FUSING JOINT INFORMATION FROM MULTIPLE KINECT CAMERAS TO DETECT ERRORS IN EXERCISES

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Sriharsha Vankamamidi

Dec, 2016

FUSING JOINT INFORMATION FROM MULTIPLE KINECT CAMERAS TO DETECT ERRORS IN EXERCISES

Sriharsha Vankamamidi

Thesis

Approved:

Advisor
Dr. Shivakumar Sastry

Co-Advisor
Dr. Forrest Sheng Bao

Committee Member
Dr. Nghi Tran

Committee Member
Dr. Jin Kocsis

Accepted:

Interim Department Chair
Dr. Joan Carletta

Interim Dean of the College
Dr. Donald J. Visco

Dean of the Graduate School
Dr. Chand K. Midha

Date

ABSTRACT

The Kinect camera is a versatile tool that is effectively used to recognize and correct errors in exercise performance using the 3D joint location data. In several exercises that involve complex sequences of motion, a single camera cannot track all joint locations because of occlusions. A natural solution is to utilize multiple cameras. However, two challenges must be addressed before multiple cameras are used. First, although the Kinect cameras are supposed to collect data at 30 fps, there is variability in the period of each camera. Second, each camera uses its center as its frame of reference for collection of skeletal joint data. A novel approach is proposed to address these challenges. Interpolation is used to sample data at a constant rate of 10 fps, to address the variability in frequency. Singular Value Decomposition is used to determine the rotation and translation parameters from each cameras frame of reference to a global reference frame. The results demonstrate the effectiveness of the approach.

ACKNOWLEDGEMENTS

First and foremost I must thank my advisor Dr. Shivakumar Sastry for giving me the opportunity to work on this project which is supported in part by the *National Science Foundation* under grant IIS-1237069 (to Dr. S. Sastry). He has shown a keen interest in my project right from the beginning and supported me in many ways in the various ways.

Then I would like to thank all my professors, staff members and fellow students in the University of Akron, for supporting me through out my Masters program. Finally I thank my loving parents for motivating me to make this a success.

TABLE OF CONTENTS

Page

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER

I. INTRODUCTION 1

II. BACKGROUND 5

2.1 Action Recognition using Joint Information 5

2.2 Kinect Camera Applications 5

2.3 Multiple Kinect Cameras 6

2.4 Singular Value Decomposition 8

III. APPROACH 10

3.1 Common Reference for 3D Joint Coordinates Data 11

3.2 Common Reference Time 14

3.3 Compensating for Jitter 15

3.4 Recognizing Exercises and Detecting Errors 18

IV. RESULTS 20

4.1 Experiment Setup 20

4.2 Selecting Interpolation Method 22

4.3 Without Interpolation 23

4.4 Transforming frame of reference 23

4.5 Kinect Data Fusion 25

V. DISCUSSION 29

VI. CONCLUSION 30

BIBLIOGRAPHY 31

LIST OF TABLES

Table Page

4.1 Cubic spline interpolation results in consistently lower error than linear interpolation. 28

LIST OF FIGURES

Figure Page

3.1 The 3D Joint coordinates from each camera are provided by considering the location of the camera as the origin. When data for the same exercise are collected using multiple cameras, it is necessary to translate and/or rotate the coordinates from one camera to the frame of reference of the other camera. 12

3.2 Although the sampling interval in a Kinect camera was expected to be around 33 ms, this interval was observed to vary from 27 ms to 42 ms as illustrated in the figure. Notice that the variability differs from one camera to another and also varies depending on the environment conditions. 16

3.3 To compensate for the jitter in the sampling interval, all the data were interpolated to a 10 ms time base. This figure illustrates linear interpolation. The sampled joints are illustrated as $\mathbf{x_i}$. The trajectory of the joint between two successive sample points was assumed to be linear. 17

4.1 Experimental Setup with three cameras. 21

4.2 Cubic spline interpolation results in consistently lower error than linear interpolation. 22

4.3 Comparison of transformation error with interpolated data vs non interpolated data 23

4.4 The average tracking index for the joints from Kinect 2 (Sagittal plane) is lower than that for Kinect 1 (Frontal plane). 24

4.5 Time Series data of the Left Ankle and Right Ankle from Kinect 1 (Frontal plane). 25

4.6 Time Series data of the Left Ankle and Right Ankle from Kinect 2 (Sagittal plane). 26

4.7 Time Series Plot of the transformed data for the Left Ankle collected using Kinect 2 and the Right Ankle collected from Kinect 1. 27

4.8 Comparison of transformation error of fused joint information with individual transformed joint coordinate information. 27

CHAPTER I

INTRODUCTION

Recognizing and analyzing human motion is essential for a broad spectrum of applications such as wellness management, sports training, rehabilitation, surveillance and assistive technologies to improve the quality of life [1, 2, 14, 36, 43]. Basing interest in the context of empowering personalized wellness management (PWM). The objective of the thesis is to recognize errors that may occur when performing exercises; in particular, the main interest is in detecting those mistakes that can lead to injury without requiring an expensive personal trainer. Such a solution is important because it will improve the self-efficacy and adherence of the participants. It is also important to provide real-time feedback instead of offering a post-exercise evaluation of the performance. Toward this end, the thesis will explore what can be achieved by analyzing the 3D skeletal joint data that was gathered using a non-invasive sensor such as the Microsoft Kinect camera [44].

The skeletal joint tracking algorithm supported for the Kinect 2 camera [45] provides time-series data of joint locations. Standard algorithms have been utilized to extract skeletal joint data in the literature [45, 53]. When an exercise involves a complex sequence of motions, some of the joints can be occluded by other parts of the body. Consequently, the estimates of the joint positions can be inaccurate. A

tracking index indicates the confidence in the estimate of a joint position. Without accurate estimates of the joint positions, it is not possible to accurately detect the exercises or analyze the motions to identify errors in exercise performance. A natural solution to this problem is to utilize multiple Kinect cameras to gather the joint location data of a participant.

Two challenges must be addressed before multiple Kinect cameras can be used to gather joint location data. First, although the Kinect camera is supposed to provide 30 frames per second, there is considerable variability in the period of the data. For example, while the period is expected to be 33.33 m/s, measure periods of 27 m/s and up to 37 m/s are observed. Thus, when multiple cameras are used, data from a pair of cameras is likely not to be synchronized. Next, the data from each camera is provided by considering the position of the camera as the origin. Thus, the location and orientation of each camera are implicitly embedded in the data. Classical methods should therefore be utilized to address these challenges.

A *Singular Value Decomposition* (SVD) of a real or complex matrix, M, of order $m \times n$ is a factorization of M into the form $U\Sigma V^\top$ where U is $m \times m$ real or complex unitary matrix, Σ is a $m \times n$ rectangular diagonal matrix with non negative real numbers on the diagonal, and V is an $n \times n$ real or complex unitary matrix. The diagonal entries σ_i of Σ are known as the *singular values* of M. The columns of U and V are called the left-singular vectors and right-singular vectors of M, respectively. SVD has been used for point set registration to find optimal rotational and translational parameters between two point sets. The approach minimizes the least

squares registration error. By viewing the joint data from each camera, in each frame, as a point set, SVD is applied to find rotational and translation parameters. This approach is viable because the IR sensors used in the Kinect camera provide more accurate estimates of the common locations than traditional image processing techniques [16]. Further, since the joint estimates from each camera are already tagged, complex image processing techniques are not utilized to register corresponding joints. Basing on our results from the thesis, this approach is useful to fuse the information from multiple cameras into a single frame of reference. In the fused data set, estimate of a joint from the camera that has the highest values of the tracking index is considered as tracked and this break ties arbitrarily.

