RECOGNIZING SPECIFIC ERRORS IN HUMAN PHYSICAL EXERCISE PERFORMANCE WITH MICROSOFT KINECT

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Ryan Staab

June 2014

COMMITTEE MEMBERSHIP

TITLE:	Recognizing specific errors in human physical exercise performance with Microsoft Kinect
AUTHOR:	Ryan Staab
DATE SUBMITTED:	June 2014
COMMITTEE CHAIR:	Franz Kurfess, PhD Professor of Computer Science
COMMITTEE MEMBER:	Jane Zhang, PhD Associate Professor of Electrical Engineering
COMMITTEE MEMBER:	Todd Hagobian, PhD Assistant Professor of Kinesiology

ABSTRACT

Recognizing specific errors in human physical exercise

performance with Microsoft Kinect

Ryan Staab

The automatic assessment of human physical activity performance is useful for a number of beneficial systems including in-home rehabilitation monitoring systems and Reactive Virtual Trainers (RVTs). RVTs have the potential to replace expensive personal trainers to promote healthy activity and help teach correct form to prevent injury. Additionally, unobtrusive sensor technologies for human tracking, especially those that incorporate depth sensing such as Microsoft Kinect, have become effective, affordable, and commonplace.

The work of this thesis contributes towards the development of RVT systems by using RGB-D and tracked skeletal data collected with Microsoft Kinect to assess human performance of physical exercises. I collected data from eight volunteers performing three exercises: jumping jacks, arm circles, and arm curls. I labeled each exercise repetition as either correct or one or more of a select number of predefined erroneous forms. I trained a statistical model using the labeled samples and developed a system that recognizes specific structural and temporal errors in a test set of unlabeled samples. I obtained classification accuracies for multiple implementations and assess the effectiveness of the use of various features of the skeletal data as well as various prediction models.

ACKNOWLEDGMENTS

I would like to thank my advising committee, Dr. Jane Zhang, Dr. Franz Kurfess, and Dr. Todd Hagobian for their oversight and contributions to this work. I would also like to thank Cal Poly and the Computer Science department for giving me the opportunity to do this thesis. I would also like to acknowledge AI & Robotics Research Team (AIRR) for the application that facilitated this research. I want to especially thank my family for supporting me through college, and my girlfriend Lisa for being patient with me while I worked for so much of my time on this thesis during my final quarter.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Tracking and recognizing the activity of human agents through video is an important and challenging task in the field of computer vision. Human activity analysis can be defined as the following: given a data stream of a person performing physical activity, (i) detect the human agent in the video, (ii) track the motion of the human agent, (iii) classify what kind of activity the person is performing, and (iv) evaluate and assess the activity. The data stream is typically video captured from a camera device.

Much work has been done in vision-based human activity analysis. In the last 5-10 years, researchers have used many sensor technologies to detect and track humans and the activities they perform. Of particular interest are depth-enabled cameras such as the Microsoft Kinect. The additional depth dimension gathered from Kinect greatly improves the capabilities of machine learning and computer vision algorithms for human activity tracking, including the ability to generate real-time skeleton models of humans in various body poses [Wei, 2012], as well as achieving successful tracking of fingers and hand articulations [Raheja, 2011] [Oikonomidis, 2011].

Human activity analysis is an important component of many applications. Reactive Virtual Trainers for instance, need to analyze the performance of a trainee so the system can provide adequate feedback. Immersive virtual reality systems also need to be able to track the person who is engaged with the system, in order to recognize their actions for interaction within the virtual world. Security and surveillance systems can utilize human activity tracking to detect malicious or fraudulent behavior. There are many more examples of applications ranging from logistics support to home-based rehabilitation monitoring for traumatic brain injuries [Pollack, 2003]. Intelligent, reactive, and natural systems of interaction such as these are becoming increasingly viable and prevalent as the technology improves.

The motivation for this work comes from the context of a virtual reality health-promotion project directed by Zhang [Zhang, 2013]. Zhang's proposed system incorporates many functional aspects: an immersive virtual environment using VR technology such as Oculus Rift, instructional guidance for the performance of exercises, conveyance of knowledge of general exercise and nutrition guidelines, social networking, aid for program adherence, and more. This thesis focuses on a related subset of that project. In particular, the focus of this work is on human activity analysis for exercises for a Reactive Virtual Trainer system.

A Reactive Virtual Trainer (RVT) is an intelligent virtual agent that supports guided exercise or physical therapy. RVTs are typically used by a single person at a time, in a one-on-one environment, with the user having some degree of freedom over the course of interaction. The RVT might show the user the proper form for exercises and stretches via a graphical avatar character, motivate, monitor, and critique the user's performance of exercises, or make sure the user adheres to his or her planned exercise routine [Ruttkay, 2006].

There are many varieties of RVTs. A common goal among RVT implementations is achieving seamless human-computer interaction. To accomplish this, RVTs may employ natural language recognition, a graphical representation of the virtual trainer, gesture-navigated menus, and audio, visual, and textual feedback. The intended users of RVTs vary from healthy to handicapped or rehabilitating individuals.

RVT technology has the potential to replace the need for physical therapists or personal trainers by providing more cost-effective, more comprehensive, and more accessible feedback to users. Through the aforementioned benefits, RVTs can help promote exercise for its users. RVTs also offer flexibility for users as they can use the system on their own time rather than schedule an appointment with a trainer, as well as repeat exercises or instructions as many times as needed. Ruttkay et. al. explored the

functional requirements of a framework that can be used to author RVTs, emphasizing the need for adjusting tempo, pointing out mistakes, rescheduling exercises, as well as the ability to compose exercises from basic motions [Ruttkay, 2006]. This thesis is significant because it works towards health promotion through higher accuracy and more capable RVT systems.

Reports show that in the United States alone there are over 78 million adults and 12.5 million children and adolescents who are obese, which is more than double the rate (from 15% to 35.7%) among adults and triple the rate (from 5% to 17%) among children and adolescents from 1980 to 2010 [Ogden, 2012]. This trend is occurring globally as well. According to the World Health Statistics 2012 report, obesity has doubled between 1980 and 2008 in nearly every region of the world [WHS, 2012]. However, it is well established that leading a healthy lifestyle involving a balanced diet and exercise can greatly lower the risk of becoming obese [OSG, 2010] [Sallis, 1992].

Exercise in particular has been shown to bring a wide range of health benefits. Exercise can help prevent weight gain [Hunter, 2010], boost mood and fight depression [Cooney, 2013], reduce the likelihood of certain cancers, e.g. colon cancer and breast cancer [McCullough, 2012], reduce the likelihood of dementia [Ahlskog, 2011], boost memory [Leavitt, 2013], improve heart function [Ferreira, 2014], reduce the risk for type-2 diabetes [Solomon, 2013], increase longevity [Koch, 2011], and promote other long-term health benefits [Wallace, 2014].

The higher capability depth sensing and skeleton tracking of the Kinect is a critical contribution that enables the development of RVTs, performing greater than previous technologies such as Sony EyeToy, which lacked a depth sensor. In addition, the Kinect has been shown to be a competitive motion tracking device [Chang, 2012] [Bonnechère, 2014]. The Kinect for Windows features synchronized 640 x 480 pixels RGB and 3-D infrared depth sensors, along with a four-microphone array and motorized

tilt. The image sensors capture video at 30 frames per second. Figure 1 displays the specifications for the interaction space of the Kinect.

