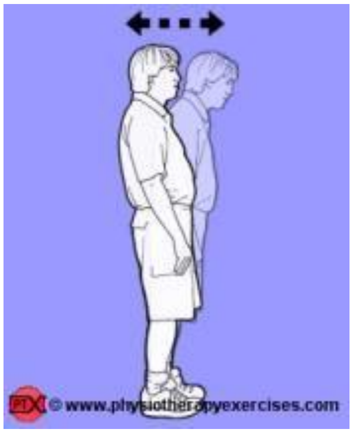
8. Squatting



Client`s aim
To strengthen the muscles that straighten your leg.

Client`s instructions
Position yourself standing holding onto the back of a chair or table. Start with your knees straight. Bend your knees and move your bottom back. Ensure to keep your back straight and your heels on the floor and your weight is equally borne through both legs.

9. Stand and shift weight forwards and backwards

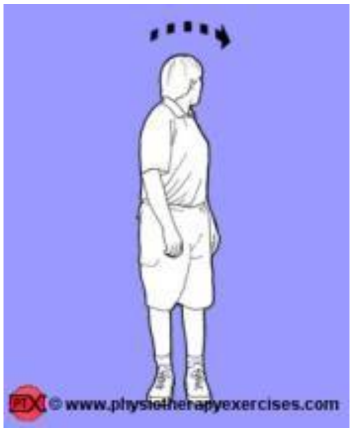


Client`s aim
To improve your ability to stand and balance.

Client`s instructions
Position yourself standing with your feet slightly apart. Practice leaning forwards and backwards. Ensure that the movement occurs at your ankles, your hips stay straight and your feet do not move. Feel your weight through the balls of your feet as you lean forwards and through your heels as you lean backwards. Go as far as you can without moving your feet or stepping.

Figure 1. Continued

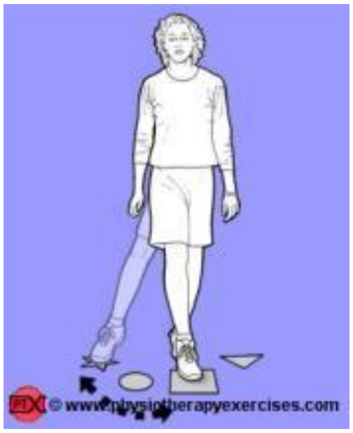
10. Stand and look behind



Client`s aim
To improve your ability to stand and balance.

Client`s instructions
Position yourself standing with your feet slightly apart. Practice turning your head to look over your shoulder. Aim to look around behind you as far as you can, without moving your feet or taking a step.

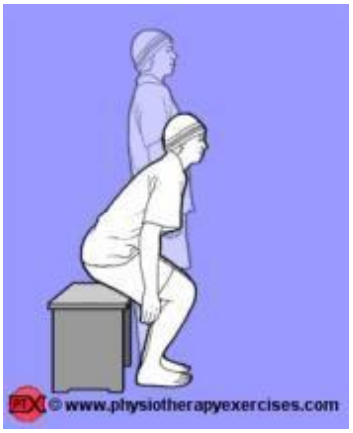
. Maintaining single-leg stance while moving the other foot to targets in a semi-circle



Client`s aim
To improve your ability to weight-bear through your affected leg.

Client`s instructions
Position yourself standing on your affected leg with targets placed in a semi-circle on the floor in front of you. Practice moving your unaffected foot from one target to another. Ensure that your unaffected foot only lightly touches the targets.

12. Standing up and sitting down

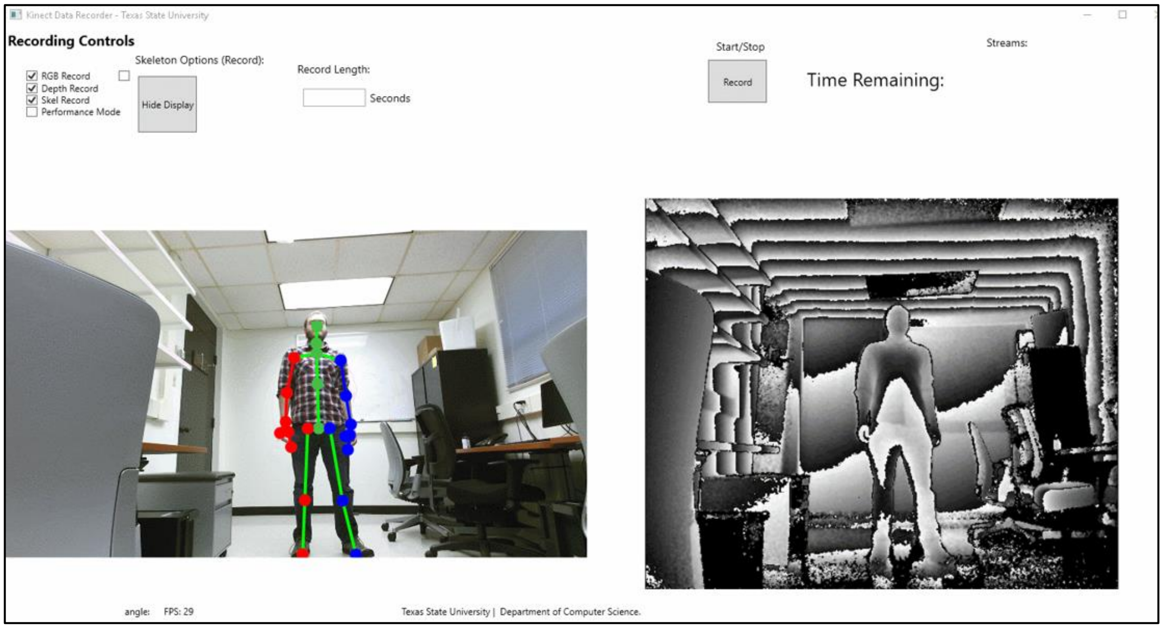


Client`s aim
To improve your ability to stand up or sit down.

Client`s instructions
Position yourself sitting with your feet underneath your knees. Practice standing up and sitting down. Ensure that your shoulders and knees move forward while you move between sitting and standing, and your weight is borne equally through both legs.