The main contribution of this thesis is the effective method to fuse data from multiple cameras into a single frame of reference and bringing them together at one point. Methods similar to our prior work [30, 41] need to be considered also, to recognize the exercise using a Support Vector Machine [46, 47] and recognize errors.

As described in [30, 41], the variety of errors and the variability in exercise performance from one individual to another dramatically increases the volume of training data that is necessary. Further, novice participants tend to make more mistakes in exercise, and it is difficult to get novices to make mistakes predictably so that so that a training dataset with labels can be assembled. Training data assembled by having experts making errors deliberately is unlikely to be useful for detecting and classifying errors made by novices. For these reasons, classifier-based approaches cannot be readily applied to address the problem of recognizing errors in exercise

performance. Following our approach in [30, 41], geometric characterizations based on the joint trajectories are used to recognize the errors using the fused data. The effective fusion of joint data from multiple cameras allowed us to extend the range of exercises reported in [30, 41]. Using the methods in this thesis, it is easy to recognize all the exercises in the High-Intensity Circuit Training [31]. Errors in these exercises are recognized as identified through discussions with domain experts in exercise science. While other works in the literature address the general problem of human action recognition [48, 49], the study focuses squarely on recognizing and detecting errors in the HICT suite.

The remainder of the thesis is organized as follows: Chapter 2 presents background information. There will be a description on the approach to fusing data from multiple views and recognizing exercises in Chapter 3. Experimental results that demonstrate the effectiveness of these approaches are also discussed in Chapter 4. After a discussion of the results in Chapter 5, conclusions and next steps are presented in Chapter 6.

CHAPTER II

BACKGROUND

The literature review part of the thesis explains and proves the main agenda of the thesis. It is related to the Kinect camera and other applications where multiple Kinect cameras apply. A description of the methods used for human action recognition and a few applications of Singular Value Decomposition are also discussed.

2.1 Action Recognition using Joint Information

The data obtained from a Kinect camera have been used in *Joint-based* [13, 28, 33, 40, 52] and *Part-based* approaches [12]. Some of the analysis techniques that are used in the part-based approaches include computation of 3D joint angles [40], Principal Component Analysis (PCA), Fourier Temporal Pyramid (FTP) [49] and Dynamic Time Warping (DTW).

2.2 Kinect Camera Applications

The Microsoft Kinect Camera has been extensively used to study and analyze human motion in the recent years [18, 24, 26, 51]. In [4] the authors evaluate the performance of dancers by comparing against a reference standard; visual feedback is provided to the participants. In [17, 35, 39], the authors discuss the role of Kinect camera in

rehabilitation. In [7], the authors present an exercise feedback system that uses a classifier to recognize the exercises; the performance is compared to a reference to provide real-time guidance and feedback in a tele-rehabilitation system. In [42], the authors estimate the anthropometry for participants by analyzing the data obtained from a Kinect camera. A good survey of the current research trends using the Kinect camera can be found in [25] and a detailed description of techniques used in Kinect camera are described in [44].

One of novelties of the Kinect camera is that there is a very efficient algorithm that identifies the 3D coordinates of a set of twenty five joints by integrating the depth and shape information [45, 53]. Because the camera captures 30 frames per second, it is possible to obtain time-series data of joint coordinate positions for the duration of an exercise. These coordinates are calibrated to the real-world units and, hence, can be readily analyzed.

Recent results have demonstrated that 3D joint data obtained from the non-invasive Kinect camera is comparable to what can be achieved using invasive and more expensive marker-based systems [10, 20, 22]. In [23], the authors exploit the joint information to ensure anonymity of the participants.

2.3 Multiple Kinect Cameras

Multiple Kinect cameras has been used in several applications. In all these cases, two problems have been addressed. The first is to calibrate the depth sensing across multiple cameras. The second problem is to fuse the joint location data.

In [5] the authors designed a system that can produce realistic, full 3-D reconstructions of foreground moving objects in real-time. All pairs of cameras were calibrated in pairs. RGB image inputs of a calibration bar were used to find point correspondences from two cameras in each pair.

Three Kinect cameras with minimal non-overlapped regions were used to track people in indoor spaces [6]. Since the non-overlapped region was minimal,a common plane is used as a feature for global coordinate system instead of point correspondences. A pole with two boards attached at the ends is utilized as a calibration tool. The tool was positioned, such that each board could be detected by two cameras, thus, creating a pair of corresponding planes.

Calibration of Kinect cameras is done using a standard checker board. in [37]. Transformation between the pair of cameras is done by using a best fit plane approach. Checker board is positioned such that it was visible to both cameras in the pair. RGB image data is used to determine the center of the checker board and the vector normal to it. Relative translation and rotation of cameras is computed using the above information. This approach requires precise positioning of the checker board in the overlapping region of the cameras. Another report using a checker board and SVD to calibrate Kinect cameras was reported in [11]. The RGB images from the cameras were used as input for the calibration.

In [50] multiple Kinect cameras were used for dismounted soldier training. Classification and fusing corresponding joints is done by using RGB image data. A spherical object, such as a basketball, is used to calibrate Kinect cameras that were

used in a gait monitoring application to predict falls. Image processing techniques were used to track the silhouette of the sphere. The transformation parameters were obtained by using SVD on the 3D centers of the spheres.

Based on review [16] authors discussed benefit of using the Kinect IR camera instead of a RGB camera. IR camera in Kinect is used to capture narrow band images filter out most undesired ambient light and makes the system robust to natural indoor illumination. Since skeleton data acquisition is done by using IR depth camera, calibration using skeleton joint information is immune to noise when compared to using RGB images.

A large part of the literature uses either point cloud data or data from the RGB and IR cameras to calibrate multiple kinects. Use of an external calibration tool or offline analysis before performing the exercise is undesirable in case of indoor exercising. In this thesis, skeleton joint data will be extracted from Kinect cameras during the exercise using java API and calibrate the multiple kinect cameras using this joint data.

2.4 Singular Value Decomposition

The authors in [19] used a registration method based on SVD to fuse PET-CT images to get both functional and anatomical information of the animals. They identified the point sources for both CT and PET images and used a least squares approach to minimize the co-registration error. SVD was used along with geometric analysis in [3] to estimate the rigid-body transformation to align the laser scans of the point

correspondences in metric space. Defects in fabric were found using SVD on sub images of the fabric in [15]. The authors computed the average singular value and used this as a threshold value for the fabric. Defective fabrics resulted in different average singular value. In [29] authors extracted embodied knowledge from the time-series data of motion by using SVD. A matrix was formed using the time-series data and they used the left singular vectors of the matrix as the patterns of the motion. These vectors were used as features to classify the gestures. Walking disability was evaluated using the singular values obtained from the SVD.

The work in [34] is closely related to the work reported here. They also used SVD to find the transformation parameters between multiple Kinect cameras. However, they did not consider the issue of synchronization across multiple Kinect cameras. For this reason, work submitted in this thesis is more likely to be useful for exercises that involve fast motions.

CHAPTER III

APPROACH

The Kinect camera and the skeletal tracking algorithm provide 3D coordinates for the location of twenty five joints. These coordinates are with respect to the location of the camera as the origin. As already noted, when an exercise involves a complex sequence of motions, not all the joints can be tracked by a single camera because of occlusions. Thus, the objective in this investigation was to use multiple Kinect cameras to accurately track the locations of all the joints.

