Multimodal Tagging of Human Motion Using Skeletal Tracking With Kinect™

A Thesis Presented

by

Debaleena Chattopadhyay

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

in

Computer Science

Stony Brook University

May 2011

Stony Brook University
The Graduate School

Debaleena Chattopadhyay

We, the thesis committee for the above candidate for the
Master of Science degree, hereby recommend
acceptance of this thesis.

Tamara L. Berg
Assistant Professor
Computer Science Department

Dimitris Samaras
Associate Professor
Computer Science Department

Alexander Berg
Assistant Professor
Computer Science Department

Margaret Anne Schedel
Assistant Professor
Music Department

This thesis is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Thesis

Multimodal Tagging of Human Motion Using Skeletal Tracking With Kinect™

by

Debaleena Chattopadhyay

Master of Science

in

Computer Science

Stony Brook University

2011

Recognizing moves and movements of human body(s) is a challenging problem due to their self-occluding nature and the associated degrees of freedom for each of the numerous body-joints. This work presents a method to tag human actions and interactions by first discovering the human skeleton using depth images acquired by infrared range sensors and then exploiting the resultant skeletal tracking. Instead of estimating the pose of each body part contributing to a set of moves in a decoupled way, we represent a single-person move or a two-person interaction in terms of its skeletal joint positions. So now a single-person move is defined by the spatial and temporal arrangement of his skeletal framework over the episode of the associated move. And for a two-person interactive sequence, an event is defined in terms of both the participating agents' skeletal framework over time. In this work we have experimented with two different modes of tagging human moves and movements. In collaboration with the Music department we tried an innovative way to tag a single person's moves with music. As a participating agent performs a set of movements, musical notes

are generated depending upon the velocity, acceleration and change in position of his body parts. We also try to recognize human interactions into a set of well-defined classes. We present the K-10 Interaction Dataset with ten different classes of two-person interactions performed among six different agents and captured using the Kinect™ for Xbox 360. We construct interaction representations in terms of local space-time features and integrate such representations with SVM classification schemes for recognition. We further aligned the clips in our dataset using the Canonical Time Warping algorithm that led to an improvement in the interaction classification results.

To my parents

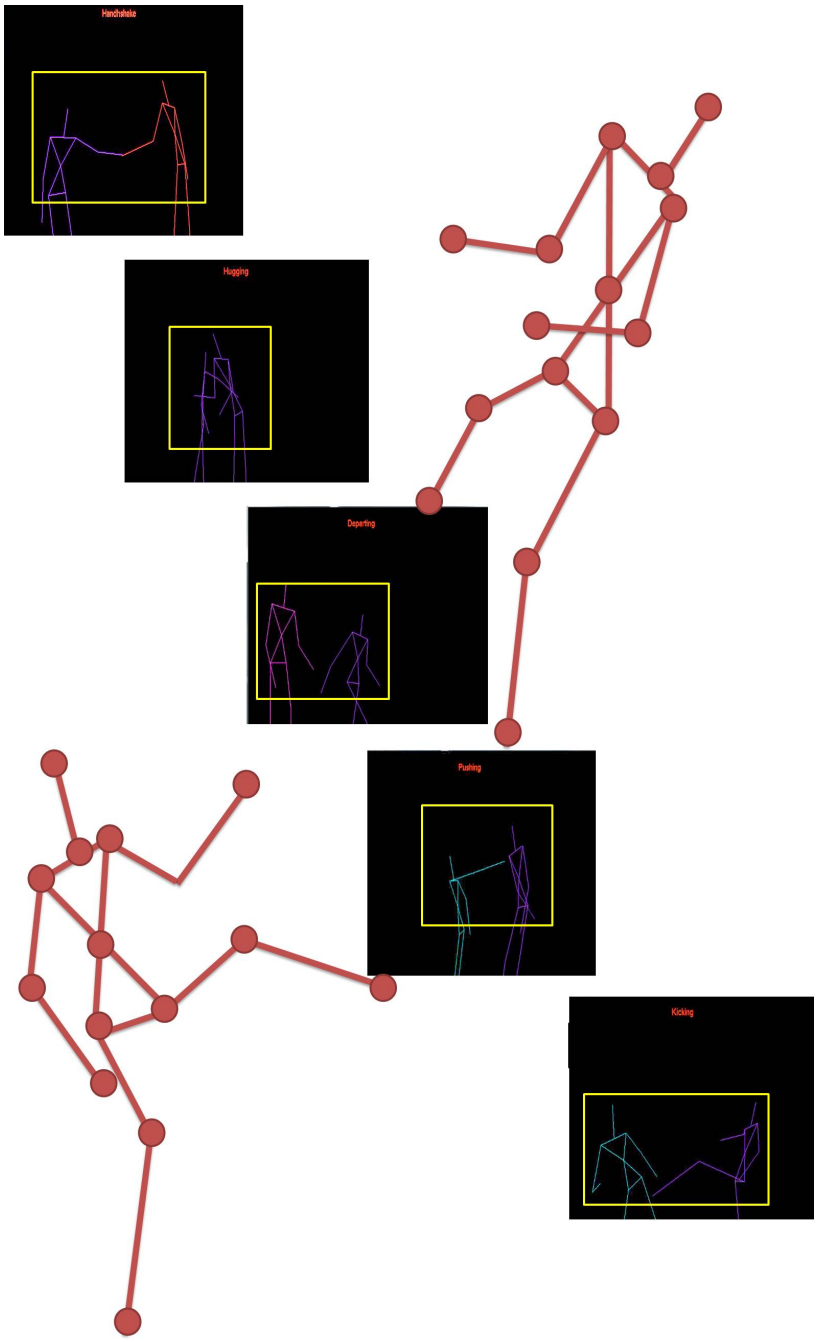
Handshake
Hugging
Departing
Pushing
Kicking