Figure 1. Continued

3.2 Collecting Data



Kinect Data Recorder - Texas State University

Recording Controls

☑ RGB Record
☑ Depth Record
☑ Skel Record
□ Performance Mode

Record Length: Seconds

Time Remaining:

angle: FPS: 29

Texas State University | Department of Computer Science.

Figure 2. Data Collection Application

Prior to collecting data from various test subjects, each provided written consent to this department’s use of the provided data. The collection involved each user to perform a sequence of physical therapy exercises as instructed, in front of a Microsoft Kinect, to the best of their abilities. There were two test cases that each subject performed. This first was performing in sequence each of a dozen exercises in a specific order

{Adductor stretch,
Hamstring strengthening in standing,
Hip abductor strengthening in standing,
Elbow flexion,
Elbow flexion and extension skimming body,
Hip flexor strengthening in standing,
Side-stepping,
Squatting,
Stand and shift weight forwards and backwards,
Stand and look behind,
Stand on one leg and move the other leg,
Standing up and sitting down}

Example 1. Exercise Sequence

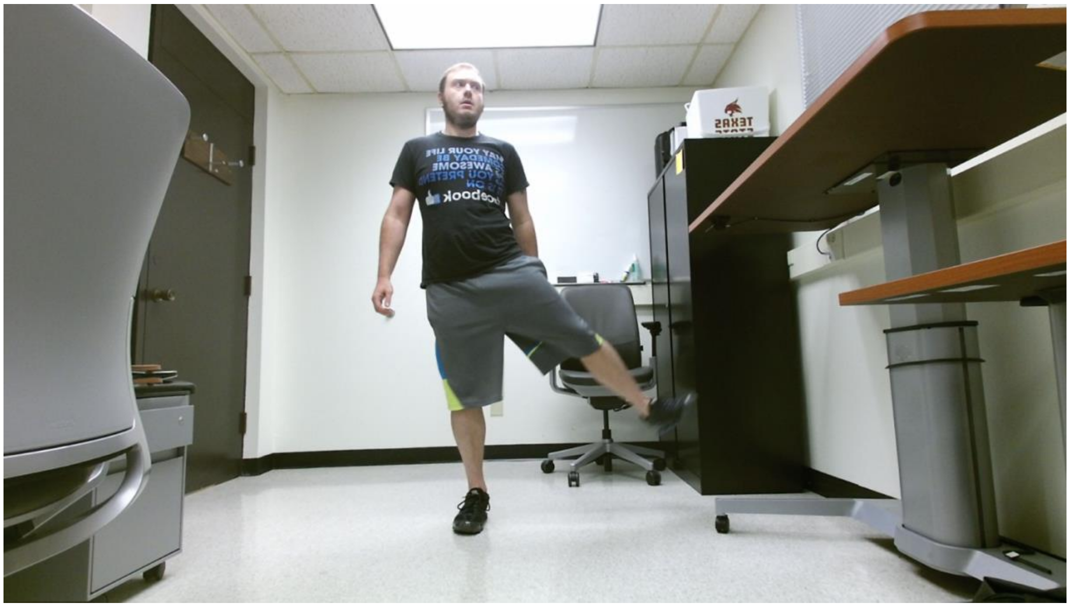
for five repetitions each. The second test case was performing the same set of exercises in a randomized order and using a random number of repetitions between two and five.

Using the Kinect we gathered three data streams. The Kinect skeleton data stream plots twenty five joints key joints along the body in 3 dimensions{x,y,z}.

Joint	X	Y	Z
SpineBase,	-0.0976	-0.0037	2.2424
SpineMid,	-0.1000	0.3168	2.2641
Neck,	-0.1018	0.6249	2.2726
Head,	-0.1067	0.7717	2.2855
ShoulderLeft,	-0.2851	0.4986	2.2465
ElbowLeft,	-0.3525	0.2544	2.2483
WristLeft,	-0.3742	0.0581	2.1579
HandLeft,	-0.3784	0.0218	2.1474
ShoulderRight,	0.0836	0.4933	2.2293
ElbowRight,	0.1548	0.2449	2.2172
WristRight,	0.1907	0.0553	2.1163
HandRight,	0.1851	0.0303	2.1103
HipLeft,	-0.1830	-0.0029	2.2049
KneeLeft,	-0.2088	-0.3321	2.2395
AnkleLeft,	-0.2343	-0.6198	2.2457
FootLeft,	-0.2343	-0.6522	2.1052
HipRight,	-0.0090	-0.0042	2.2022
KneeRight,	0.0453	-0.3593	2.2627
AnkleRight,	0.0945	-0.6178	2.2793
FootRight,	0.0893	-0.6502	2.1386
SpineShoulder,	-0.1015	0.5495	2.2728
HandTipLeft,	-0.3638	-0.0423	2.1290
ThumbLeft,	-0.3830	0.0335	2.1003
HandTipRight,	0.1818	-0.0298	2.1100
ThumbRight	0.1503	0.0360	2.0888

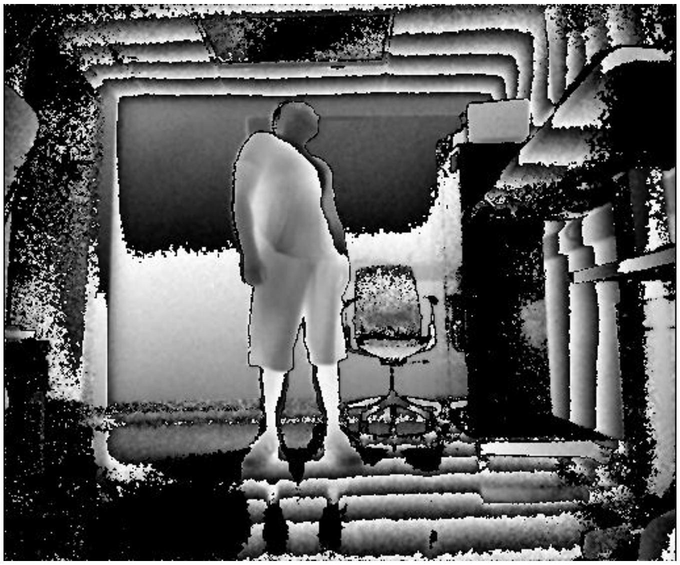
Example 2. Skeleton Frame

The RGB stream is a typical high quality color camera and each frame is stored as an individual image.