Three challenges were addressed when collecting data from multiple Kinect cameras. First, since the data from each camera had to transformed common reference frame. For this purpose, one of the cameras (Camera 1) was selected as the reference. Rotation and translation operators were designed to transform the data from every camera to the common reference. Second, the cameras had to be synchronized so that every camera collected data at the same time. For this purpose, cameras are networked and a command response model is designed to carry out the experiment. Finally, it is important to cope with the the differences in the sampling rates of the cameras that are inherent in the electronic subsystem of the cameras.

This chapter describes the approach that was used to address the above three challenges.

3.1 Common Reference for 3D Joint Coordinates Data

To obtain a common reference, well-known SVD spatial transformation was used [9]. SVD is a decomposition of an input matrix, M, $M = U \cdot \Sigma \cdot V^T$ where U and V are the orthonormal eigenvectors of $M \cdot M^T$ and $M^T \cdot M$, respectively. Σ is a diagonal matrix containing the square roots of the eigenvalues of U (or V) in descending order.

To spatially transform the data to a common reference, two cameras are considered, 1 and 2, as illustrated in Figure 3.1. Rotation is a 3×3 matrix, represented in terms of $R_x(\theta_x), R_y(\theta_y), R_z(\theta_z)$ where $\theta_x, \theta_x, \theta_x$ are the rotations of Kinect camera 1 with respect to x,y and z axes respectively. The translation T is represented as $T = (\Delta_x, \Delta_y, \Delta_z)$. This means that a joint coordinate $c = (x, y, z)$ that is collected from camera 1 will have the coordinates $c \cdot R + T$ with respect to camera 2.

3.1.1 Calculation of R and T from Kinect data

Let the data at time instant t from the i^{th} Kinect camera for joint j be $\vec{\mathbf{J}}_{i,j} = ({x_j}^i, {y_j}^i, {z_j}^i)$ [1]. To compute the centroids, joints that are tracked with high confidence in both the cameras are considered [2]. Let Q represent the set of joints that are tracked

[1] The major benefit of using the skeletal tracking algorithm is that the joint registration problem is already resolved. That is, the data from camera 1 for joint j corresponds to the same joint from camera 2.

[2] The tracking algorithm also provides a *tracking state*; this is a value between 0 and 1 that indicates the confidence in the estimate of the joint position.

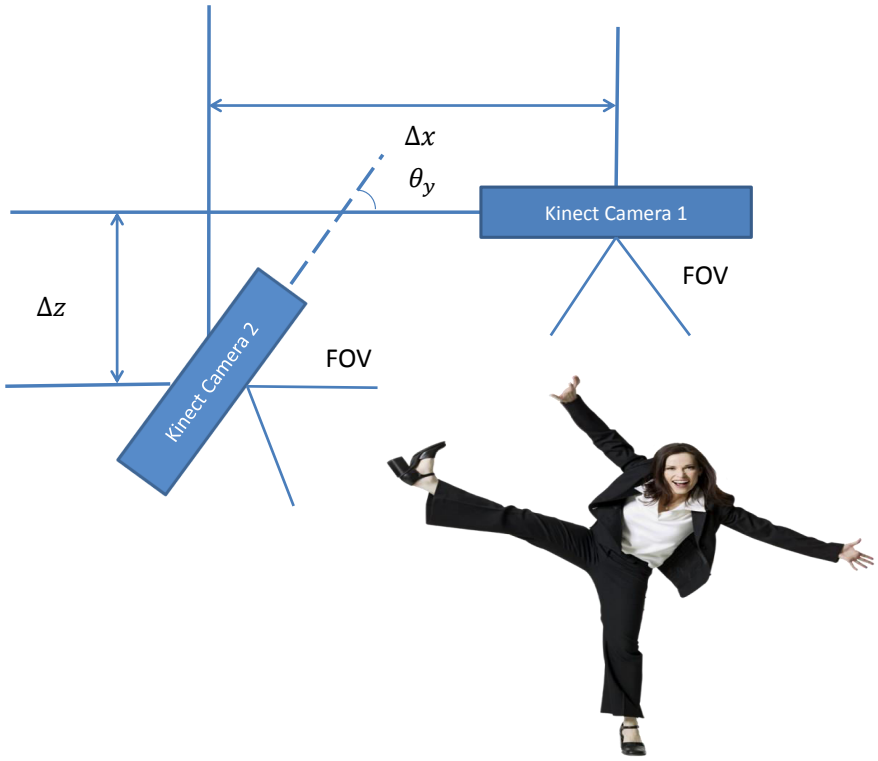

Figure 3.1: The 3D Joint coordinates from each camera are provided by considering the location of the camera as the origin. When data for the same exercise are collected using multiple cameras, it is necessary to translate and/or rotate the coordinates from one camera to the frame of reference of the other camera.

by both camera 1 and camera 2, and let $n = |Q|$. Then,

$$\vec{\mu}_1 = \frac{1}{n}\Sigma_{i \in Q}\vec{\mathbf{J}}_{1,i} \tag{3.1}$$

$$\vec{\mu}_2 = \frac{1}{n}\Sigma_{j \in Q}\vec{\mathbf{J}}_{2,j} \tag{3.2}$$

The input matrix, M, that is decomposed is constructed as

$$M = \frac{1}{N} \cdot \Sigma_{i=1}^{N} \cdot [(\vec{\mathbf{J}}_{1,j} - \vec{\mu}_1) \cdot (\vec{\mathbf{J}}_{2,j} - \vec{\mu}_2)], \tag{3.3}$$

where $N = 25$ for the Kinect 2.0 cameras because the skeletal tracking algorithm provides information for 25 joints [38].

SVD of M gives, $M = U \cdot \Sigma \cdot V^T$.The SVD is useful because calibration is done using the cameras that are collecting joint data and obtain the rotation and translation matrices as [9] as:

$$R = V \cdot U^T, \tag{3.4}$$

and

$$T = -R \cdot \vec{\mu}_1 + \vec{\mu}_2, \tag{3.5}$$

where $\vec{\mu}_1$ and $\vec{\mu}_2$ are the centroids of joint coordinate data that are tracked by both cameras 1 and 2.

To guide the SVD, an objective function that represents the current error is considered. For this purpose, the mean square error will be defined as

$$error = \frac{1}{N} \cdot \Sigma_{i=1}^{N} ||\vec{\mathbf{J}}_{1,i} \cdot R + T - \vec{\mathbf{J}}_{2,i}||^2. \tag{3.6}$$

The SVD method provided a decomposition that minimized the above error.

3.2 Common Reference Time

To fuse joint coordinates data from two cameras, it is important to ensure that the time at which the data are collected in the respective cameras is synchronized. Because of the large volume of data that are collected from a Kinect camera, only one camera could be connected to a computer at a time. Thus, it was necessary to network multiple computers and synchronize the collection operation across these computers.

Two separate problems were addressed to synchronize the collection. The first problem involved synchronizing the data collection by estimating the network delays over each link. Our approach to address this problem is described in this section. Also, it was noted that although each camera was supposed to collect data at 30 frames per second, or 33.33 ms per frame, this rate actually varied between 26 ms and 45 ms from one camera to another. Solutions to both these problems are now described.

To ensure that the cameras collected data at the same time, the cameras were connected in a client-server configuration. Each computer that was connected to a Kinect camera was a client and the computer that was collecting the data from all the cameras was the server. A standard Ethernet connection using the TCP protocol was used. The main idea for the synchronization was to estimate the network delay between the server and each client. An assumption was made that that the network delay would be no-worse than the estimated value for the duration of the experiment.