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor, Prof. Tamara Berg, for her guidance during my research and study at the Stony Brook University.

I would like to thank Prof. Margaret Anne Schedel and Timothy Vallier from the Music Department at the Stony Brook University for their kind collaboration in this work.

I would also like to thank my other committee members Prof. Dimitris Samaras and Prof. Alexander Berg.

I am grateful to my fellow friends from the Computer Vision and the Digital Media Lab: Vicente Ordonez, Kiwon Yun, Xufeng Han, Yifan Peng and Yun Zeng who had actively helped me in recording the K-10 Interaction Dataset and made this work possible.

Lastly, my deepest gratitude goes to my parents who had always been and will be a constant inspiration in my academic endeavors.

Chapter 1

Introduction

A full grown adult body has 206 bones and over 230 moveable and semi-moveable joints. The maximum number of degrees of freedom that any joint can have is three. However, the effect of adjacent joints may be summated to express the total amount of freedom between one part of the body and an area more distant to it. The more distant a segment, the greater the degrees of freedom it will possess relative to the torso. Jones et al. [2] cites the example of the degrees of freedom between the distant fingers of the hand and the torso amounting to 17. Now, with such a wide choice of poses and possibilities the human body is capable of numerous moves and movements. And, as it happens, human beings use their bodily movements more than often as a mode to interact. But interaction needs the participation of more than one agent.

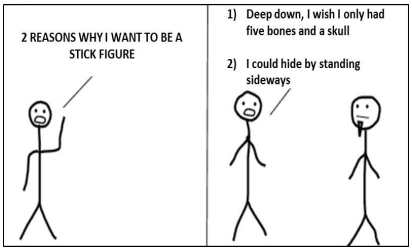

Figure 1.1. 2 Reasons Why I Want To Be A Stick Figure. *[Comic Credit: ThadGuy.com]*

Hence since not long before, interactions utilizing human motion were restricted only to human-human interactions. However with the recent developments in technology, the field of Human Computer Interaction has been exploiting human motion as one of the multimodal interaction possibilities.

Human Computer Interaction applications exploiting gesture recognition, full body tracking and motion detection has become a commoner in today's everyday world. Among the recent advances is the launch of the videogame console Kinect ™ for Xbox 360 in the late Fall of 2010. The following one-liner from Microsoft's marketing campaign for Kinect™ defines what human motion was made capable of:

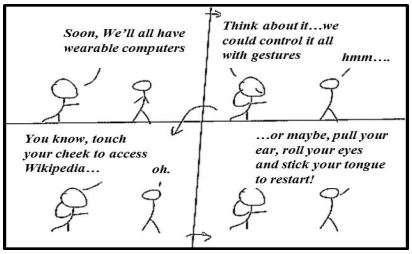

Figure 1.2. HCI Future. *[Comic Credit: Wearable Interaction]*

"You are the controller."