Example 3. RGB Frame

The Depth stream uses an infrared sensor array to create a gray scale showing depth gradients. Each depth frame is stored into an individual image file.



Example 4. Depth Frame

We also store recording timings by frame and stream for use in aligning frames from one stream to the corresponding frames in parallel streams. For classification purposes the RGB and Depth frames are not used. Initial testing with these frames included did not significantly impact the results but greatly increased the time taken to perform classification.

3.3 Skeleton Normalization

As each subject is recorded the 3D skeleton data is gathered based on the distance from and orientation to the Kinect sensor. As subjects are at variable distances and orientations, comparing the 3D skeleton data between users does not give meaningful results. A user performing an exercise may get misclassified just because they happen to be standing in a particular spot that has nothing to do with how well the exercise was performed. To get a meaningful comparison the skeletons must be spacially aligned. The methods used to normalize the skeleton are based on the methods described by Wang, Yao, and Lee [7]. The result is a skeleton aligned to a standard orientation. Translated so that the base of the spine is at the origin and rotated so that the shoulders begin at an equal depth. Typically the subject is rotated at a slight angle in relation to the Kinect sensor no matter how they might try to position themselves perpendicular to it and this normalization compensates for that. Additionally we have implemented scaling. The entire skeleton is scaled to in relation to the distance between the base of the spine and the top of the spine so as to match a standard spine length.

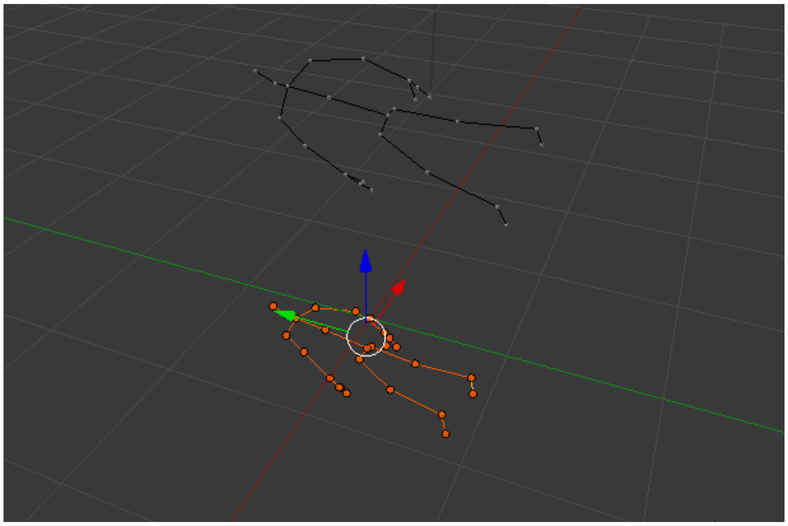


Figure 3. Skeleton Normalization

One change to the original normalization method was the frame sample used for making the calculations. Wang, Yao, and Lee used an average of the first 120 frames of as a representation by which to translate and rotate the body. In my result set the first 120 frames did not accurately represent the skeleton sequence as whole. I used an average of the entire sequence for my calculations. Not only did this not negatively impact the performance, it greatly improved the accuracy of my results. Movement comparisons became consistent across sequences from different users. In addition to account for inaccuracies with the Kinect, I tried various methods of smoothing the sequence data. The goal was to adjust for outliers in a joint sequence that appear as jitters on the video and ultimately found that smoothing with rloess gave the best results. Rloess stands for robust locally estimated scatterplot smoothing. This smoothing method takes a small segment in a sequence and finds the best fit curve for the local segment. Using the best fit curve, any outlier points that stray far away from the curve are revised to fit. This The Kinect would often have joints that seem to pop in and out of place when they came into close proximity to each other. After applying the rloess filter the movements are much more fluid and natural.

3.4 Stream Alignment

All three streams, RGB, Depth, and Skeleton, record concurrently and data collection was coded in an effort to maximize frame rates. The streams often do not have matching frame rates. To reconcile these differences I used the timings recorded with each frame to align one frame to the nearest frame in the parallel streams. The Skeleton stream typically has the highest frame rate so it is used as the master and the RGB and Depth frames are matched to the nearest skeleton frame. This usually leaves gaps in the sequence that have lower frame rates. To fill these gaps I used dynamic time warping as a means of interpolation. The two frames temporally nearest to a gap are aligned to each other using dynamic time warping and then merged into a new frame that fills in the sequence gap. For example say there are three adjacent skeleton frames but there are only two RGB frames within the same time window due to mismatched frame rates. The first RGB frame is aligned to the first skeleton frame and the second RGB frame is aligned to the last skeleton frame. This leaves the middle skeleton frame with no pairing. To match frame rates a new RGB frame is generated from the first and second RGB frame by aligning them with each other using DTW and merging the images together. The result is paired to the middle skeleton frame. This gap filling is a slow process usually taking several minutes per data set. While I ultimately did not use the RGB and Depth frames in the experiments, this will make the data much more accessible for future experiments where all of the streams are used.

3.5 Dynamic Time Warping

The dynamic time warping algorithm uses a programming technique wherein we take two sequences that we want to compare and compare each value in the first sequence to every other value in the second sequence and store their distances in a table corresponding to their placement in the original sequence. Using this table we select a path from the start of one sequence to the end of the other so that the sum of distances for the selected path is the least possible value.