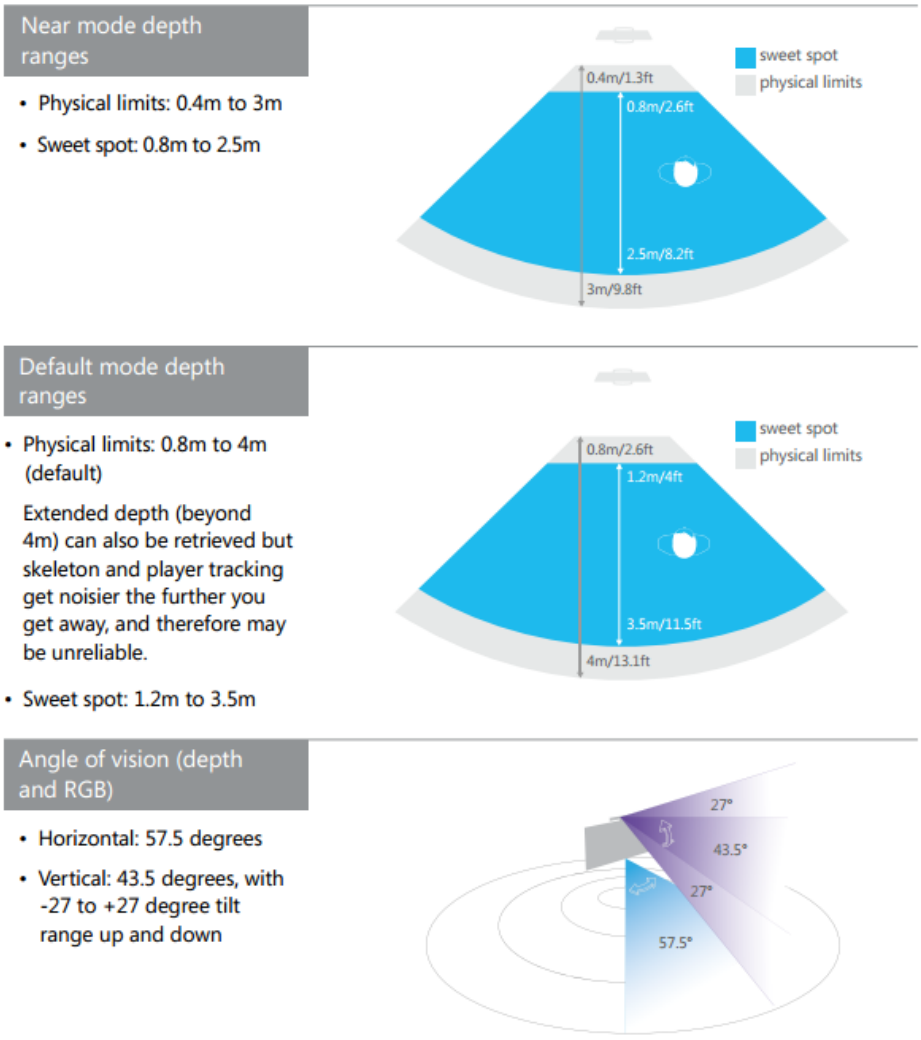

Figure 1. Kinect interaction space

The Kinect package also includes built-in skeleton tracking. The PrimeSense depth sensor hardware infers depth by emitting and analyzing a speckle pattern of infrared laser light through a technique known as structured light. Two approaches,

depth from focus and depth from stereo, determine the depth image of a scene. Depth from focus relies on the property that farther objects appear more blurry. The Kinect uses an astigmatic lens with varying focal lengths in x and y dimensions to improve the accuracy of this method. Depth from stereo relies on parallax, the property that when a scene is observed from two different angles, objects that are closer to the camera appear shifted laterally compared with objects that are farther away. The software is proprietary, however it is speculated that body parts are inferred using a randomized decision forest, learned from over 1 million training examples [Maccormick, 2014]. Other researchers have applied their own skeletal tracking algorithms, such as Kar's use of an extended distance transform skeletonisation algorithm using HAAR-Cascade detectors [Kar, 2010], however, the Kinect built-in skeleton tracking is used in this research.

Since the release of the original Kinect for Xbox in 2010, many researchers and independent parties hacked it to take advantage of the new Primesense RGB-D sensor hardware to achieve higher performance, higher accuracy gesture recognition. In 2012, Microsoft released the official Kinect for Windows along with its Software Development Kit, making development and research with Kinect manageable and accessible. The Kinect does, however, have the following limitations. A user must be in front of Kinect, facing towards the Kinect sensor. This makes it difficult to do error recognition on a variety of exercises such as sitting or lying exercises in which the user may be out of the viewing range of the Kinect. Additionally, there is a synchronization period in which the Kinect initializes the skeleton tracking.

The goal of this research was to achieve a high rate of classification accuracy in identifying specific human activity performance errors using Microsoft's Kinect for Windows v1.8. This work aims to establish a proof-of-concept for recognizing specific errors in form and tempo for predefined exercises and stretches. This work does not attempt to distinguish one exercise from another, rather, error classes are recognized

within a set of known exercises. Three specific exercises have been chosen for this work: arm curls, jumping jacks, and arm circles. These exercises exhibit a variety of motion characteristics, are well known, and are not arduous or associated with high risk of injury. The intended users targeted in this research were healthy individuals in the age range of 18 to 30.

My implementation uses the Kinect to detect and track the human performer, but does not need to determine which exercise or stretch is being performed. The user will select the exercise to be performed from a menu prior to execution. The core of my work lies in evaluation of the activity. I use a statistical machine learning methods to identify specific form and tempo errors. The work of this thesis contributes towards the field of computer vision and to the development of RVTs and thereby to the promotion of physical activity and promotion of health.

The rest of this paper is broken down into the following sections. Section II: Related Work discusses the current state of the art in human activity analysis and performance error recognition. Section III: Approach discusses the algorithms and machine learning models I have chosen to solve the problem of performance error recognition, and their considerations. Section IV: Methodology discusses in detail the development environment, the collection, preprocessing, and structure of the data, and details of implementation. Section V: Evaluation and Results discusses how I evaluated the performance and accuracy of my implementation and the results obtained. Section VI: Conclusion summarizes the work and results. Section VII: Future Work discusses further potential improvements.

II. RELATED WORK

A number of RVT systems have been developed. In the academic realm, much research has focused on using the Microsoft Kinect for rehabilitation and improving physical and mental function of the elderly. Kayama developed a Kinect-based exercise game for assessing dual-task function of the elderly [Kayama, 2014]. Bierla assessed the impact of Kinect based training on balance measures in older adults [Bierla, 2012]. Shin developed a game-based virtual system for rehabilitation of patients with stroke [Shin, 2014]. Rantz developed a continuous, unobtrusive, and environmentally mounted in-home fall risk assessment and detection system using the Kinect [Rantz, 2013]. There are also a number of Kinect-based virtual trainer games for the XBox platforms. Examples include UFC Personal Trainer, Nike+ Kinect Training, and XBox Fitness which features virtual avatars of athletes to guide the user. The work of Ruttkay and van Welbergen is similar to this thesis, except their focus is on the entire RVT system, including a strategy to provide feedback to the user, rather than on reaching high accuracy recognition of errors [Ruttkay, 2006]. As an alternative approach to the Kinect, Reyes proposes a tool for body posture analysis and skeleton joint estimation from a variety of sensor inputs, and uses this to perform gesture recognition on the correctness of physical exercises [Reyes, 2013].

Many human activity recognition tasks are similar to recognition of exercise performance errors. However, many of these tasks focus on recognition of more complex, more contextual human behavior that happens on longer timescales than does exercise error recognition. Cooking, entering a building, or walking a dog are examples of human activities that such systems seek to recognize. A common approach to that task uses Hidden Markov Models (HMM) and has been shown to perform well [van Kasteren, 2011]. HMM is a temporal probabilistic model that models correlations

between activities and observed sensor data. HMM can enable many interesting systems. Early on, HMMs have been used to track and recognize activities in which there is a grammatical context such as recognizing American Sign Language [Starner 1995]. While tracking arm and hand motions is important for my work, the motions that I analyze are repetitive and isolated rather than having meaningful context for each motion. Ikizler and Forsyth created a way to do text-queries of specific limb motions in long data streams such as surveillance and security videos [Ikizler, 2008]. Their system used a HMM to model short time scale limb behaviour built using a labelled motion capture set. Another example of HMM used for activity recognition is in the work of Trabelsi [Trabelsi, 2013]. However, Trabelsi's approach used unlabeled data and focused on wearable accelerometers rather than RGB-D or skeleton data. Nergui et. al. demonstrated that Kinect and HMM can be used as part of an autonomous mobile healthcare robot [Nergui, 2013]. Their system recognized patient's gait behaviors from calculated joint angles. Tang demonstrated that Kinect can also be used for recognizing hand gestures using a combination of features such as major/minor axis length rotation, eccentricity, orientation, radial histogram, dominant gradient direction, and SURF descriptors along with a HMM [Tang, 2014]. Many of these features, however, are not useful for recognizing body-scale structure and motion.