Further assumptions were made that all the computers on the network had synchronized time. The server sent the estimated delay to each client and initiated data collection at a future time.

When the Kinect camera was powered up, it requested a connection to the server. The server sent a probe message for each client and the client responded to this with an ACK. By recording the time stamps at which these messages were sent at either end, average network delay was computed, ND, for the link between the client and the server. After the delay with respect to each client was computed, the server computed the maximum delay. The server initiated a start collection message at a future time of twice the maximum delay over all the links.

Upon receipt of the start collection message, each client staggered the collection to the future time by accounting for the delay on the link connecting it to the server.

3.3 Compensating for Jitter

Although a Kinect camera is expected to sample frames at an interval of 33.33 ms (30 frames per second), the observed sampling interval varied from 26 ms to 45 ms. The difference in this interval was different for different cameras as illustrated in Figure 3.2. This is the well-known jitter problem in real-time systems and arises both because of hardware (electronic) and software issues. Although this jitter would be acceptable for exercises involving little or slow motion, it results in incorrect data

in fast moving exercises because it is difficult to fuse data that are not on a consistent time base.

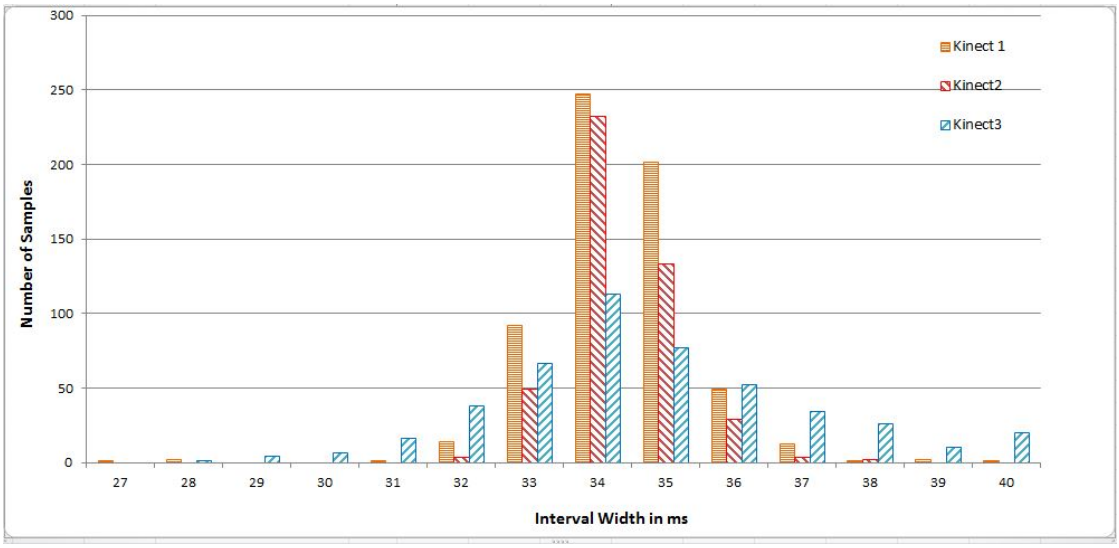

Figure 3.2: Although the sampling interval in a Kinect camera was expected to be around 33 ms, this interval was observed to vary from 27 ms to 42 ms as illustrated in the figure. Notice that the variability differs from one camera to another and also varies depending on the environment conditions.

To compensate for this jitter, the data needs to be interpolated from the cameras to a common 10 ms time base. The key idea here is illustrated in Figure 3.3. Here, the $\mathbf{x_i}$ represent the joint coordinates collected using the Kinect camera. Recall, each sample is the location of a joint in 3D space. These samples are approximately 33 ms apart in time. The objective is to interpolate these data to a common time base of 10 ms.

Figure 3.3 depicts the linear interpolation approach explored. In this approach, it is assumed that the trajectory of a joint between two adjacent joint locations $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$ is a line that connects these two locations (in 3D space). Let $\mathbf{a_i}$

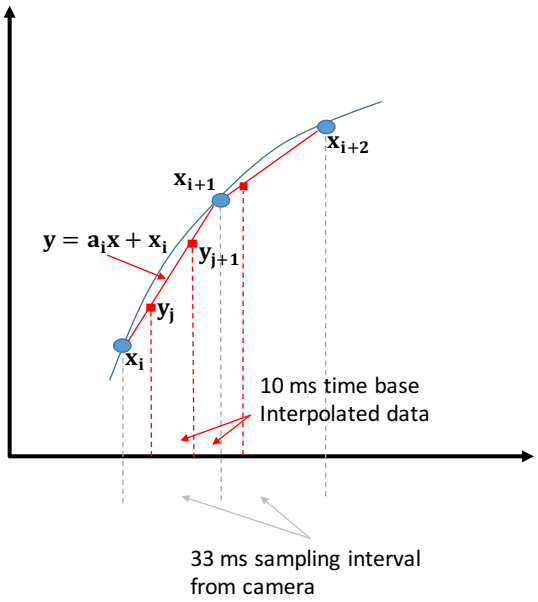

Figure 3.3: To compensate for the jitter in the sampling interval, all the data were interpolated to a 10 ms time base. This figure illustrates linear interpolation. The sampled joints are illustrated as $\mathbf{x_i}$. The trajectory of the joint between two successive sample points was assumed to be linear.

represent the direction vectors (equivalent of slope in 2D space) of the line between $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$. Then, in vector form, the interpolated values

$$\mathbf{y = a_i x + x_i}.$$

Intuitively, using the line between $\mathbf{x_i}$ and $\mathbf{x_{i+1}}$, the value $\mathbf{\hat{x}_{i+2}}$ is estimated. An actual value at $\mathbf{x_{i+2}}$ is already known, so the error is calculated and added this mean squared error over all joint locations to compute the total error in the linear interpolation approach. Since the samples $\mathbf{x_i}$ are collected sequentially, s interpolated values $\mathbf{y_j}$ are

computed sequentially.

The trajectories of all the joints were not linear. For example, in the jumping jacks exercise, the elbow and wrist move along an arc. For this reason, cubic spline interpolation is explored using the Java Scientific Library [21] and three successive joint coordinates collected via the Kinect camera. Cubic Spline Interpolation uses a third degree polynomial to interpolate the values.

$$\mathbf{y = a_i x^3 + b_i x^2 + c_i x + d_i}.$$

Again, since the actual data were already available, computations were done to obtain total error in the interpolated data.

The results obtained using this approach are presented in the next chapter.

3.4 Recognizing Exercises and Detecting Errors

Following the approach reported in our prior work [30, 41], a two step-process to recognize exercises and detect errors that can occur when performing these exercises. Using the 3D joint coordinate data, features were extracted that represent key aspects of the exercise and represent these as a feature vector. A Support Vector Machine (SVM) was trained to classify such feature vectors to one of the known exercises. This approach was validated in experiments and the approach correctly recognized exercises 83% of the time.

For each exercise in the HICT suite [31], a list of potential errors which occur were identified through discussions with experts in exercise science. These errors were codified as geometric properties. Using the 3D joint data, it is possible to accurately detect these errors within four repetitions of an exercise being performed [30, 41].

CHAPTER IV

RESULTS

This chapter describes the experimental setup and presents results from the experiments that validate the approach.

4.1 Experiment Setup

Three Kinect cameras were used as illustrated in Figure 4.1. The data collection from Kinect Camera is done by using *Java4Kinect* package [8] in Java. The center camera designated as the frame of reference to which all the data were transformed. Each camera was connected to a separate laptop computer for collecting the data via its USB port. The computers were connected to a central server as described earlier.