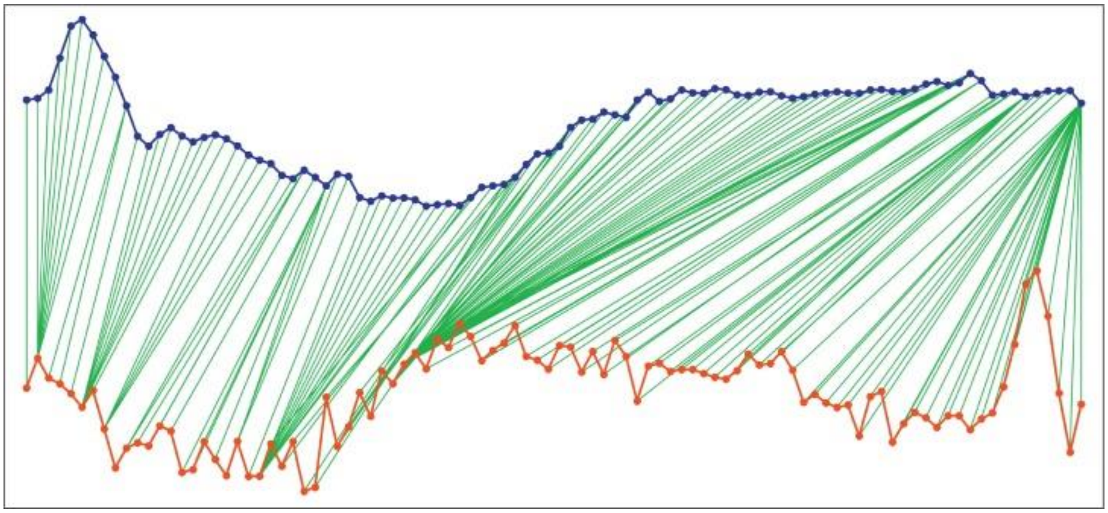


Figure 4. Visualization of DTW in Two Dimensions

Figure 4 shows the basic visualization the dynamic time warping algorithm for two 2D sequences. The green lines show which points in the blue sequence are compared to which points in the red sequence. The data collected has multiple joints with 3D coordinates plus RGB and depth information at each step of the sequence. I based the algorithm on a multidimensional time warping solution by Wöllmer, Al-Hames, Eyben, Schuller, and Rigoll [4]. Their method works for the skeleton data alone but there is an added layer of complexity because of the RGB and depth frames. Running DTW on sequences that include RGB and Depth frames is very costly in terms of performance which is why ultimately only the skeleton data is used.

DTW classification is based on comparing skeleton data collected from subjects performing exercises to skeleton data that represents the ideal standard of that exercise. I used myself performing each exercise as the standard by which the others are measured. DTW in this case always measures my skeleton data to the other users. The algorithm will classify a segment of the second sequence as a particular exercise if the distance measure for that segment is within a specific tolerance limit. The tolerance was developed by testing on a smaller subset and calibrated based on test results.

Each joint is segregated into its own sequence to only be compared with a matching joint for example wrists only matching to wrists and elbows only comparing to elbows. The Kinect software has a built-in mechanism to classify joints which was visually verified at the start of each experiment. Once segregated into individual sequences the joint movements can be compared using a basic 3D DTW algorithm. Their distances are aggregated and this aggregate is used to determine the best comparison for the sequence overall. For the RGB and Depth images the individual image pixels are first aligned using DTW and then those overall distance for one image is used to run another layer of DTW on the whole sequence of images. Ultimately this RGB and depth data was not used in my experiments as it did not change the end results in a reliable way and the process is time consuming to do even when attempting to leverage a GPU for acceleration.

3.6 Long Short-Term Memory Networks

Long short-term memory networks are deep learning networks that adapt and refine classifiers for a given training sequence which can be used to classify other sequences within a certain level of accuracy and even make predictions. My LSTM attempts to learn the long-term dependencies between values in exercise sequence. It uses iterative gradient descent to refine the dependency values and create a more accurate classification. I trained the network on a sample data subset that with all twelve exercises as one defined class. After training the network I use to classify a separate test data set. Both data sets were collected from different users. These methods were developed and tested in Matlab using new tools in the Machine Learning Toolbox. Within Matlab the LSTM architecture is defines as follows

```
inputSize = 75;
numHiddenUnits = 200;
numClasses = 12;

layers = [ ...
    sequenceInputLayer(inputSize)
    bilstmLayer(numHiddenUnits,'OutputMode','sequence')
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer]
```

The inputSize specifies the number of features we are giving the network at each point in the sequence. We have 25 joints that each have 3 coordinate numbers totaling to 75 inputs. The numHiddenUnits specifies the number of variables being calibrated to make a classification. I experimented with a number of different variable numbers and found that generally 200 works best. The more hidden units the slower each test iteration will be but typically it means fewer iterations to achieve a desired result. numClasses is 12 for the 12 exercise classes that we have selected. LSTM classification is done in two phases. For the first layer of classification only classifies which type of exercise is performed for each section of the sequence without trying to count repetitions. For the second layer we want to count the number of repetitions for each exercise so two classes are used. One class is for the first half of the exercise and one class is for the second half of the exercise. Then we count how many times both the first and second half of the exercise was performed to measure the number of exercise repetitions. For these experiments I opted for the bidirectional version of the lstmLayer which has the added advantage of being able to look ahead at the sequence data for dependencies as well as backward in the sequence. The softmaxLayer is also an optional added layer that includes that probability of each classification made at each iteration. This can slow the training for each new class added but increases the accuracy of the training at each stage.

In addition to defining the parameters for the LSTM we have to define our training methods

```
options = trainingOptions('adam', ...
    'GradientThreshold',1, ...
    'MaxEpochs',200,...
    'MiniBatchSize',10,...
    'SequenceLength','shortest',...
    'Shuffle','never',...
    'InitialLearnRate',0.009, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',20, ...
    'LearnRateDropFactor',0.49,...
    'Verbose',0, ...
    'Plots','training-progress');
```

Adam is short for adaptive momentum and this specifies algorithm used for the rate of change for the weights of hidden variables. GradientThreshold sets the maximum scale for which the weights can shrink and grow at a given time. MaxEpochs specifies the number of iterations run on the training set. MiniBatchSize is the length of the sequence considered for each point on the sequence. In other words, how far ahead or behind to look. SequenceLength shortest specifies to truncate the start and end of the sequence instead of padding them to equal 10. InitialLearnRate is the scale at which to start modifying variable weights. The learn rate is lowered by the LearnRateDropFactor after a number of iterations equal to the LearnRateDropPeriod. The last 2 options specify the output details. The shuffle options is set to never as it is necessary to preserve the order of frames within each exercise. It is however beneficial to randomize the exercises themselves before input so I created a separate function that reorders the training set before input into the LSTM but preserves individual exercises. Without this randomization the LSTM would learn to expect all of the exercises to be in a specific order. This is because about half of the training sets are intentionally ordered the same way. Lowering the batch size and adding this randomization worked well to resolve this issue. All of the numeric values were altered in different experiments to achieve the best results and modifying the training set usually requires making adjustments to the training variables. The values above are what have become standard for the experimental results below.