One-shot learning gesture recognition is another task closely related to exercise performance recognition. Models of gesture classes are learned from single examples of each class. Konečný approaches this problem with parallel temporal segmentation using histogram of gradients and histogram of optical flow with Quadratic-chi distance used to measure the discrepancy between histograms [Konečný, 2013]. Wan approaches the problem by clustering the gestures with K-means clustering to find codewords using a bag-of-features method with novel spatio-temporal feature representation [Wan, 2013].

A number of other research papers have focused on unsupervised learning techniques as well. In those cases, models are learned from unlabeled data, using clustering techniques such as the K-means algorithm in order to uncover the classes of interest from the data. Charles utilizes K-means algorithms to cluster 3D poses using a pictorial structure model and a mixture model of probabilistic masks [Charles, 2011]. His work, however, does not address dynamic motion. Weber developed a method for automatic generation of motion segmentation models to find recurring patterns in unlabeled motion data, and applies it to a virtual rehabilitation trainer [Weber, 2012].

III. APPROACH

Due to ethical constraints, it is difficult to acquire many samples of volunteers performing incorrect forms of exercises. Specifically, it is considered unethical to ask volunteers to perform exercises incorrectly when it may lead to increased risk of injury. To address that difficulty, in this research I collected my own personal data in which I exhibited both correct and incorrect forms. In addition, I collected samples from other volunteers who were asked to perform correct form, but who may have exhibited incorrect form unintentionally. My data was used to train the statistical model while the data from other volunteers was used to test the model.

I acted as an expert for the task of assigning class labels to exercise repetitions. The work of this thesis is a proof-of-concept for automatically recognizing performance errors, and thus the meaningfulness of the results does not depend on the particular errors chosen. In addition, access to resources such as personal trainers is expensive and I would like to develop a framework that does not rely on using a personal trainer. The most important aspect of the error classes is that they should exhibit distinct characteristics that differentiate each class from one another. Distinct classes lend themselves more easily to machine recognition. The error classes chosen do not necessarily represent universally accepted improper form. Instead, the error classes represent deviant performance techniques that do not necessarily increase risk of injury. Furthermore, the framework I establish strives to be as general as possible, and thus may be applied similarly by physical trainers or other parties with their own definitions of error classes. Thus the discussion remains useful regardless of the specific definition of performance errors.

I chose one error class for arm circles, one error class for arm curls, and 2 error classes for jumping jacks. Specifically, arm circles are classified as either correct or too

large. Arm curls are classified as either correct or the elbow is too far from the torso. Jumping jacks are classified as either correct, incorrect synchronization of arms and legs, or the hands are raised too low. Each exercise repetition receives a single error class label.

Figure 2. Jumping jacks: correct form

Figure 3. Jumping jacks: incorrect synchronization

Figure 4. Jumping jacks: low hands

Figure 5. Arm circles: correct form

Figure 6. Arm circles: large radius

Figure 7. Arm curls: correct form

Figure 8. Arm curls: elbow in front of torso

The Kinect provides access to 20 tracked joints. Figure 9, below, shows the joints overlayed onto a human body, and Figure 10 shows the hierarchical structure of the joint orientations. The images are taken from Microsoft's Developer Network documentation for natural user interfaces [MSDN 2014]. The joint orientations are structured such that each joint, except for the hip center joint, represents the rotation necessary to orient the bone from the parent to the child joint. The hip center represents the person's orientation relative to the Kinect sensor. Orientation rotations are represented as quaternions. Quaternions are a number system that extends the complex numbers. With the quaternion representation, there are 4 numerical components to each joint orientation, with each numerical component being a floating-point number between -1 and 1. Quaternions are an efficient way to store rotations, however they are hard to visualize because the familiar x, y, and z axes are coupled into the 4 quaternion components. The quaternion representation can be mapped into an 3 dimensional rotation matrix, however the extra computation yields little additional benefit.

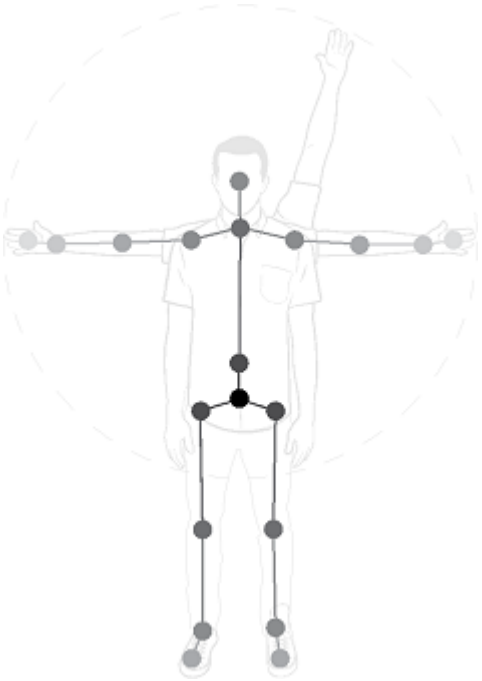

Figure 9. Overlay of tracked joints

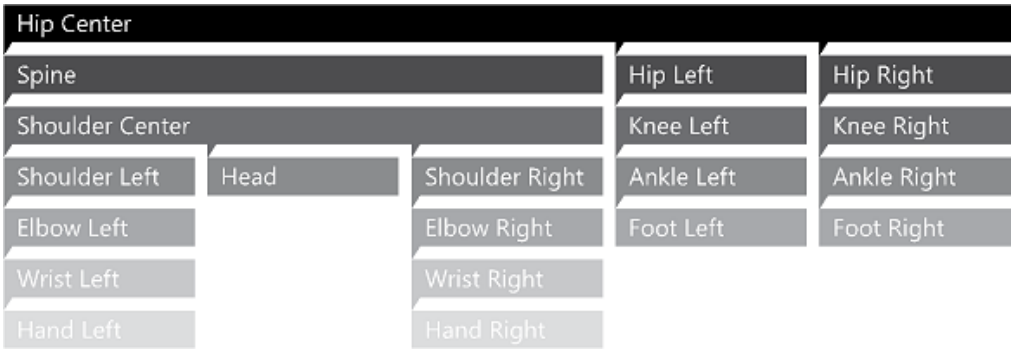

Figure 10. Hierarchical structure of joints

For features, I test with multiple subsets of the joint orientations. Joint orientations are appealing as features because they are invariant to lighting conditions (within the reasonable range in which the Kinect can track skeletons), invariant to projective distortion, and invariant to camera angle because they are hierarchical in relation to the skeleton. Furthermore, the quaternion representation is appealing because it is normalized within the range of -1 to 1, so no further data preprocessing is necessary. I incorporated another set of features that captures temporal information by

calculating the difference (delta) of joint orientations between adjacent frames. Ideally, the delta of orientations between frames would be divided by the time elapsed between those frames in order to get an orientation 'velocity' feature that is invariant to the data stream's frame-rate. However, the frame-rate of the Kinect is nearly constant, so this extra computation is not necessary.