1.1 Our Approach

This work talks about two different types of interactions; human-computer interaction and human-human interaction. And what bridges these two domains in this study is the mode of interaction: human motion. In this thesis, we have experimented with human motion initiated computer interaction as well as used human motion as cues to classify human-human interactions. We also collected our own dataset (K-10 Interaction Dataset) of two-person interactions using the Kinect. We recorded both the depth map and the image map for ten deifferent set of interactions with 6 different subjects. The details of this dataset is presented

in Section 3.2 and available for download at our project website [3].

We have used the technology for skeletal tracking available with the Kinect™ videogame console and developed some algorithms and applications to do multimodal tagging of human moves and movements. The two different modes of tagging used here are *tagging human moves with music* and *tagging human-human interactions into classes*. The former is a Human Computer Interaction project and later is a standard classification problem in Computer Vision. To use the Kinect videogame console for Xbox 360, we had to first interface it with a computer. For that we have used the OpenNI ™ framework [4] and NITE Middleware from PrimeSense™ [5].

1.2 Thesis Overview

In Chapter 1 we introduce the thesis work as well as give a short background study. In Chapter 2, we talk about the application—Tagging moves with music. This is where we have used human motion as a mode of Human Computer Interaction and given expression to that motion in the form of musical notes. In Chapter 3 we set out for solving a standard Computer Vision classification problem. Firstly we talk about the collection of our working dataset. We delve into how the videogame console technology was interfaced with the computer and talk about pros and cons of using an out-of-the-shelf skeletal tracking API. We give the specifications for the K-10 Interaction dataset and also describe why these 10 classes of interactions were chosen for our set of experiments. Finally we discuss how we learn to automatically recognize the interactions captured in the K-10 Interaction Dataset. We give the details of the algorithm, comment on the features used and insights on how this can be extended to any

other activity recognition problem. In Chapter 4 we conclude the thesis with an overall discussion of the work and possible future extensions.

1.3 Background

This background study consists of three subsections. Firstly, we discuss some relevant works that talk about the connection of human motion and music, how one can be mapped into another and possible ways of interactivity. Then we briefly talk about the action and interaction recognition literature, state the two main approaches towards analysis of motion and mention a few state-of-the-art methods for such tasks. The final section gives a short background on the Kinect™ technology and presents a few published works that exploit the available depth sensors of the Kinect™.

1.3.1 Music & Motion

"Is there a true perceptual experience of movement when listening to music, or is it merely a metaphorical one owing to associations with physical or human motion?"

Honing [6] gives an informal yet informative description on how the apparent relation between motion and music has been investigated in a considerable number of works. This article reviews a family of computational models called kinematic models that create explicit relation between motion and music which can be tested and validated on real performance data. The main purpose of citing this article in this section is not to delve into how these computational models express, test or validate this relation, but to justify that tagging human motion with musical notes is not an arbitrary experiment.

The key component behind the symbiotic relationship between dance and music is a series of body movements or human motion. In the computer music literature and the sensor system literature, different systems are proposed from time to time [7] to record different context of motion to better understand this relation.

There are existing sensor systems that capture various forms of gestures using spatial mapping for building interactive surface like smart walls as proposed by Paradiso et al. [8] or dance floors for tracking dance steps as described by Griffith et al.[9]. Paradiso et al. [10] designed an arrangement of tilted accelerometers and pressure sensors at various positions to capture high-level podiatric gesture and proposes an interface for interactive dance. The goal of their work had been to capture a collection of action-to-sound rules for improvisational dancers. Lee et al. [11] proposed a system to extract rhythmic patterns from movement of a single limb using accelerometers in real-time. Wechsler et al. [12] introduces a camera-based motion sensing system that is essentially an interactive video environment which permits performers to use their movements to control or generate sounds.

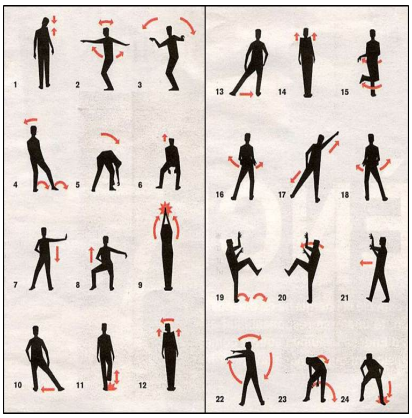

Figure 1.3. Thriller Dance Steps: Music Video Originally from *Michael Jackson's Thriller* In the video's most iconic scene, Jackson leads other actors costumed as zombies to dance in a choreographed routine heavily influenced with music beats and sound effects. *[Image Credit: brian.hoover.net.au]*

In our work, we propose an interactive system that uses the depth sensors of Kinect^{TM} for a whole body skeletal tracking. It is able to automatically gen-

erate musical notes based on the changes in velocity, acceleration and position of a set of skeleton joints in a performing agents body.