4. EXPERIMENTS AND RESULTS

4.1 DTW Experiment

The dynamic time warping experiments were run using a single user as a gold standard for the ideal movement. Each repetition of a particular exercise for that singled out baseline user were stretched using dynamic time warping to be the same length and then the results averaged together to get a single standard. The remaining data that did not contain that user was used for the experimental comparisons. For the frame classification without knowing where the start and end point of each exercise is we had to set some boundary parameters. First minimum exercise length is determined by the length of the shortest exercise performed in data collection which is 20 frames. DTW does not compare segments shorter than 20 frames. Similarly the maximum is set for the slowest an exercise at 120 frames. DTW does not compare segments longer than 120 frames. On a separate set of data I found the maximum distance between the baseline exercise and the corresponding exercise matches to use as a cutoff. The cutoff is padded by 20%. If the best match distance is not below this cutoff then it is not considered a match. Within these limits the segment with the best possible match is found to the nearest baseline exercise using the multiple joint DTW distance. After the first exercise is found then the same process is run recursively to the left and right of the already classified segments until the entire set is classified.

Table 1. Frame Classification By DTW

Class	True Negatives	True Positives	False Positives	False Negatives	Total	Recall	Precision	Error Rate
1	2258	240	14	25	265	0.90566	0.944882	0.015372
2	2235	213	76	13	226	0.942478	0.737024	0.035081
3	2388	103	22	24	127	0.811024	0.824	0.018132
4	2428	71	19	19	90	0.788889	0.788889	0.014978
5	2442	46	36	13	59	0.779661	0.560976	0.019314
6	2287	183	12	55	238	0.768908	0.938462	0.026409
7	2430	76	9	22	98	0.77551	0.894118	0.012219
8	2308	165	28	36	201	0.820896	0.854922	0.025227
9	2170	286	46	35	321	0.890966	0.861446	0.031927
10	2379	109	12	37	146	0.746575	0.900826	0.019314
11	2347	118	54	18	136	0.867647	0.686047	0.02838
12	2298	159	52	27	186	0.853305	0.7284	0.031336

Macro-averaged Precision	Macro-averaged Recall	Micro-averaged Precision	Micro-averaged Recall
0.821085545	0.827763722	0.837997635	0.841980198

The above table shows the accuracies for a frame by frame classification of DTW without predetermined segmentation. You can see a high level of predictability with most of these exercises. Some of the challenges with this method are time and scalability. This problem in the worst case is O(n^3) which is why we want to explore alternative methods. There is about a 17% error rate which might seem high but most of these errors are attributed to the boundaries of each exercise. The first few frames and the last few frames being misclassified do not affect the overall count for number of exercises.

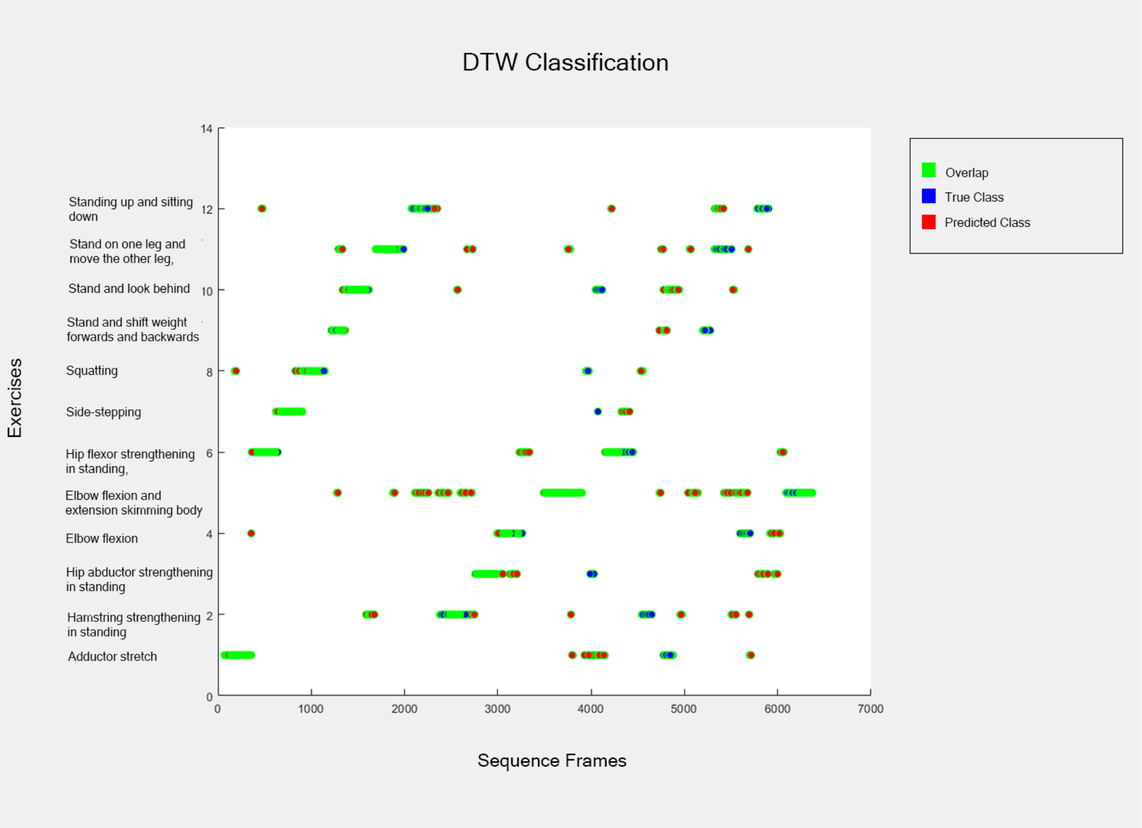


Figure 5. Frame Classification By DTW

Above you can see that the missed frames in red are in close proximity to the correct green frames and the same holds true for the blue frames which are true exercise frames but not predicted as such.