I selected joints for the reduced set of features by choosing joints that yield the most discriminative information between classes. Intuitively, the orientations of bones that yield the most information are the bones from shoulders to elbows, elbows to wrists, hips to knees, and knees to ankles as they display the widest range of motion in each of the exercises. The wrist, elbow, ankle, and knee joints contain the information for the aforementioned bones, due to the hierarchical storage of orientation information. A bone's orientation is stored in the child joint according to the skeleton hierarchy. Furthermore, the error classes chosen are largely defined by the delta of these bone orientations between frames.

I provide here the rationale behind excluding certain features for the reduced set. I excluded hip center orientation because it is the root joint in the skeleton hierarchy. Thus, the hip center joint orientation is the only orientation that is absolute (its orientation is relative to the Kinect sensor rather than hierarchical relative to the other joints). For this reason, the hip center orientation is not invariant to the camera angle. I also excluded the left and right hip, lower spine, and right and left shoulder joints. These orientations are more person-dependent than exercise-dependent due to the fact that a person cannot noticeably extend their hips or shoulders, and thus these orientations are largely fixed for a given person. Finally, I excluded the neck, spine, hand, and feet orientations. While there is variation in the spine between exercises, and there may be error classes associated with the angle of the person's back, they are not relevant to any of the error classes that I examined.

There are some advantages and disadvantages to excluding features. One reason for exclusion is to increase speed of model training and of classification. Speeding up model training is not especially advantageous for this research, as this process occurs offline, and saving seconds or even minutes is not critical. On the other hand, speedup of classification is important, as this step is typically done in real-time for RVTs. However, even with inclusion of all orientations and all orientation deltas, the total feature count is 160 features (20 joints, 4 quaternions per joint, and their deltas), which is not inhibitively large. Therefore it may not be necessary to reduce the number of features for the sake of increasing speed. Another consideration is that reducing the number of features does not always result in higher classification accuracy, depending on the significance of the features chosen. The best features are those that lead to higher classification accuracy, even if the contribution is marginal, and as long as classification is within desirable computational bounds. It is good to eliminate noisy features and features that exhibit no discriminative quality between the error classes. An example is the left arm and leg for the arm curl exercise. The joints on the left arm and leg are irrelevant to the error classes for arm curls because the right arm is the only part of the body involved in the exercise. Inclusion of those features leads to degraded classification accuracy due to increased overlap in feature space between the classes.

For the model, I sought out model training and classification frameworks that can classify incorrect form based on only a few correct samples. One approach is to use single-class SVM (SC-SVM), which builds the model and estimates the surrounding feature space. SC-SVM is able to detect errors by their deviation from the correct model within the established feature space. However, SC-SVM may not be appropriate for identifying specific classes of errors. First, SC-SVM can only detect if a test sample is deviant from the correct form, that is, if the test sample is incorrect. SC-SVM cannot distinguish multiple error classes from one another. Second, SC-SVM cannot identify

specific joints or bones that may contribute to the cause of error. To achieve that level of diagnostic capability, the skeleton joints that are most deviant from the correct model must be identified. Another problem with SC-SVM is that the feature space is very large, while the range of human motion, which is limited by the musculoskeletal structure of the human body, is relatively small within the feature space. Thus, both correct and incorrect exercise forms will be relatively similar within the large feature space, and will be difficult to distinguish. To map out the effective feature space, data would need to be recorded of people bending and stretching in various ways in order to establish the range of human motion and incorporate that information into the model.

Another appealing model is Hidden Markov Model because of its success in human activity detection and recognition. HMM does well when the data is temporal and repetitive, as exercises are. However, HMM may not be warranted in the case of exercise performance error recognition. One significant challenge with applying HMM is the choice of discrete states to be used. Joint orientations are continuous so they don't lend themselves readily to HMM. Discretizing the continuous space is not straightforward because the choice of the optimal number of states is complex and has a significant impact on performance. Another option for states in HMM is to use paths generated from tracking certain joints, typically hands, through time. Hand paths work well for gesture recognition in general, but are not ideal in my case as there is much more information associated with each exercise beyond the path the hands take. I could have tracked paths of all joints, but I may have run into computational complexity as well as the problem of optimal discretization.

In the classification problem of this thesis, there is evidence that dynamic information across an entire exercise repetition is not necessary for accurate classification. Non-dynamic features of single frames combined with dynamic features between adjacent frames (deltas) may be sufficient for recognizing error classes. The

reason for this is that each class has a set of frames with unique skeletal structure that can be used to distinguish between classes. For example, jumping jacks in which the participant does not raise his or her hands high enough usually exhibit bent elbows, a structural signature unique to that error class. These sorts of unique skeletal structures can be used to determine which error classes are present given observed frames. Thus it is suitable to use statistical learning models such as K-nearest neighbor (KNN) and Support Vector Machine (SVM) that can be trained and classify per frame of stream data.

For models, I've elected to use multi-class SVM with a variety of kernels, as well as KNN. SVM is a supervised learning model that creates a linear barrier (a hyperplane) between classes such that there is the widest margin possible between class samples. When the classes are not separable, kernels can be applied to transform the feature space to make classes more separable for the SVM barrier. Alternatively, imperfect separation can be permitted if a penalty is applied for crossover data. Although SVM is a binary classifier, meaning it creates a barrier between 2 classes, it can be extended to handle 3 or more classes by creating discriminant barriers between each class and the others.

KNN on the other hand, uses a voting system to classify new observations. New observations are compared to samples from the training phase, and the new observation is assigned to the most popular class among the nearest samples in the feature space. These models are well supported in OpenCV, are computationally fast when the feature space is relatively small, and are popular and well known machine learning algorithms. One tradeoff is that both the SVM and KNN models do not readily capture the temporal nature of the data beyond the delta feature being extracted.

The kernels I tested with for SVM are linear (no kernel), sigmoid, and radial basis function. These kernels are well supported in OpenCV and are useful in most common

data distributions for discriminating between classes. KNN on the other hand is interesting for this classification problem because it can address the issue of multiple errors per sample. The voting scheme of KNN allows us to keep track of neighbors in a given class above a certain threshold. Thus if a sample has many neighbors of a certain class, the sample is likely to be in that class as well, even if that class doesn't receive the highest vote. Another reason KNN is interesting is because it performs reasonably fast when the number of training samples is low, as is the case for my data which has less than 1000 sample frames of data per class.

IV. METHODOLOGY

The development environment chosen was Visual Studio 2013, along with the Kinect for Windows Software Development Kit (SDK), and C++ as the programming language for all applications. Visual Studio and C++ were suitable candidates due to their compatibility with the Kinect SDK as well as OpenCV. The OpenCV open computer vision library was used extensively for its convenience with working with images, general matrix operations, and machine learning library tools. OpenCV was also appealing for its compatibility with C++ and Visual Studio 2013, and because it is open source so it is freely available for use.

Exercises were selected for this research to meet the following criteria: (i) the exercise is stationary and can be performed in front of a static Kinect, (ii) the exercise has a variety of ranges of motion with which to test the robustness of the classification system, (iii) the exercise has clearly defined errors or improper forms, and (iv) the exercise is not dangerous or associated with moderate or high risk of injury. Volunteers for performing the exercises were selected according the following criteria: (i) the volunteer is healthy according to the ACSM guidelines for exercise testing and prescription, (ii) the volunteer is willing to perform the exercises, (iii) volunteers have a basic knowledge of the exercises, (iv) enough volunteers were selected to exhibit random variation in athleticism and body type, and (v) volunteers are in the age range 18-30.

The data was collected in the following fashion. The Kinect was set up approximately 2 feet off the ground, facing toward the location where the participants stood, approximately 6 feet away. The room had moderate overhead lighting, with white walls as background. Clothing color varied between participants. Volunteers performed five repetitions of jumping jacks, arm circles, and arm curls. Volunteers were asked to

perform the exercises correctly. In addition, I collected data of myself performing each exercise. I performed 9 repetitions for each error class, consisting of 3 repetitions for each of 3 different angles: facing the camera, facing slightly right, and facing slightly left.