The joint coordinates data were collected from each Kinect camera using a Java API. The local computer assembled these data and sent these data to the server using the standard Transmission Control Protocol (TCP) supported in Ethernet. These data were interpolated on the server.

A participant was asked stand in the surrender pose for 20 seconds before any exercises were performed. The data from this pose were used to assemble an input matrix for the SVD. The SVD transformation was performed using the standard Java linear algebra package called JAMA [27]. The data from the cameras were

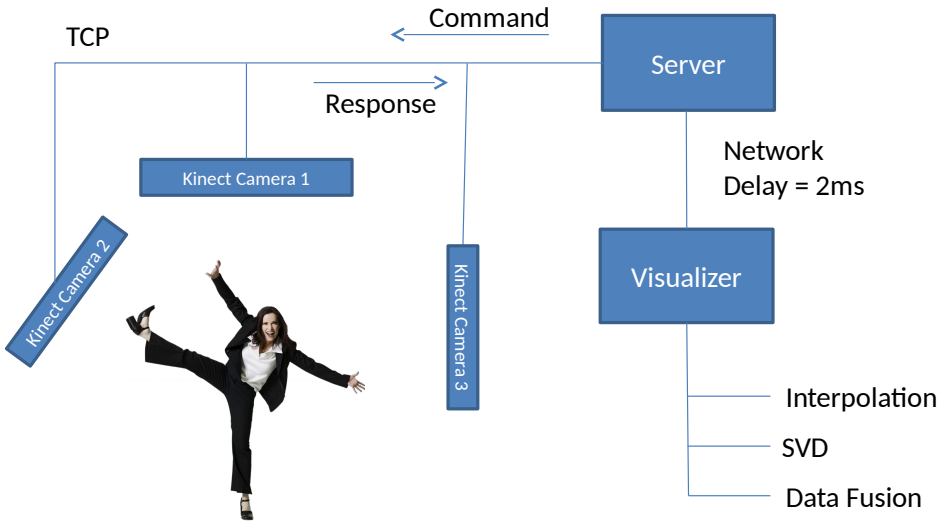

Figure 4.1: Experimental Setup with three cameras.

interpolated to a 10 ms time base before applying SVD. The resulting transformation matrices, R and T, were used to transform the data that were collected when the participant performed exercises. The exercise data were collected using the three cameras as mentioned and interpolated to a 10 ms time base. The data in each camera that was not already a reference camera, were transformed using the R and T matrices. After receiving data from multiple cameras, these data were fused using the tracking index. If the tracking index value was below a threshold value, the data was discarded. When the tracking index for a joint was high from multiple cameras,

the transformed data from these cameras were averaged to produce a single joint coordinate for each joint.

4.2 Selecting Interpolation Method

Data from the cameras were interpolated using linear interpolation and cubic spline interpolation using the Java Scientific library [21]. Since the actual measured data is available, error is measured using both these methods. The error is shown in Figure 4.2 and Table 4.1 for a specific exercise called Jumping Jacks. Notice that the cubic spline interpolation consistently resulted in smaller error.

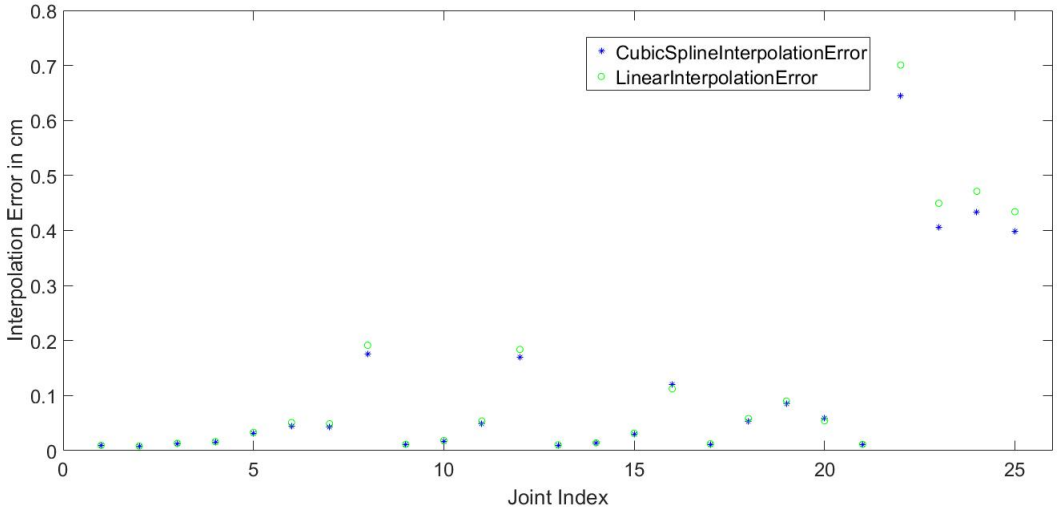

Figure 4.2: Cubic spline interpolation results in consistently lower error than linear interpolation.

4.3 Without Interpolation

To understand whether the interpolation was necessary, transformation error was computed for a stationary pose. Figure 4.3 illustrates the in transformation, i.e., mean squared distance between estimated and measured joint coordinates. Notice that the error is consistently lower when interpolation is used. This justified the need for interpolating the data to a common time base.

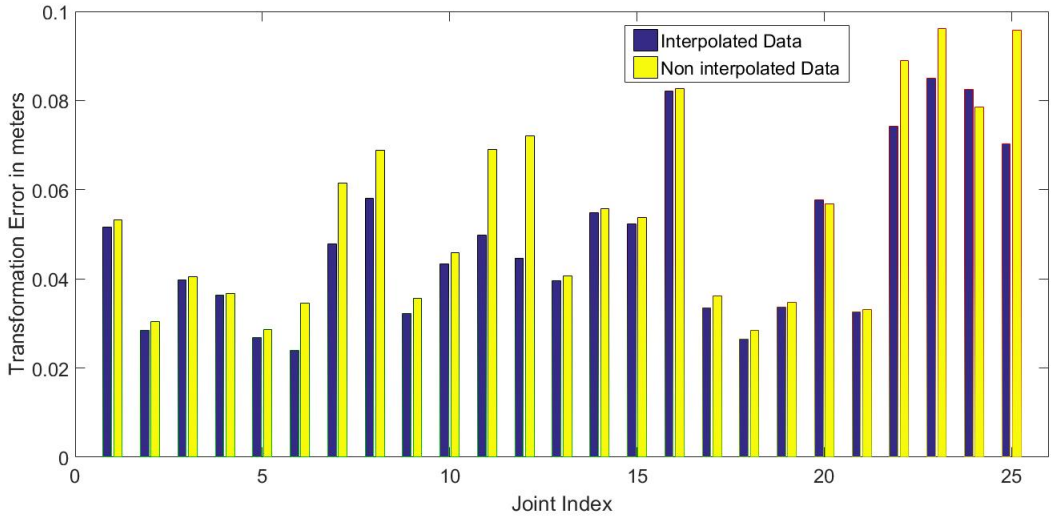

Figure 4.3: Comparison of transformation error with interpolated data vs non interpolated data

4.4 Transforming frame of reference

To validate the SVD based transformation approach, data was collected when Jumping Jacks exercise was performed. The data were collected from two cameras, one in the Frontal plane and one in the Sagittal plane. The average tracking index was

computed for each joint from each camera and is illustrated in Figure 4.4. The data from Kinect 2 (Sagittal plane) were insufficient to analyze the synchrony between the two ankles when this exercise is performed. Note that in this experiment, Kinect 1 data was collected as a reference.