This next experiment is a little different. In this the start of a sequence of repetitions for a given exercise is given and DTW is used to count the number of reps for the sequence. The reason for this is to evaluate DTW solely for the purpose of counting repetitions regardless of how the frames are originally classified.

Table 2. Repetition Classification By DTW Ground Truth

Class	TRUE	Predicted	Root Square Error
1	95	95	0.00%
2	97	96	1.03%
3	96	95	1.04%
4	97	97	0.00%
5	89	93	4.49%
6	92	92	0.00%
7	94	95	1.06%
8	95	100	5.26%
9	82	80	2.44%
10	85	86	1.18%
11	87	87	0.00%
12	85	85	0.00%
		Root Mean Square Error:	1.38%

As the table above shows, DTW is very reliable with just 1.38% error rate on average when counting repetitions. The left column contains the class number that corresponds to the same exercise number. The TRUE column is the actual number of repetitions preformed for that exercises in the tested sequence. The Predicted column are the number of repetitions for each exercise measured using DTW. The last column is the percentage error rate.

4.2 LSTM Experiments

These next experiments were done in two phases. The first layer just classifies the movements of the sequence without attempting to count the repetitions. By doing this the two functions can be evaluated independently. In the following table the movements are shown by classification number and the error rates are shown individually and as averages. These represent the number of frames classified but not the movements as a whole.

Table 3. First Layer Classification By LSTM Recognizing exercise Type

Class	True Negatives	True Positives	False Positives	False Negatives	Total	Recall	Precision	Error Rate
1	2177	215	14	50	265	0.811321	0.938865	0.026059
2	2141	208	89	18	226	0.920354	0.700337	0.043567
3	2311	121	18	6	127	0.952756	0.870504	0.009772
4	2366	58	0	32	90	0.644444	1	0.013029
5	2397	50	0	9	59	0.847458	1	0.003664
6	2209	140	9	98	238	0.588235	0.939597	0.043567
7	2351	77	7	21	98	0.785714	0.916667	0.011401
8	2223	173	32	28	201	0.860697	0.843902	0.02443
9	2065	246	70	75	321	0.766355	0.778481	0.059039
10	2275	145	35	1	146	0.993151	0.805556	0.014658
11	2271	117	49	19	136	0.860294	0.704819	0.027687
12	1776	521	62	97	618	0.843042	0.893654	0.064739

Macro-averaged Precision	Macro-averaged Recall	Micro-averaged Precision	Micro-averaged Recall
0.866031726	0.822818402	0.843241042	0.82019802

The following graph shows us how even missing a number of frames does not mean a bad result. This graph shows the sequence frames along the x axis and the classification along the y axis. Blue dots are places where a frame should have been classified as a certain class or a false negative. Red dots represent a frame that was wrongfully classified or a false positive. Green dots are the true positive classifications. As you can see all of the misclassifications are always adjacent to correct classifications and are only around boundaries when switching between different movements. This happens because most of the exercises share a common starting and ending position which is a subject standing with arms at their sides.

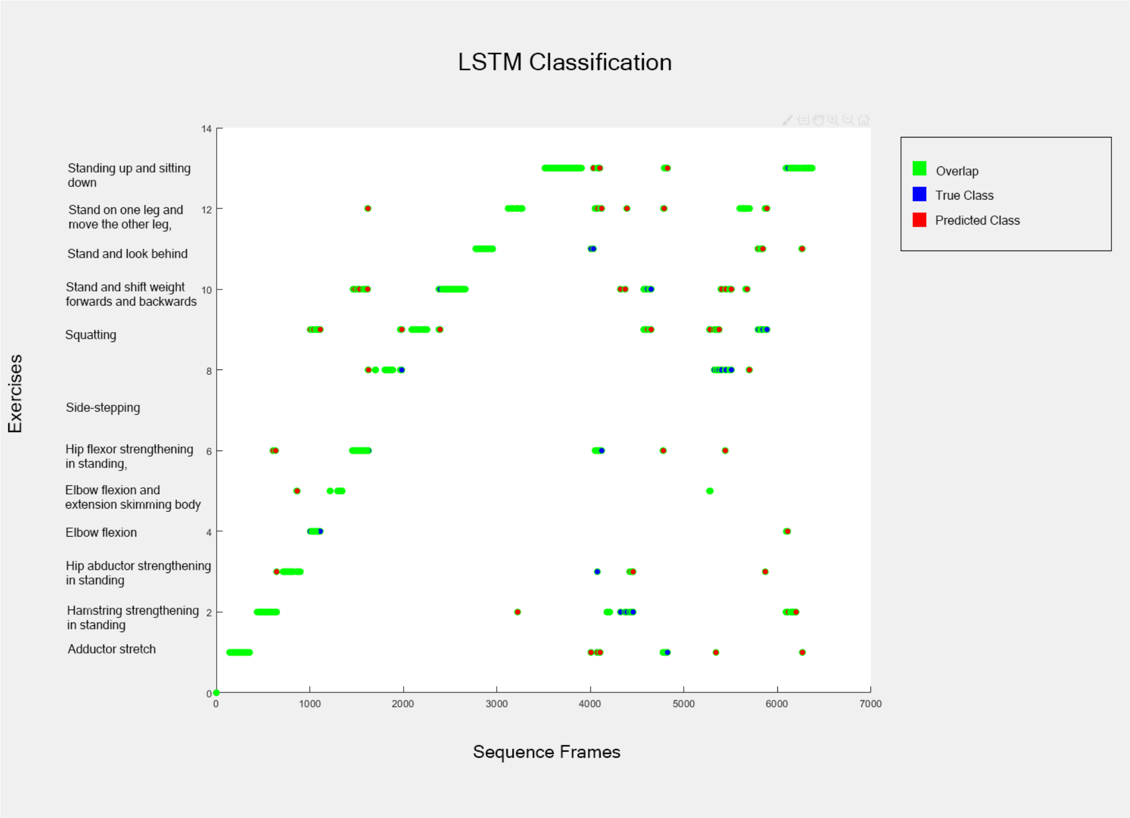


Figure 6. Frame Classification By LSTM

Table 4.
Repetition Classification By LSTM Using Ground Truth Exercise Type Input

Class	TRUE	Predicted	Root Square Error
1	95	95	0.00%
2	97	96	1.03%
3	96	93	3.13%
4	97	87	10.31%
5	89	92	3.37%
6	92	99	7.61%
7	94	93	1.06%
8	95	101	6.32%
9	82	80	2.44%
10	85	86	1.18%
11	87	86	1.15%
12	85	85	0.00%
		Root Mean Square Error:	3.13%

The first layer of the LSTM only classifies the frames but it does not give an exercise count. To do this a second network is used that is trained on two classifications. One class represents the first half of a movement and the second class represents the second half of the individual movement. This is done independently for each movement. To measure the accuracy of this method independently from the first layer of classification I executed this layer on sequences where the first level of classification was already known to be correct. The results of this test are in the above table.