For saving the streamed data from the Kinect, I borrowed and modified source code from Dolatabadi [Dolatabadi, 2013]. The source code is free to download at http://kinectstreamsaver.codeplex.com/. The stream saver application is multithreaded so as to be concurrent with the incoming data from the sensor. The application allows RGB and depth data to be saved as either .jpg files or binary files, and skeleton data to be saved as binary files. The skeleton data saved includes joint position, frame time and count data, and hierarchical joint orientation data. Another program reads the binary data and stores the data in XML format. One disadvantage of this code is that it operates at a framerate of 15 frames per second rather than at the Kinect operating limit of 30 frames per second, meaning about half the frames from the Kinect stream were lost during the recording process.

I iterated through each frame of the raw data and spliced the data stream into distinct repetitions for each exercise. The end result is many collections of frames of data, with each collection consisting of frames that make up a single repetition. Each repetition belongs to an exercise, a participant, and an error class (see below). Some repetitions from the collected data were discarded and not used in training or testing, either because part of the sequence was cut-off at the start or end of the stream, or because the performance was anomalous. The final set of repetition data consisted of 30 jumping jacks, 37 arm circles, and 27 arm curls from other volunteers, and 27 jumping jacks, 27 arm circles, and 27 arm curls from my own recorded performance.

Each exercise repetition consists of a collection of ordered frames of data (in the order they were collected by the Kinect stream). Each exercise repetition received a single class label, which was assigned manually by myself. This means that each frame

of data, or sample, within the exercise repetition is given the same error class label. The class labels were stored in a CSV file with one class label per exercise repetition in collection order. To assign these per-repetition labels to each sample frame within repetitions, this was done dynamically by reading in each sample, and counting the number of samples per exercise repetition. The labels per-sample were stored in an OpenCV matrix, which was ultimately written to an XML file for later retrieval.

I analyzed the features by taking means and covariances of the joint orientations and joint orientation deltas across all samples, as well as means and covariances across samples within each class label. The class-specific means and variances were compared to one another to identify the features most discriminant between the classes. Programmatically, this was done using calcCovar() in OpenCV which produces a covariance matrix as well as an array of feature means. I chose to include feature joints that displayed significant discrimination between the classes. Features that had high variance among all samples, but did not have distinguishable means between the classes were considered noise or irrelevant features.

Support Vector Machine and K-Nearest Neighbor models were trained using OpenCV's 'ml' module classes. The OpenCV SVM implementation is based on LibSVM, an open-source SVM library created by C.-C. Chang and C.-J. Lin [Chang 2011]. I used the following parameters for SVM. For svm_type, CvSVM::C_SVC was used, indicating n-class classification with imperfect separation of classes allowed with penalty multiplier c for outliers (the default value of 1 was used for c). For kernel_type, I experimented with CvSVM::LINEAR, indicating no kernel transformation of feature space, as well as CvSVM::SIGMOID and CvSVM::RBF, which correspond to sigmoid kernel and radial basis function kernel, respectively. Additionally, the termination criteria for SVM training was set to a maximum of 100 iterations with required termination accuracy of 10^{-7}.

For KNN, the non-regression implementation supported in OpenCV was used. The implementation caches all training samples and performs classification by a voting system of the K most similar sample responses. I experimented with the value of k = 10. For each input vector, neighbors are sorted by their distances to the vector. This means that in the case of a tie, The closer vectors will win the vote.

V. EVALUATION

A performance error recognition system is successful if it is able to achieve high rates of classification accuracy for a significant set of exercise data. In this work I collected new exercise performance data with the Kinect from volunteers and from myself. The models are evaluated by their classification accuracies per frame of data (as opposed to per exercise repetition, where each repetition contains a sequence of data frames). Evaluation of the models and features was done in the following fashion. One separate model was trained for each exercise type. Each model was trained using only data collected from a single volunteer (myself), while the model was tested on the data from the remaining volunteers. This is a realistic scenario for an RVT system in which the system is likely to be developed by a small team but used by many. This method reduced biases that could occur if frame samples are trained and tested from the same volunteers, as is the case with a random sampling approach. For each frame of data in the test set, its output response from the model was compared to the expert labels to determine classification accuracy. Each model was trained and tested to obtain per-class and overall recognition rates in confusion matrices. The recognition rates obtained for each model were used to evaluate the effectiveness of each model.

I provide here the distribution of correct and incorrect classifications for each exercise. The samples which form the training set, contain exercise repetitions from a single volunteer, and there is equal representation for each error class. That is, for jumping jacks, 33.3% of the samples are correct form, 33.3% are the wrong rhythm, and 33.3% have the hands raised too low. For arm circle, 50% are correct form and 50% are too large. For arm curl, 50% are correct form and 50% have the elbow too far forward. For the test set, the same data was used as for the first test approach. That is, for

jumping jack, 59.4% of the samples are correct form. For arm circles, 46.0% are correct form. For arm curls 64.2% are correct form.