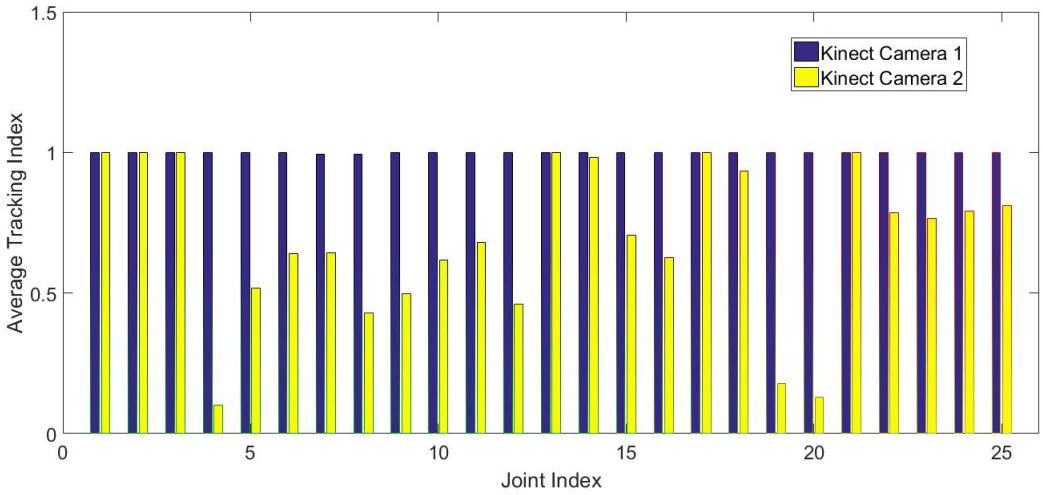

Figure 4.4: The average tracking index for the joints from Kinect 2 (Sagittal plane) is lower than that for Kinect 1 (Frontal plane).

The validity of the SVD based approach is illustrated in the next three figures. Figure 4.5 presents the time series data of the left ankle and the right ankle when the Jumping Jacks exercise is performed. These motions should be synchronized and in contrast to an ideal value of -1, the correlation between these data was computed to be -0.7451. This means that a novice likely performed the exercise, and the motion of the ankles was not very well synchronized.

Figure 4.6 presents the data for the same motion that was collected using Kinect 2 in the Sagittal plane. Notice that the left ankle is occluded and the computed

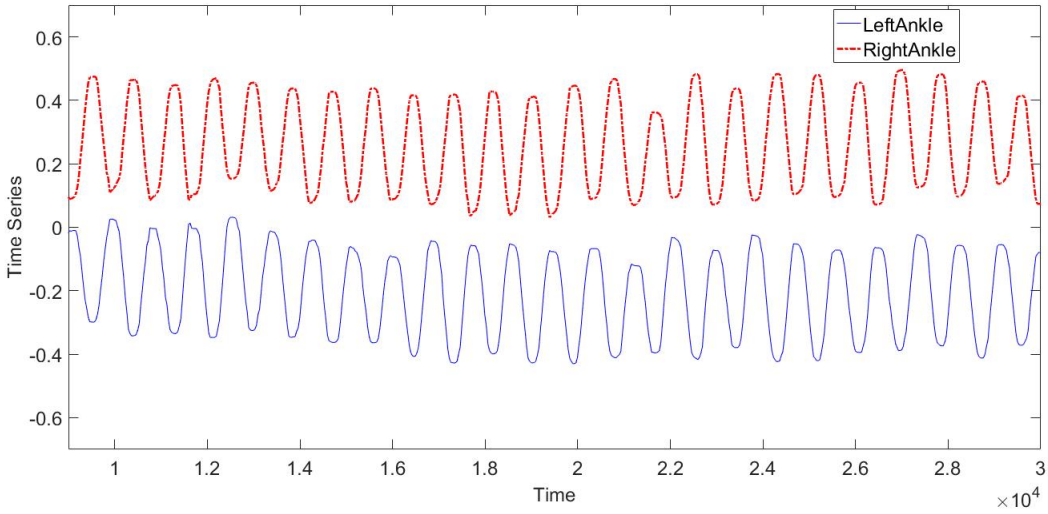

Figure 4.5: Time Series data of the Left Ankle and Right Ankle from Kinect 1 (Frontal plane).

correlation was -0.5852. If this value is used to analyze the exercise, it would result in a false conclusion.

Figure 4.7 shows the data for the left ankle collected in Kinect 2 that was transformed to the frame of reference of the Kinect 1 camera. Notice that the transformation is visually close to the original data shown in Figure 4.5. The computed correlation using the transformed data was -0.7218 and this is within 10% of the original value of -0.7451.

4.5 Kinect Data Fusion

After transformation of joint data from all the Kinect cameras to desired frame of reference, fusion is done by using averaging the corresponding joint information with

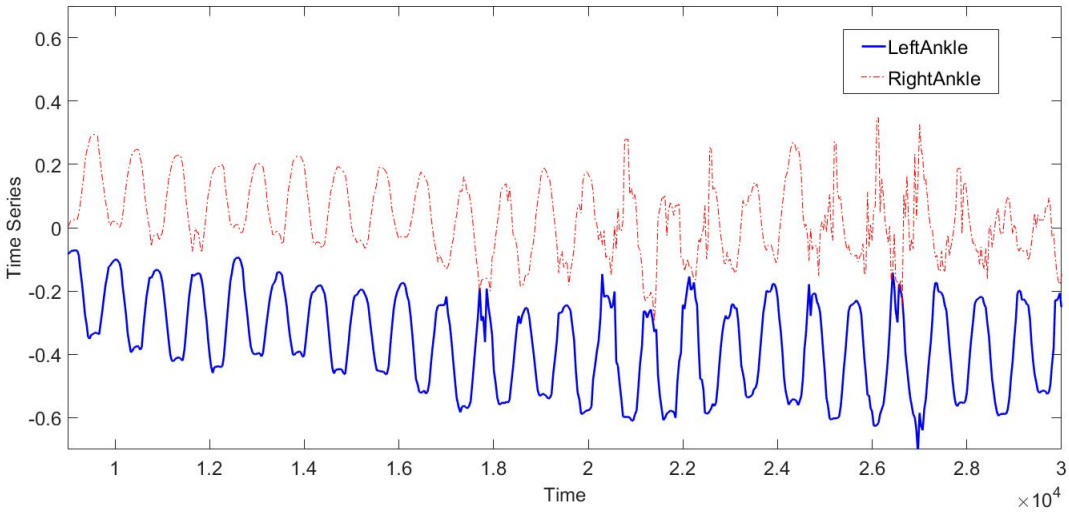

Figure 4.6: Time Series data of the Left Ankle and Right Ankle from Kinect 2 (Sagittal plane).

highest tracking index. In Figure 4.8 we can observe the transformation error of the fused set is low when compared to individual transformed joint sets.