To test the overall accuracy of using the methods end to end the same test was run on the output from the first layer. This means that there was already a certain degree of error with the classification of frames. This same test sequence was also used for counting repetitions by DTW as a comparison.

Table 5.
Repetition Classification By DTW Using LSTM Predicted Exercise Type Input

Class	TRUE	Predicted	Root Square Error
1	95	96	1.05%
2	97	96	1.03%
3	96	94	2.08%
4	97	96	1.03%
5	89	93	4.49%
6	92	92	0.00%
7	94	92	2.13%
8	95	92	3.16%
9	82	80	2.44%
10	85	83	2.35%
11	87	83	4.60%
12	85	85	0.00%
		Root Mean Square Error:	2.03%

Table 6.
Repetition Classification By LSTM Using LSTM Predicted Exercise Type Input

Class	TRUE	Predicted	Root Square Error
1	95	96	1.05%
2	97	98	1.03%
3	96	91	5.21%
4	97	85	12.37%
5	89	95	6.74%
6	92	99	7.61%
7	94	98	4.26%
8	95	100	5.26%
9	82	88	7.32%
10	85	87	2.35%
11	87	85	2.30%
12	85	85	0.00%
		Root Mean Square Error:	4.63%

5. DISCUSSION

The two algorithms discussed have some vast differences in implementation and usage. From these analyses LSTM seems far superior to DTW in terms of classification speed and accuracy. More time must be invested initially in the training process but once that is complete then classification can be done in real time. LSTM is superior in speed but the drawback seems to be in the second layer of classification that is used for counting repetitions. The best result comes from using the LSTM to classify the sequences by exercise alone and then use DTW to classify individual exercises and count repetitions. This has the added advantage of coming with a distance result for each repetition which can be used to gage how well the exercise was performed. DTW has a quadratic time complexity so naturally one would want to avoid using this method for any significant set of classification. However for the repetition counting only 1 class is considered at a time allowing for fast analysis.

6. CONCLUSION AND FUTURE WORK

The results show that neural networks are a very powerful tool when it comes to classifying Kinect skeleton data. These methods still have many possibilities that can be explored. The best results came from combining both LSTM and DTW. LSTM results were more accurate in classifying the exercises in the first layer and scale much better than DTW. One of the greatest challenges with using DTW alone for classification is that in including additional exercises to the set of classes. The time taken by DTW grows exponentially with each new exercise. By using the LSTM in the first layer, the number of classes required to be run by DTW for repetition counting is always limited to two classes(first half of an exercise and second half of an exercise). This changes the scaling from exponential to linear. With these performance improvements we can utilize these algorithms to create applications that will classify exercises and count repetitions in real time. These applications can be used to guide patients through physical therapy sessions and report results to a medical professional for analysis without either person having to be in the same room. The data collected can be shared between doctors and by having more empirical data should give them more insight into how patients recover and advance through therapy.

There are many future improvements that can be made for better accuracy within these algorithms. While the results are promising, improved accuracy would make for a more useful tool in a clinical setting. Future research should explore changes such as creating an additional LSTM classification layer. This layer would run before the existing layers do by classifying the exercises into groups based on which part of the body is moving. Localizing the body parts and narrowing down the exercises in this layer reduces the chance of mismatch in subsequent layers. Another challenge that I hope will be explored in future research is incorporating the RGB and Depth data streams more effectively. The skeleton data is generated based on the RGB and Depth streams but this is only a fraction of the data captured. There is much potential and I think I am only scratching the surface

REFERENCES

[1] E. E. Stone and M. Skubic, "Unobtrusive, Continuous, In-Home Gait Measurement Using the Microsoft Kinect," in *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2925-2932, Oct. 2013. doi: 10.1109/TBME.2013.2266341

[2] Pfister, A., West, A. M., Bronner, S., & Noah, J. A. (2014). Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis. *Journal of Medical Engineering & Technology, 38*(5), 274-280. doi:10.3109/03091902.2014.909540

[3] A. Fern'ndez-Baena, A. Susín and X. Lligadas, "Biomechanical Validation of Upper-Body and Lower-Body Joint Movements of Kinect Motion Capture Data for Rehabilitation Treatments," *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*, Bucharest, 2012, pp. 656-661.
doi: 10.1109/iNCoS.2012.66

[4] Wöllmer, M., Al-Hames, M., Eyben, F., Schuller, B., & Rigoll, G. (2009). A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. *Neurocomputing,* 73(1-3), 366-380.
doi:10.1016/j.neucom.2009.08.005

[5] Silva, D.F. (2016). On the Effect of Endpoints on Dynamic Time Warping.

[6] Chiang, An-Ti & Chen, Qi & Wang, Yao & R. Fu, Mei. (2018). Kinect-Based In-Home Exercise System for Lymphatic Health and Lymphedema intervention. *IEEE Journal of Translational Engineering in Health and Medicine.* PP. 1-1. 10.1109/JTEHM.2018.2859992.

[7] Wei, T & Qiao, Y & Lee, B. (2014). Kinect skeleton coordinate calibration for remote physical training. *MMEDIA - International Conferences on Advances in Multimedia.* 66-71.

[8] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *In Advances in neural information processing systems* (pp. 3104-3112).

[9] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., & Zheng, N. (2018). View Adaptive Neural Networks for High Performance Skeleton-based Human Action Recognition. *arXiv preprint arXiv:*1804.07453.