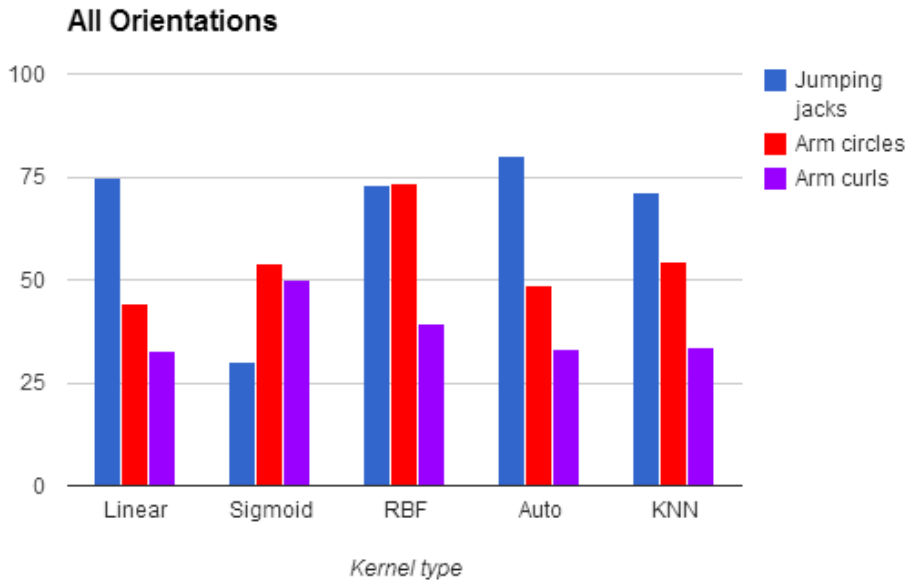

Figure 11. Total accuracies using all orientations

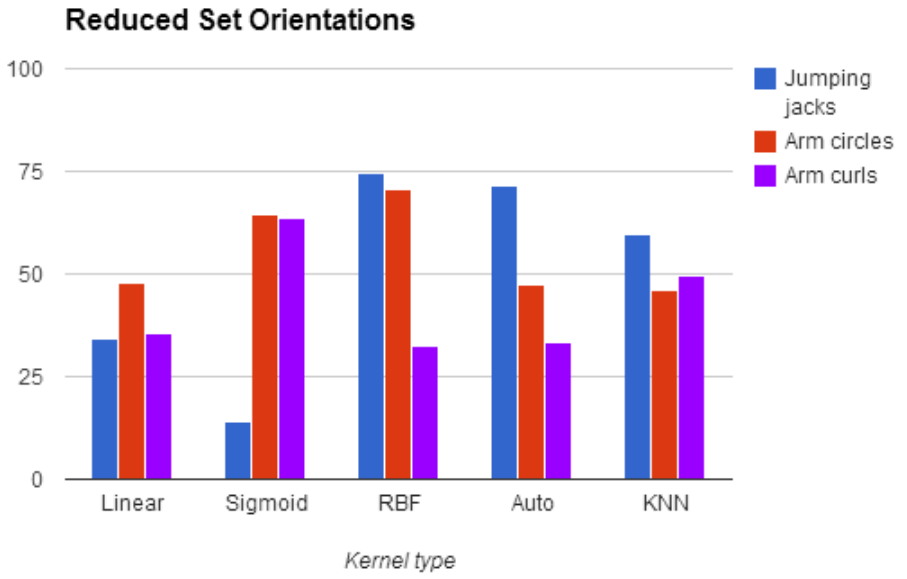

Figure 12. Total accuracies using reduced set of orientations

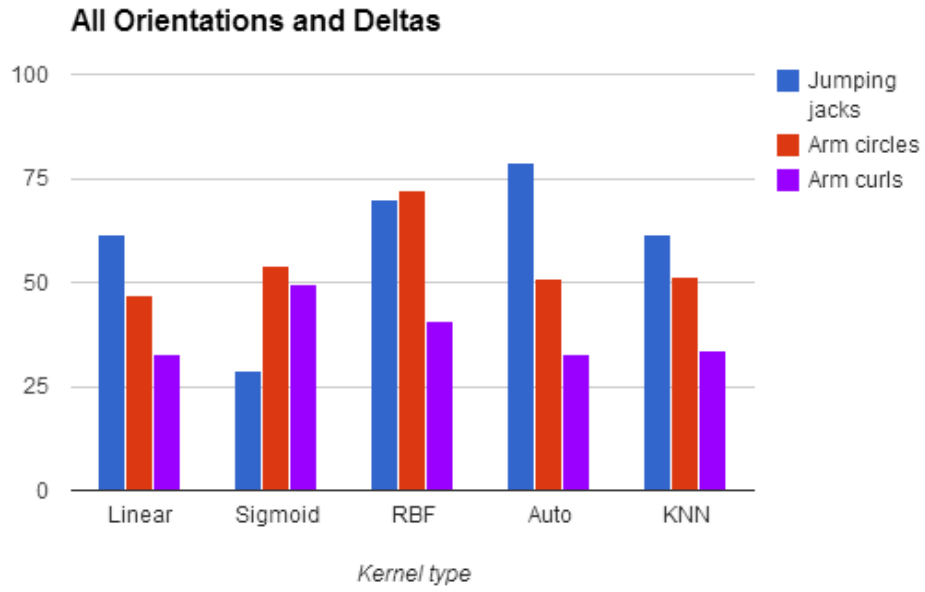

Figure 13. Total accuracies using all orientations and deltas

There are a number of things to note from these results. First, SVM with radial basis function as the kernel tended to have the best overall performance, getting near 75% total accuracy for jumping jacks and arm circles, while doing much worse for arm curls at around 35%. The confusion matrix obtained for this model and feature set is shown in Figure 14, illustrating typical per-class accuracies. The true class labels are given by the row index, while the response labels are given by the column index. For jumping jacks, class 0 represents correct form, class 1 represents incorrect rhythm, and class 2 represents arms raised too low. For arm circle, class 0 represents correct form while class 1 represents too large. For arm curl, class 0 represents correct form while class 1 represents elbows too far from the body.

	0	*1*	*2*
0	225	32	61
1	19	43	0
2	30	2	123

Jumping J. 73.1%

	0	*1*
0	360	23
1	199	250

Arm circle 73.3%

	0	*1*
0	278	187
1	380	89

Arm curl 39.3%

Figure 14. SVM with radial basis function kernel, all orientations

Second, SVM with the Auto training method did particularly well for jumping jacks, reaching 80%, although doing much worse for the other two exercises. Third, nearly all the models and feature sets performed poorly for classifying arm curls, with most being between 30% and 40%, with one exception. SVM with a sigmoid kernel achieved up to 63.7% for arm curls using the reduced set of features, doing equally well for arm circles and drastically worse for jumping jacks at less than 15%. Fourth, KNN did not perform especially well, getting around 50% for all three exercises across the 3 different feature sets. KNN's numbers are deceptively high, as the confusion matrices reveal a strategy of guessing that every frame belongs to the same class as is shown in figure 15. Even with this poor strategy, KNN will guess correctly on about half of the test samples.

	0	*1*	*2*
0	318	0	0
1	62	0	0
2	155	0	0

Jumping J. 59.4%

	0	*1*
0	383	0
1	449	0

Arm circle 46.0%

	0	*1*
0	465	0
1	469	0

Arm curl 49.8%

Figure 15. KNN, reduced set orientations

Fifth, the reduced set of features which used 10 joints instead of 20, performed worse when used in conjunction with some kernels and better for others when compared with using all joint orientations and using all joint orientations and their deltas. Specifically, the reduced set did worse for linear kernel and better for sigmoid kernel. The incentive behind the reduced set of features was to eliminate the noisy, insignificant joints in order to build a stronger classifier. However, it appears that the eliminated joints contained enough information in most cases to marginally improve classification. Finally, using all orientations and all deltas as features performed slightly worse for every kernel as well as KNN when compared with only using orientations without the deltas. This indicates that the delta information is either irrelevant or inaccurate. A likely explanation is that the quaternion representation for joint orientations does not lend itself to meaningful deltas.

The results of this evaluation show much variability in performance across the different models and features used, as well as across each of the exercises. This may indicate that the error classes are clustered differently for the different exercises, such that a single model may not be able to discriminate the data in the same fashion for two different exercises. Another explanation is that there may be significant overlap between error classes in feature space. As evidence for this, consider the jumping jacks classes in which the low-hands error class has a shorter range of motion for the shoulder joint, but both the correct class and the low-arms class have similar joint orientations for the middle portion of the repetition, when the arms have not reached their minimum point or their peak. This overlap makes it difficult to separate the classes by entire repetitions. The error classes also may not be very distinct inherently. Since the error classes were chosen after the data was collected in order to keep the data collection process natural and not force volunteers to perform predefined errors, the chosen errors had to distinguish the exercise repetitions as they were. Since most of the data collected was of

fairly good form, especially for arm curl which showed only very slight differences in performance, the chosen error classes were not especially distinguishable to begin with.

For arm curls in particular, performance is expected to be the worst of the three exercises for a number of reasons. First, the tracking of the left arm and left leg joints is highly degraded. Because the left arm and leg is occluded by the rest of the body when the volunteer is turned sideways, those joints are inferred by the Kinect and there is little surrounding-joint context so those inferences are poor. As a result, those joint's positions and orientations are highly noisy, often appearing to jump around rapidly and randomly when visualized. Second, arm curls exhibit the least variation between correct and incorrect classes, making it more difficult to discriminate. Third, arm curls are the slowest of the three exercises resulting in the most frames of data per repetition and thus more overlap between classes and possibly overfitting of the model. I expected arm curls to be significantly independent of temporal features (the delta of joint orientation between frames), as the elbow error class was primarily based on elbow position relative to the body (specifically, the orientation of the shoulder), rather than on its dynamic motion. An additional test was done for arm curl in which only features from the right arm were used. Using only right arm features for arm curl with SVM and radial basis function for the kernel yielded total accuracy of 45.3%. This, intuitively, is an improvement because most of the information for arm curl is in those joints. However, 45.3% is still a poor result overall.

VI. CONCLUSION

In this research I systematically collected skeleton tracking data from Microsoft Kinect of volunteers performing jumping jacks, arm circles, and arm curls. I labeled the exercise repetitions by hand, and used those labels to test the effectiveness of various statistical models and features in the task of recognizing performance error classes.

In conclusion, the methods used in this research have proven to be effective in recognizing specific performance errors for jumping jacks and arm circles, while being less effective for arm curls. I have achieved satisfactory classification for jumping jacks and arm circles using a single model, however arm curls required a different model and achieved slightly worse results overall. These results show that recognition of specific, predefined error classes is possible, and these results work towards enhancing the capabilities of Reactive Virtual Trainers. The work of this thesis can be extended to be included in an RVT system with minor modifications, such as making the classification real-time.