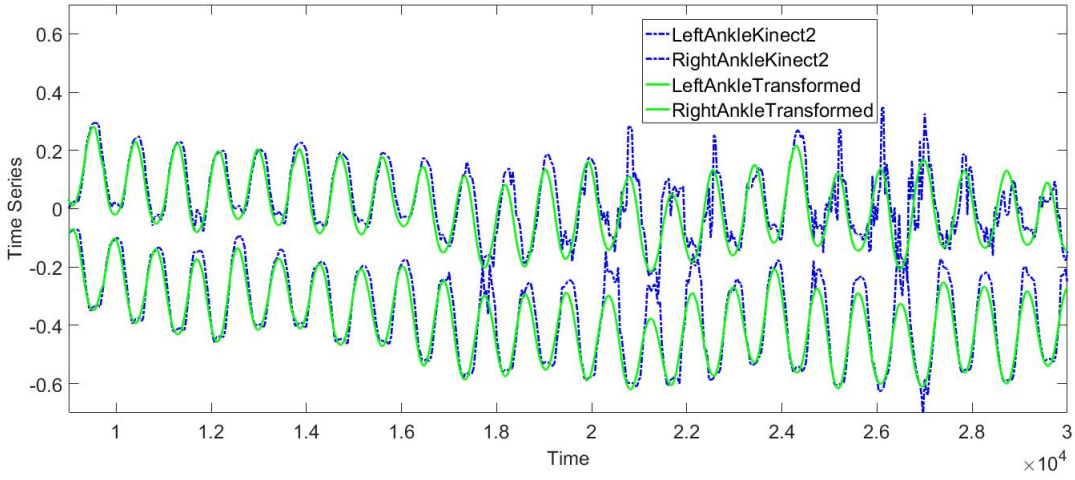

Figure 4.7: Time Series Plot of the transformed data for the Left Ankle collected using Kinect 2 and the Right Ankle collected from Kinect 1.

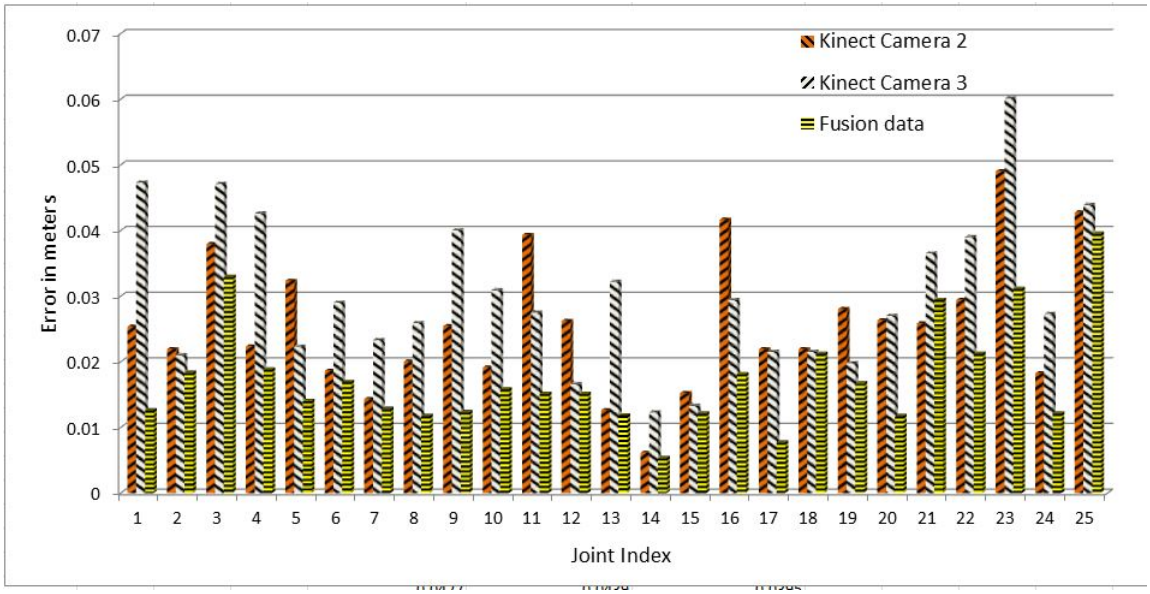

Figure 4.8: Comparison of transformation error of fused joint information with individual transformed joint coordinate information.

Table 4.1: Cubic spline interpolation results in consistently lower error than linear interpolation.

Id	Joint	LinearError	CubicError
1	SPINE BASE	0.902	0.691
2	SPINE MID	0.906	0.721
3	NECK	1.009	0.872
4	HEAD	0.905	0.713
5	SHOULDER LEFT	1.039	0.892
6	ELBOW LEFT	1.233	1.140
7	WRIST LEFT	2.369	2.366
8	HAND LEFT	2.928	2.812
9	SHOULDER RIGHT	0.974	0.768
10	ELBOW RIGHT	1.307	1.225
11	WRIST RIGHT	1.773	1.696
12	HAND RIGHT	1.687	1.284
13	HIP LEFT	0.958	0.759
14	KNEE LEFT	1.051	0.928
15	ANKLE LEFT	0.953	0.874
16	FOOT LEFT	2.544	2.462
17	HIP RIGHT	0.872	0.674
18	KNEE RIGHT	0.949	0.867
19	ANKLE RIGHT	1.108	1.119
20	FOOT RIGHT	2.836	2.574
21	SPINE SHOULDER	0.980	0.831
22	HAND TIP LEFT	3.246	3.122
23	THUMB LEFT	3.850	3.814
24	HAND TIP RIGHT	2.390	2.113
25	THUMB RIGHT	3.344	3.337

CHAPTER V

DISCUSSION

The approach presented in this thesis to fuse data from multiple Kinect cameras is effective. The results in the preceding chapter demonstrated that by using a common time reference, interpolating the acquired data to a common time base and transforming the data to common reference coordinate system effectively allows us to use the average value as the fused coordinate. Several other options can be explored in the future. For example, we can select the value that has the highest tracking index. Another option is to use a Kalman Filter to fuse the data by assuming that the trajectories of the joints are linear [32].

Fusing joint coordinate data from multiple cameras enables us to encode expert domain knowledge as detection rules. The current state-of-the-art for exercise diagnosis is a system that plays back recorded video frame by frame. Our system on the other hand can alert the participants when errors are identified. When the experts have fully verified these diagnoses, we can deploy the software system for routine use at different locations so as to improve exercise adherence.

CHAPTER VI

CONCLUSION

The thesis presented an approach for fusing 3D joint coordinates data from multiple Kinect cameras in the context of recognizing and detecting errors that can occur during the performance of exercises. This capability allowed us to recognize all the exercises in the HICT suite. Following our prior work, the error detection relies on the formulation of computationally efficient geometric properties that could be extracted from the time-series data of the joint locations.By fusing the data from multiple Kinect cameras, an alternative approach is offered to mitigate the effects of occlusions. Nevertheless, this system can be extended in the future to provide cost-effective solutions to improve exercise adherence and wellness management.

BIBLIOGRAPHY

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16–58, 2011.

[2] J. K. Aggarwal and L. Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 48:70–80, 2014.

[3] S. Ahuja and S.L. Waslander. 3d scan registration using curvelet features. In *Canadian Conference on Computer and Robot Vision*, pages 77–83. IEEE, 2014.

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

[5] D.S. Alexiadis, D. Zarpalas, and P. Daras. Real-time, full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras. *Multimedia, IEEE Transactions on*, 15(2):339–358, 2013.

[6] E. Almazan and G. Jones. Tracking people across multiple non-overlapping rgb-d sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 831–837, 2013.

[7] D. Anton, A. Goni, A. Illarramendi, J. J. Torres-Unda, and J. Seco. Kires: A kinect-based telerehabilitation system. In *Proceedings of 15th International Conference on e-Health Networking, Applications Services*, pages 444–448. IEEE, 2013.