[10] Choubik, Youness & Mahmoudi, Abdelhak. (2016). Machine Learning for Real Time Poses Classification Using Kinect Skeleton Data. 307-311. 10.1109/CGiV.2016.66.

[11] Procházka, Aleš & Vyšata, Oldřich & Valis, Martin & Ťupa, Ondřej & Schätz, Martin & Mařík, Vladimír. (2015). Use of the image and depth sensors of the Microsoft Kinect for the detection of gait disorders. *Neural Computing and Applications.* 26. 1621 - 1629. 10.1007/s00521-015-1827-x.

[12] Procházka, Aleš & Schätz, Martin & Vyšata, Oldřich & Valis, Martin. (2016). Microsoft Kinect Visual and Depth Sensors for Breathing and Heart Rate Analysis. *Sensors.* 16. 996. 10.3390/s16070996.

[13] A. D. Gama, T. Chaves, L. Figueiredo and V. Teichrieb, "Guidance and Movement Correction Based on Therapeutics Movements for Motor Rehabilitation Support Systems," 2012 *14th Symposium on Virtual and Augmented Reality,* Rio Janiero, 2012, pp. 191-200.
doi: 10.1109/SVR.2012.15

[14] Rodrigo Ibañez, Álvaro Soria, Alfredo Teyseyre, Marcelo Campo, Easy gesture recognition for Kinect, *Advances in Engineering Software,* Volume 76, 2014, Pages 171-180, ISSN 0965-9978

[15] Fabrizio Pedersoli, Nicola Adami, Sergio Benini, and Riccardo Leonardi. 2012. XKin -: eXtendable hand pose and gesture recognition library for kinect. *In Proceedings of the 20th ACM international conference on Multimedia* (MM '12). ACM, New York, NY, USA, 1465-1468

[16] Rosenberg M, Thornton AL, Lay BS, Ward B, Nathan D, Hunt D, et al. (2016) Development of a Kinect Software Tool to Classify Movements during Active Video Gaming. PLoS ONE 11(7): e0159356

[17] T. Hachaj, M. R. Ogiela and K. Koptyra, "Effectiveness Comparison of Kinect and Kinect 2 for Recognition of Oyama Karate Techniques*,"* 2015 *18th International Conference on Network-Based Information Systems,* Taipei, 2015, pp. 332-337

[18] Alina Delia Călin, "Variation of pose and gesture recognition accuracy using two kinect versions*", Inovations in Intelligent SysTems and Applications (INISTA) 2016 International Symposium on,* pp. 1-7, 2016.

[19] Wei-Yuan Kuo, Chien-Hao Kuo, Shih-Wei Sun, Pao-Chi Chang, Ying-Ting Chen, Wen-Huang Cheng, "Machine learning-based behavior recognition system for a basketball player using multiple Kinect cameras*", Multimedia & Expo Workshops (ICMEW) 2016 IEEE International Conference on,* pp. 1-1, 2016

[20] Alexiadis, D.S., Kelly, P., Daras, P., O'Connor, N.E., Boubekeur, T., Moussa, M.B.: Evaluating a dancer's performance using kinect-based skeleton tracking. *In: Proceedings of the 19th ACM International Conference on Multimedia*, pp. 659–662. ACM (2011)

[21] *Gowing M., Ahmadi A., Destelle F., Monaghan D.S., O'Connor N.E., Moran K. (2014) Kinect vs. Low-cost Inertial Sensing for Gesture Recognition. In: Gurrin C., Hopfgartner F., Hurst W., Johansen H., Lee H., O'Connor N. (eds) MultiMedia Modeling. MMM 2014. Lecture Notes in Computer Science, vol 8325. Springer, Cham*

[22] Antón, David & Goni, Alfredo & Illarramendi, Arantza. (2015). Exercise Recognition for Kinect-based Telerehabilitation. *Methods of Information in Medicine.* 54. 145-155. 10.3414/ME13-01-0109.

[23] Huang, F., & Huang, S. (2011). Interpreting American Sign Language with Kinect.

[24] T. T. Um, V. Babakeshizadeh and D. Kulić, "Exercise motion classification from large-scale wearable sensor data using convolutional neural networks," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Vancouver, BC, 2017, pp. 2385-2390

[25] R. Poppe, "A survey on vision-based human action recognition*", Image and Vision Computing,* vol. 28, no. 6, pp. 976-990, 2010

[26] F. Ofli, G. Kurillo, R. Bajcsy Obdrlek, H.B. Jimison, M. Pavel, "Design and evaluation of an interactive exercise coaching system for older adults: Lessons learned*",* I*EEE Journal of Biomedical and Health Informatics,* vol. 20, no. 1, pp. 201-212, Jan 2016

[27] M. Weber, M. Liwicki, D. Stricker, C. Scholzel and S. Uchida, "LSTM-Based Early Recognition of Motion Patterns," *2014 22nd International Conference on Pattern Recognition,* Stockholm, 2014, pp. 3552-3557

[28] Lefebvre G., Berlemont S., Mamalet F., Garcia C. (2013) BLSTM-RNN Based 3D Gesture Classification. In: Mladenov V., Koprinkova-Hristova P., Palm G., Villa A.E.P., Appollini B., Kasabov N. (eds) Artificial Neural Networks and Machine Learning – ICANN 2013. ICANN 2013. *Lecture Notes in Computer Science,* vol 8131. Springer, Berlin, Heidelberg

[29] Y. Wu, B. Zheng and Y. Zhao, "Dynamic Gesture Recognition Based on LSTM-CNN," *2018 Chinese Automation Congress (CAC),* Xi'an, China, 2018, pp. 2446-2450

[30] Vonstad, Elise Klæbo; Su, Xiaomeng; Vereijken, Beatrix; Nilsen, Jan Harald; Bach, Kerstin. (2018) Classification of movement quality in a weight-shifting exercise. *CEUR Workshop Proceedings. vol. 2148*

[31] Li, L., & Vakanski, A. (2018). Generative Adversarial Networks for Generation and Classification of Physical Rehabilitation Movement Episodes. *International journal of machine learning and computing, 8(5), 428–436.*