1.3.2 Interaction Recognition

Figure 1.4. Human Interactions: These are two clips from the silent movie *The Kid (1921)* by Charlie Chaplin. Before the modern motion pictures, the silent films mostly expressed their storyline through human interactions and occasional on-screen flash of written dialogues. This urges us to think how expressive human actions and interactions are. While the image in the left shows some complex facial emotion, which, given the state-of-the-art Computer Vision Methods still may not be easy to be automatically annotated as an event; we are getting closer towards automatically understanding the event happening in the right image. *[Image Credit: doctormacro.com]*

Action and Interaction Recognition can more generally be categorized under the domain of Human Motion Analysis. Human Motion Analysis concerns detection, tracking and recognition of people, and also, the understanding of human behaviors. The tremendous interest in this field is mainly fuelled by a wide range of potential applications such as smart surveillance, advanced user

interface, motion based diagnosis etc. Aggarwal et al. [13] gives an extensive survey on the prevalent methods in Human Motion Analysis. It can be summarized as in the Fig. (1.4). The three main stages of Human Motion Analysis,

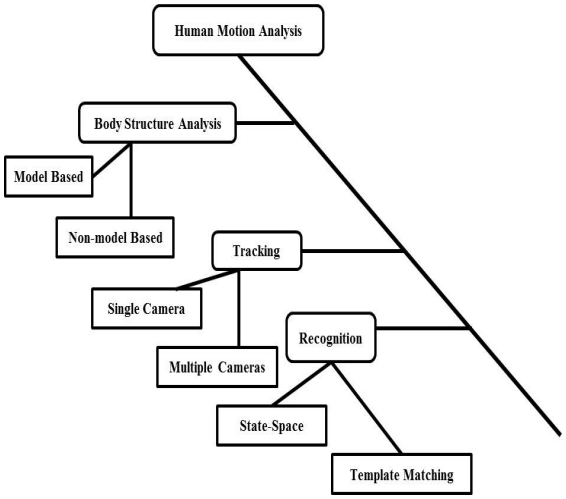

Figure 1.5. Relationship among three major areas of Human Motion Analysis

as shown in the figure, are body structure analysis, tracking and recognition. Among the appearance based and model based approaches for body structure analysis, the Kinect^{TM} technology uses a novel intermediate body parts representation designed to spatially localize joints of interest at low computational cost and high accuracy as proposed by Shotton et al. in [14]. Regarding action recognition, some of the different approaches are image based methods as described by Junejo et al. in [15], [16], part-based methods as proposed by Allin et al. in [17] , and state-space methods using HMM as described by Ramanan et al. [18]. Park [19] proposed an appearance based and state-space modeled method for event recognition of human actions and interactions. He worked towards estimating body-part features (ellipses and convex hulls extracted from already developed segmentation) into body poses using a Bayesian Network. Then the pose estimation results are concatenated to form a sequence which is

classified using a dynamic Bayesian network. Finally, a verbal semantic description of the interaction is generated. Park et al. [20] proposed a method to figure out semantic interpretation of several human interaction sequences. They adopt the verb argument structure in linguistics to represent human action in terms of <agent—motion—target> triplets. They are finally able to annotate various human interactions with user-friendly natural language description and also describe positive, neutral, and negative interactions occurring between two persons.

Our work of interaction recognition is similar in the lines of [19] . We also perform two-person interaction recognition and tag or classify them into one of the 10 well-defined classes. However we use the model based approach of [14] in discovering the skeletal framework of the participating agents and a template-matching method to classify interactions. For classification our template exploits the participating agents joints spatial and temporal information in an inter-dependent way. We give details of our algorithm and furnish results in Chapter 3.