[8] A. Barmpoutis. Tensor body: Real-time reconstruction of the human body and avatar synthesis from rgb-d. *IEEE transactions on cybernetics*, 43(5):1347–1356, 2013.

[9] P.J. Besl and N.D. Mckay. A method for registration of 3-d shapes. *IEEE transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[10] B. Bonnechre, B. Jansen, P. Salvia, H. Bouzahouene, L. Omelina, F. Moiseev, V. Sholukha, J. Cornelis, M. Rooze, and S. V. S. Jan. Validity and reliability of the kinect within functional assessment activities: Comparison with standard stereophotogrammetry. *Gait and Posture*, 39(1):593–598, 2014.

[11] M. Caon, J. Tscherrig, Y. Yue, O.A. Khaled, and E. Mugellini. Extending the interaction area for view-invariant 3d gesture recognition. In *Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on*, pages 293–298. IEEE, 2012.

[12] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal. Bio-inspired dynamic 3d discriminative skeletal features for human action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 471–478. IEEE, 2013.

[13] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, and J. Xiao. Learning a 3d human pose distance metric from geometric pose descriptor. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1676–1689, 2011.

[14] C.H. Chen, J. Favre, G. Kurillo, T.P. Andriachhi, R. Bajcsky, and R. Chellappa. Camera networks for healthcare, teleimmersion, and surveillance. *IEEE Computer*, 47:26–36, 2014.

[15] S. Chen and J. Feng. Research on detection of fabric defects based on singular value decomposition. In *IEEE International Conference on Information and Automation*, pages 857–860. IEEE, 2010.

[16] G. Choe, J. Park, Y.W. Tai, and I. So Kweon. Exploiting shading cues in kinect ir images for geometry refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3922–3929, 2014.

[17] E. Davaasambuu, C. Chiang, J. Y. Chiang, Y. Chen, and S. Bilgee. A microsoft kinect based virtual rehabilitation system. In *The 5th International Conference FITAT*, pages 44–50, 2012.

[18] N.M. DiFilippo and M.K. Jouaneh. Characterization of different microsoft kinect sensor models. *IEEE Sensors Journal*, 15(8):4554–4564, 2015.

[19] B. Feng, S. Yan, M. Chen, D.W. Austin, J. Deng, and R.A. Mintzer. Automated least-squares calibration of the coregistration parameters for a micro pet-ct system. *IEEE Transactions on Nuclear Science*, 58(5):2303–2307, 2011.

[20] A. Fern'ndez-Baena, A. Susin, and X. Lligadas. Biomechanical validation of upper-body and lower-body joint movements of kinect motion capture data for rehabilitation treatments. In *4th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pages 656–661. IEEE, 2012.

[21] M.T. Flanagan. Java scientific library. `http://www.ee.ucl.ac.uk/~mflanaga/java/`, 2015.

[22] S.P. Gaddam, M.K. Chippa, S. Sastry, A. Ange, V. Berki, and B.L. Davis. Estimating forces during exercise activity using non-invasive kinect camera. In *International Conference on Computational Science and Computational Intelligence*, pages 825–28, 2015.

[23] A. Ghose, K. Chakravarty, A. K. Agrawal, and N. Ahmed. Unobtrusive indoor surveillance of patients at home using multiple kinect sensors. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, pages 40–42. ACM, 2013.

[24] A. Gonzalez, P. Fraisse, and M. Hayashibe. Adaptive interface for personalized center of mass self-identification in home rehabilitation. *IEEE Sensors Journal*, 15(5):2814–2823, 2015.

[25] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013.

[26] J. Hernandez-Aceituno, R. Arnay, J. Toledo, and L. Acosta. Using kinect on an autonomous vehicle for outdoors obstacle detection. *IEEE SENSORS JOURNAL*, 16(10), 2016.

[27] J. Hicklin, C. Moler, P. Webb, R.F. Boisvert, B. Miller, R. Pozo, and K. Remington. Jama: Java matrix package. `http://math.nist.gov/javanumerics/jama/`, 2012.

[28] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2466–2472. AAAI Press, 2013.

[29] Y. Jiang, I. Hayashi, and S. Wang. Knowledge acquisition method based on singular value decomposition for human motion analysis. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3038–3050, 2014.

[30] J. Juvancic-Heltzel, H. Pidaparthy, V.E. Pinhiero, and S. Sastry. Detection of exercise errors analyzing time-series video-captured data. *Medicine and Science in Sports and Exercise*, 47(5S):228, 2015.

[31] B. Klika and C. Jordan. High-intensity circuit training using body weight: Maximum results with minimal investment. *ACSM Health and Fitness*, 17(3):8–13, 2013.

[32] W. Kong, A. Hussain, and M. H. Saad. Essential human body points tracking using kalman filter. In *World Congress on Engineering and Computer Science*, 2013.

[33] N. Kumar and R. V. Babu. Human gait recognition using depth camera: A covariance based approach. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 21–26. ACM, 2012.

[34] S. Li, P.N. Pathirana, and T. Caelli. Multi-kinect skeleton fusion for physical rehabilitation monitoring. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5060–5063. IEEE, 2014.

[35] T. Y. Lin, C. H. Hsieh, and J. D. Lee. A kinect-based system for physical rehabilitation: Utilizing tai chi exercises to improve movement disorders in patients with balance ability. In *Modelling Symposium (AMS), 2013 7th Asia*, pages 149–153, 2013.

[36] R. Lun and W. Zhao. A survey of applications and human motion recognition with microsoft kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(05), 2015.

[37] R. Macknojia, A. Chávez-Aragón, P. Payeur, and R. Laganiere. Calibration of a network of kinect sensors for robotic inspection over a large workspace. In *2013 IEEE Workshop on Robot Vision (WORV)*, pages 184–190. IEEE, 2013.

[38] Microsoft. Microsoft xbox one features. `http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx`, 2014.

[39] S. Obdrzalek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly. In *IEEE Annual International Conference of the Engineering in Medicine and Biology Society*, pages 1188–1193. IEEE, 2012.

[40] E. Ohn-Bar and M. M. Trivedi. Joint angles similarities and hog2 for action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 465–470. IEEE, 2013.

[41] H. Pidaparthy, S. Narayan, and S. Sastry. Automated exercise feedback system using non-invasive sensor. *Applications of Computer Vision in Graphics and Image Processing, ICVGIP*, 2014.

[42] M. Robinson and M. B. Parkinson. Estimating anthropometry with microsoft kinect. In *Proceedings of the 2nd International Digital Human Modeling Symposium*, 2013.

[43] S. Ruffieux, D. Lalanne, E. Mugellini, and O.A. Khaled. Gesture reognition corpora and tools: A scripted ground truthing method. *Computer Vision and Image Understanding*, 131:72–87, 2015.

[44] H. Sarbolandi, D. Lefloch, and A. Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer Vision and Image Understanding*, 139:1–20, 2015.

[45] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304. IEEE, 2011.

[46] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML)*, pages 104–112. ACM, 2004.

[47] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[48] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by represention 3d skeletons as points in lie group. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595. IEEE, 2014.

[49] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297. IEEE, 2012.

[50] B. Williamson, J. LaViola, T. Roberts, and P. Garrity. Multi-kinect tracking for dismounted soldier training. In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, pages 1727–1735, 2012.

[51] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. El Saddik. Evaluating and improving the depth accuracy of kinect for windows v2. *IEEE Sensors Journal*, 15(8):4275–4285, 2015.

[52] X. Yang and Y.L. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 14–19. IEEE, 2012.

[53] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012.