One goal of this research was to develop a generalizable framework that would perform well for any kind of exercises and error classes respecting the limitations of the interaction space available to the Kinect, or at least a framework in which researchers can efficiently and straightforwardly choose a proper model to use for each case. However, because I was not able to find a single model and set of features that performed well for all exercises selected in this research, the framework may not be an effective generalizable solution to all exercises and error classes. Machine learning in general is highly data dependent, so it is a difficult problem to solve.

One possible explanation is that each exercise has a unique distribution in feature space, with the distributions of each exercise, and each error class being different. Thus, a model that performs well for one exercise may not be appropriate for

another. In more specific terms, a model using one kernel may not be able to separate classes equally well. For example, SVM with radial basis function as the kernel performed fairly well for jumping jacks and arm circles, but did poorly for arm curls, while SVM with sigmoid kernel did much better for arm curl and much worse for the other two.

The errors I've defined and chosen to classify in this work can be thought of as variations of correct form due to the subjectivity of labeling samples as correct or incorrect. The result is that the classes I'm discriminating are quite similar to one another. Therefore, it is reasonable to think that the classification rates I've achieved given the state of the data is promising. Additional data in which error classes are more distinct (quite possible in a real RVT system) is likely to result in even higher classification accuracy.

Furthermore, due to the limited data collected, in which the data came from very few participants and from the same collection environment, there is a difficult balance between using enough of the data to build a successful model and having enough unique samples to do unbiased testing of the model. There is a representation bias between correct and incorrect samples. An improved data set would involve more participants, as well as a varied collection environment including different recording angles and distances.

VII. FUTURE WORK

One avenue of future research is to explore additional exercises and error classes. There are many varieties of errors that can occur in any given exercise. For example, investigation of hand grip for arm curls may reveal another possible error class. Of course, expanding the features used to include hand joints would be necessary.

Another consideration worth exploring is the possible presence of multiple error classes simultaneously in a single sample. For classification under these circumstances, researchers may decide to label the sample with all error classes that exceed a certain probability threshold. One possible addition to my framework could be to include a similarity metric along with a threshold, thus transforming the problem to a regression problem rather than a classification problem. This alternate framework would also allow easier application of a cost function that could, for example, punish classifiers for misclassification of erroneous form as correct form more heavily than the other way around. Alternatively, researchers could duplicate the samples that have both error classes in the training set, and label those samples as one of each of the two error classes. The problem with this approach is that, if using the statistical models used in this research, only one error class may be dominant even while there may be overlap between the error classes.

Another avenue of future research is to take the results obtained thus far and assess how they may contribute to the development of a whole RVT system. Included in this assessment could be how the RVT can further motivate users to exercise more and continue exercising, as well as comparing the effectiveness of the RVT versus a physical trainer.

Another area to explore is the use of different models and classification methods. One possible way to achieve better results would be to classify errors per frame of data. However that is very tedious and time expensive and prone to human error. Similarly, I could break each exercise repetition into fractions, such as thirds, and divide the class labels further into new categories such as Correct 1st third, Correct second 3rd, Correct final third, Incorrect 1st third, and so on. This approach would integrate more temporal information into the model. Of course, this approach brings new considerations such as the optimal fraction to divide each repetition by, and whether or not the optimal fraction varies for different exercises or for different error classes.

This thesis demonstrated the effectiveness of classifying per frame of exercise stream data. However, improved performance may be achieved by applying a meta-classifier. Such a classifier would store and review the per-frame classifications from the previous X number of frames, and deduce an enveloping classification for those frames. This approach is promising because there is overlap in structure between error classes, meaning classification per frame is not the most direct route to classifying entire repetitions. It is possible, with this approach, to classify entire repetitions with 100% accuracy even while classifying individual frames with only 75% accuracy. For example, if a regular occurrence of error frames are observed, the system can be reasonably confident in the true presence of consistent incorrect form. Additional considerations arise such as choosing the optimal number of frames to review with the meta-classifier

This thesis showed that moderate to satisfactory classification accuracy can be achieved using my methods. One possible next step is to integrate these classification methods into a real-time system, enabling real-time feedback to users performing exercises. That step is reasonable, as offline classification was shown to be fast. Given that a real-time system must also process the incoming data stream and extract skeleton information, the total computation will be more expensive. The additional real-time

processing would not be expected to significantly degrade the classification frame rate. Furthermore, as hardware becomes more efficient, the problem will become increasingly tractable.

There are other challenges to successfully implementing real-time classification. Unlike with my organized sample data, in a real-time data stream, it is not readily known in which frame the human performer begins the exercise repetition. Extra recognition capabilities would need to be developed in order to identify the beginning and end of exercises.

Kinect for Windows v2 will be released in Summer 2014. Version 2 features significantly improved specifications, including 1080p HD video, expanded field of view, improved skeletal tracking with up to 6 full skeletons of 25 joints tracked per skeleton, improved joint orientation and active infrared detection providing better tracking even in low-light environments. I anticipate that much of the work done in this thesis could benefit from the improvements of Kinect Version 2 as well. Using Version 2 would likely result in higher classification rates due to the improved skeleton tracking, and could also benefit from the increased physical interaction space.

REFERENCES

[Ahlskog, 2011] Ahlskog, J. Eric, et al. "Physical exercise as a preventive or disease-modifying treatment of dementia and brain aging." *Mayo Clinic Proceedings*. Vol. 86. No. 9. Elsevier, 2011.

[Bierla, 2012] Bierla, Kathleen, and Eric Balaban. "Xbox Kinect training may improve balance measures in older adults." *JOURNAL OF AGING AND PHYSICAL ACTIVITY*. Vol. 20. 1607 N MARKET ST, PO BOX 5076, CHAMPAIGN, IL 61820-2200 USA: HUMAN KINETICS PUBL INC, 2012.

[Bonnechère, 2014] Bonnechère, B, B Jansen, P Salvia, H Bouzahouene, L Omelina, F Moiseev, V Sholukha, J Cornelis, M Rooze, and S Van Sint Jan. "Validity and Reliability of the Kinect Within Functional Assessment Activities: Comparison with Standard Stereophotogrammetry." *Gait & Posture*, 39.1 (2014): 593-598.

[Chang, 2011] Chang, C.-C. and C.-J. Lin. LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>.

[Chang, 2012] Chang, Chien-Yen, Belinda Lange, Mi Zhang, Sebastian Koenig, Phil Requejo, Noom Soomboon, Alexander A. Sawchuk, and Albert A. Rizzo. "Towards Pervasive Physical Rehabilitation Using Microsoft Kinect." *The 6th International Conference on Pervasive Computing Technologies for Healthcare* (2012): n. pag.

[Charles, 2011] Charles, James, and Mark Everingham. "Learning shape models for monocular human pose estimation from the Microsoft Xbox Kinect." *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011.

[Cooney, 2013] Cooney, Gary M., et al. "Exercise for depression." *Cochrane Database Syst Rev* 9 (2013).

[Dolatabadi, 2013] Dolatabadi, Elham, Babak Taati, Gemma S. Parra-Dominguez, Alex Mihailidis, “A markerless motion tracking approach to understand changes in gait and balance: A Case Study”, the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA 2013 Annual Conference). Student Scientific Paper Competition Winner. <http://kinectstreamsaver.codeplex.com/>.

[Ferreira, 2014] Ferreira, Rita, et al. "Lifelong exercise training modulates cardiac mitochondrial phosphoproteome in rats." *Journal of proteome research* 13.4 (2014): 2045-2055.

[Hunter, 2010] Hunter, Gary R., et al. "Exercise training prevents regain of visceral fat for 1 year following weight loss." *Obesity* 18.4 (2010): 690-695.

[Ikizler, 2008] Ikizler, Nazlı, and David A. Forsyth. "Searching for Complex Human Activities With No Visual Examples." *International Journal of Computer Vision* 80.3 (2008): 337-57. 28 Apr. 2008. Web. 19 June 2014.