1.3.3 The Kinect™

The recent advances on imaging hardware and computer vision algorithms had led to the emerging technology of markerless motion capture using a camera system. The commercial solution for markerless motion capture currently available in the market is the Microsofts Kinect videogame console. The technology associated with the Kinect™ console discovers the 3D skeleton for a human body and gives us a robust tracking output [14]. The Kinect essentially uses a range camera technology developed by PrimeSense™ that interprets 3D

Figure 1.6. The Kinect™ Game Console. *[Image Credit: gamentrain.com]*

scene information from a continuously-projected infrared structured light. The depth sensors in Kinect consist of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. After recording the 3D scene information, the Kinect first evaluates how well each pixel fits certain features for example, is the pixel at the top of the body, or at the bottom? This gives each pixel a certain score. The score for each feature is then combined with a randomized decision forest search through. A randomized decision forest search is essentially a collection of decisions that asks whether a pixel with a particular set of features is likely to fit a particular body part. The Kinect technology has already been trained on a collection of motion capture data (around 500,000 frames). Once the body parts have been identified, the system then calculates the likely location of the joints within each one to build a 3D skeleton. The Microsoft Xbox runs this algorithm 200 times per second, which is around ten times faster than any previous body-recognition techniques ensuring players can easily be tracked fast enough for

their motions to be incorporated in to games.

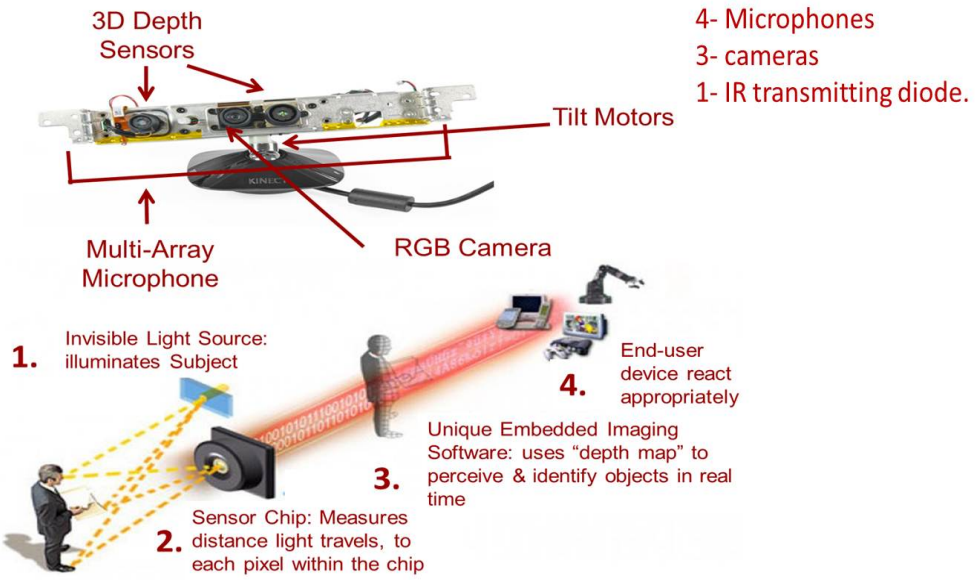

Figure 1.7. The Kinect^{TM} Sensors in Play *[Image Credit: wired.com]*

The depth sensors of Kinect are being recently utilized into different Computer vision tasks. Bleiweiss et al. [21] proposes a real-time framework for blending full-body tracking of a player with pre-defined set of gestures to enhance interactive gaming experience. Xia et al. [22] uses the depth information from Kinect and proposes an algorithm with 2-D head contour and a 3-D head surface model for human detection.

We have used the depth sensors of the Kinect^{TM} to do full-body tracking of human agents and tagged the moves and movements into different modes like musical notes and classification categories.

Chapter 2

Tagging Moves with Music

2.1 Introduction

The idea of tagging moves with musical notes came to us while on a discussion on how music can give us a sense of depth. While designing an application, depth visualizer [screenshots in Figure 2.1] that will enable a visualization of how an agents/objects 3D position is changing and also sync an Open Sound Control-enabled application to generate musical notes based on those 3d co-ordinate positions, we realized it will be worthwhile to think of an extension.

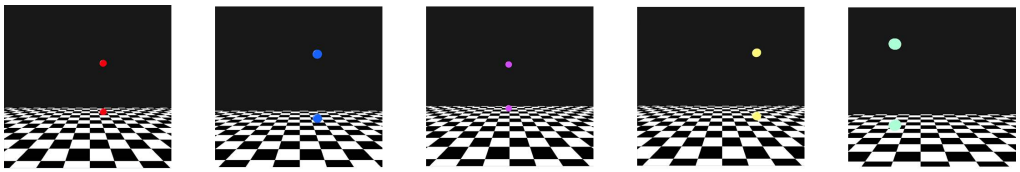

Figure 2.1. A Simple Depth Visualizer Synced with OSC-enabled Multimedia Applications

Now, with the Kinect™ console interfaced with the computer using proper interfaces, what was required was to make a bridge between the Kinect™

Console and Open Sound Control. This would enable us to actually use human Motion in real time to generate musical signatures. So, essentially we could tag certain moves and movements into musical notes. We built a system to make this possible using the OpenNI, the NITE Middleware, the Open Sound Control Library and the Open Frameworks Library. Using all this available frameworks, we built a system that can essentially permit human agents to interact with an application using their motion and create music seamlessly. This system uses the Kinect™ and a computer as its hardware components and hence is very portable and inexpensive to use.

2.2 Human Motion Capture System with Kinect™

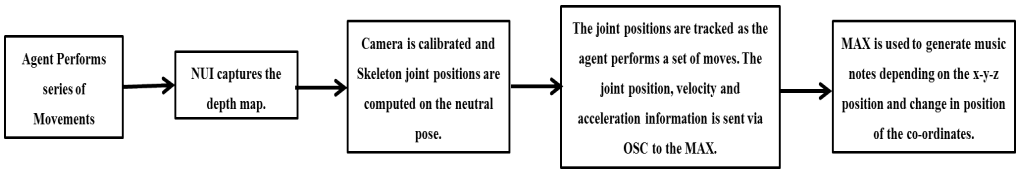

Figure 2.2. Motion To Music Working Flow Chart.

We present all our software systems at [3]. We also present a work-flow of the final system that we have used to tag moves with music in Figure 2.2.

2.3 Music Generation

When approaching the OSC Skeleton application we wanted a solution that was accessible and interesting. The goal was that each joint should have its own customizable sound source and that the performer and audience should easily be able to discern the sound changing and have a general idea of which sounds are coming from which joints.

Joint name
User Id
”confidence of the joint position co-ordinate”
/xjoint
/yjoint
/zjoint
9 values of the joint orientation matrix.
”confidence of the joint orientation”

Table 2.1. Skeletal Joint Information as sent over OSC

The entry point of this project is an application called Max/MSP or Max for short. Max is a visual object oriented programming language which has three cores. The first core is the Max core which handles mathematic functions. The second core is MSP which is used for signal processing to generate sound and manipulate existing sound. The thirds core is Jitter which is used for video processing. All of the cores are fully accessible from application which makes Max a very powerful multimedia visual language.

The software OSC Skeleton [3] sends Open Sound Control or “OSC” data packets through the local network. OSC is an ideal method of passing data because unlike MIDI, it can be passed very easily over the local network connection. The first step in building the Max patch receiver for OSC Skeleton is the unpacking process. OSC Skeleton sends data in a particular way. Information for all joints sent from the Kinect to the OSC is as shown in Table 2.1. The first function seen in Figure 2.3 tells the program to receive all UDP data on port 3333 and route everything under the joint heading along the path of j which stands for joint.

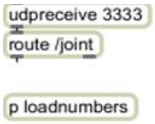

Figure 2.3. Joint Information Routing Building Block of Motion-to-Music Appication

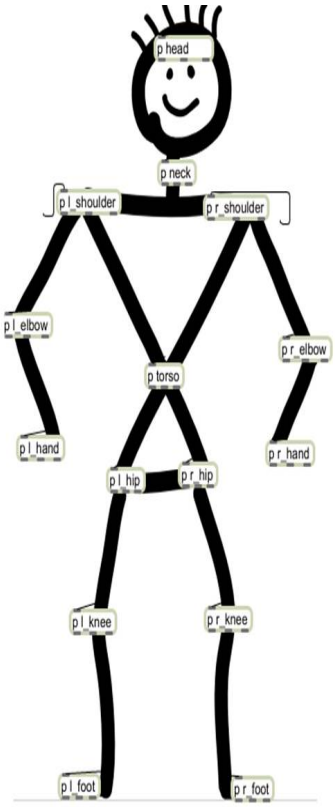

Figure 2.4. Visual Aide of Motion-to-Music Appication

Next, Figure 2.4 shows a visual aide that is constructed to assist in organizing the unpacking of each of the 15 joints. This hand drawn stick figure helps to better visualize how the Kinect is tracking the agent, and where the joints are located on the body. Each one of the boxes seen in Figure 2.4 receives the *joint* data and unpacks it in a *sub-patch*, which allows users to create programs or *patches* inside of an existing patch. The sub-patches that unpack the *joint* data look like what is in Figure 2.5. As we can see, the only data being unpacked for this project is the position of the X, Y, and Z coordinates of the joints,