[Kar, 2010] Kar, Abhishek. "Skeletal tracking using microsoft kinect." *Methodology* 1 (2010): 1-11. <http://www.cs.berkeley.edu/~akar/cs397/Skeletal%20Tracking%20Using%20Microsoft%20Kinect.pdf>.

[Kayama, 2014] Kayama, Hiroki, Kazuya Okamoto, Shu Nishiguchi, Minoru Yamada, Tomohiro Kuroda, and Tomoki Aoyama. "Effect of a Kinect-based Exercise Game on Improving Executive Cognitive Performance in Community-dwelling Elderly: Case Control Study." *Journal of Medical Internet Research*, 16.2 (2014): e61-e66.

[Koch, 2011] Koch, Lauren Gerard, et al. "Intrinsic aerobic capacity sets a divide for aging and longevity." *Circulation research* 109.10 (2011): 1162-1172.

[Konečný, 2013] Konečný, Jakub, and Michal Hagara. "One-Shot-Learning Gesture Recognition using HOG-HOF Features." *arXiv preprint arXiv:1312.4190* (2013).

[Leavitt, 2013] Leavitt, V. M., et al. "Aerobic exercise increases hippocampal volume and improves memory in multiple sclerosis: Preliminary findings." *Neurocase* ahead-of-print (2013): 1-3.

[Maccormick, 2014] Maccormick, John. "How Does the Kinect Work?" (n.d.): n. pag. *Selected Talks by John MacCormick*. Dickinson College, 6 Sept. 2011. Web. 19 June 2014. <http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf>.

[McCullough, 2012] McCullough, Lauren E., et al. "Fat or fit: the joint effects of physical activity, weight gain, and body size on breast cancer risk." *Cancer* 118.19 (2012): 4860-4868.

[MSDN, 2014] Microsoft Developer Network. Natural user interface, skeletal tracking, joint orientations documentation. (2014). <http://msdn.microsoft.com/en-us/library/hh973073.aspx>

[Nergui, 2013] Nergui, M, Y Yoshida, N Imamoglu, J Gonzalez, M Sekine, and Wenwei Yu. "Human Motion Tracking and Recognition Using HMM by a Mobile Robot."*International Journal of Intelligent Unmanned Systems*, 1.1 (2013): 76-92.

[Ogden, 2012] Ogden, Cynthia L, Margaret D. Carroll, M.S.P.H.; Brian K. Kit, M.D., M.P.H.; and Katherine M. Flegal, Ph.D., "Prevalence of Obesity in the United States, 2009-2010", *NCHS Data Brief*, No. 82, January 2012, <http://www.cdc.gov/nchs/data/databriefs/db82.pdf>.

[Oikonomidis, 2011] Oikonomidis, Iason, Nikolaos Kyriazis, and Antonis A. Argyros. "Efficient model-based 3D tracking of hand articulations using Kinect." *BMVC*. Vol. 1. No. 2. 2011.

[OSG, 2010] Office of the Surgeon General, "The Surgeon General's Vision for a Healthy and Fit Nation." Rockville, MD, U.S. Department of Health and Human Services; 2010. <http://www.surgeongeneral.gov/initiatives/healthy-fit-nation/index.html>.

[Pollack, 2003] Pollack, Martha E., et al. "Autominder: An intelligent cognitive orthotic system for people with memory impairment." *Robotics and Autonomous Systems* 44.3 (2003): 273-282.

[Raheja, 2011] Raheja, Jagdish L., Ankit Chaudhary, and Kunal Singal. "Tracking of fingertips and centers of palm using kinect." *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*. IEEE, 2011.

[Rantz, 2013] Rantz, MJ, M Skubic, C Abbott, C Galambos, Y Pak, DKC Ho, EE Stone, LY Rui, J Back, and SJ Miller. "In-Home Fall Risk Assessment and Detection Sensor System." *Journal of Gerontological Nursing*, 39.7 (2013): 18-22.

[Reyes, 2013] Reyes, M, A Clapes, J Ramirez, JR Revilla, and S Escalera. "Automatic Digital Biometry Analysis Based on Depth Maps." *Computers in Industry*, 64.9 (2013): 1316-1325.

[Ruttkay, 2006] Ruttkay, Z, J Zwiers, H van Welbergen, D Reidsma, J Gratch, M Young, R Aylett, D Ballin, and P Oliver. "Towards a Reactive Virtual Trainer." *Intelligent Virtual Agents. 6th International Conference, IVA 2006. Proceedings (Lecture Notes in Artificial Intelligence Vol.4133)*, 4133.2006 (2006): 292-303.

[Sallis, 1992] Sallis JF, Simons-Morton BG, Stone EJ, Corbin CB, Epstein LH, Faucette N, Iannotti RJ, Killen JD, Klesges RC, Petray CK, et al., "Determinants of physical activity and interventions in youth", *Med Sci Sports Exerc.* 1992 June, 24(6 Suppl):S248-57.

[Shin, 2014] Shin, Joon-Ho, Hokyoung Ryu, and Seong Ho Jang. "A Task-specific Interactive Game-based Virtual Reality Rehabilitation System for Patients with Stroke: A Usability Test and Two Clinical Experiments." *Journal of Neuroengineering and Rehabilitation*, 11.1 (2014): 32.

[Solomon, 2013] Solomon, Thomas PJ, et al. "The influence of hyperglycemia on the therapeutic effect of exercise on glycemic control in patients with type 2 diabetes mellitus."*JAMA internal medicine* 173.19 (2013): 1834-1836.

[Starner, 1995] Starner, Thad E. *Visual Recognition of American Sign Language Using Hidden Markov Models*. MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF BRAIN AND COGNITIVE SCIENCES, 1995.

[Tang, 2014] Tang, Matthew. "Recognizing Hand Gestures with Microsoft's Kinect." (n.d.): n. pag. *Recognizing Hand Gestures with Microsoft's Kinect*. Michael Heydt. Web. 19 June 2014. <http://www.scribd.com/doc/68532986/Recognizing-Hand-Gestures-with-Microsoft-s-Kinect>.

[Trabelsi, 2013] Trabelsi, Dorra, Samer Mohammed, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. "An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression." *IEEE Transactions on Automation Science and Engineering*, 10.3 (2013): 829-835.

[van Kasteren, 2011] van Kasteren, T. L. M., Gwenn Englebienne, and B. J. A. Kröse. "Human activity recognition from wireless sensor network data: Benchmark and software." *Activity Recognition in Pervasive Intelligent Environments*. Atlantis Press, 2011. 165-186.

[Wallace, 2014] Wallace, Ricky, et al. "Effects of a 12-week community exercise programme on older people: Nurses should promote exercise to reduce patients' social isolation and increase their independence, say Ricky Wallace and colleagues."*Nursing older people* 26.1 (2014): 20-26.

[Wan, 2013] Wan, Jun, et al. "One-shot learning gesture recognition from RGB-D data using bag of features." *The Journal of Machine Learning Research* 14.1 (2013): 2549-2582.

[Weber, 2012] Weber, Markus, Gabriele Bleser, Marcus Liwicki, and Didier Stricker. "Unsupervised Motion Pattern Learning for Motion Segmentation." *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, (2012): 202-205.

[Wei, 2012] Wei, Xiaolin, Peizhao Zhang, and Jinxiang Chai. "Accurate Realtime Full-body Motion Capture Using a Single Depth Camera." *ACM Transactions on Graphics (TOG)*, 31.6 (2012): 1-12.

[WHS, 2012] World Health Statistics 2012, <http://www.who.int/gho/publications/world_health_statistics/2012/en/>.

[Zhang, 2013] Zhang, Jane. "Using Virtual Reality Technology to Promote Healthy Lifestyle." Electrical Engineering, Cal Poly, San Luis Obispo. 2013.