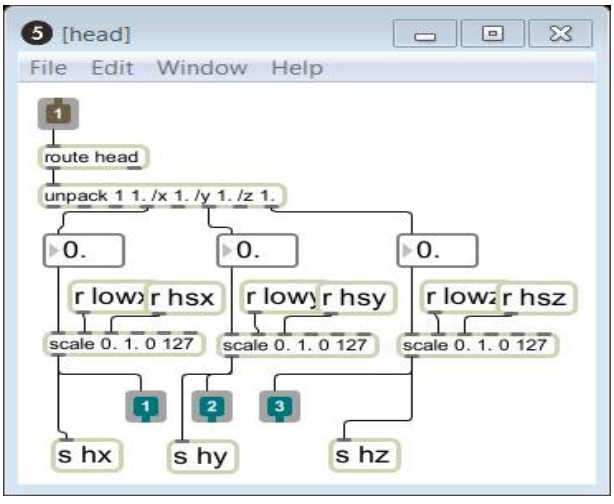

Figure 2.5. Motion To Music Application Sub-patch.

and not the orientations. These values are then packed into the range of 0 and 127 which is the standard range for MIDI. This is done for simplification purpose and to allow better interaction with components inside of Max and also for quick redirecting of data to programs outside of Max.

The last part of the Figure 2.5 sends the three values (X,Y,Z) to another sub-patch, seen in Figure 2.6. This sub patch receives the XYZ data and filters it through switches, which can be globally and locally activated and deactivated. This allows one to easily turn on or off and join with one click. Next, the XYZ values are visualized to give the composer and performer feedback.

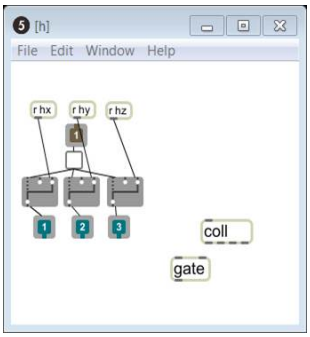

Figure 2.6. Another Motion To Music Application Sub-patch.

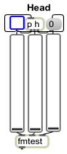

Figure 2.7. Visual Panels for each Joints

One of the visual panels as shown in Figure 2.7 is created and labeled for each of the 15 joints. The blue box in the top left is a toggle switch. When the box is empty, the joint is inactive. When one clicks the box, the joint becomes active. The final part of the patch is passing the data from the sliders to another sub patch which takes the values and generates sound. The type of sound being generated is called *Frequency Modulation* which takes a carrier frequency, modulator frequency, and amplitude to generate a complex waveform.

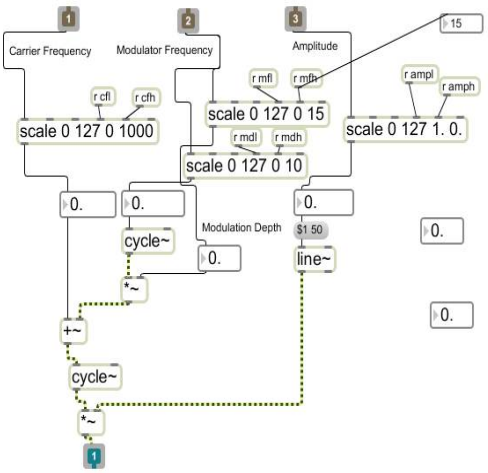

Figure 2.8. Frquency Modulation

As shown in Figure 2.8, the three values of X-Y-Z are assigned respectively to carrier frequency, modulator frequency, and amplitude. Each joint has a dedicated frequency modulation sound generator allowing them to act as unique instruments.

After the sound is generated, it is passed to two sliders which act as stereo volume control. Also, all of the scaling for the incoming XYZ values and respective carrier frequency, modulator frequency, and amplitude can be scaled easily in the final patch along clearly labeled along the side walls. The final patch (with the sub patches hidden) looks like in Figure 2.9.

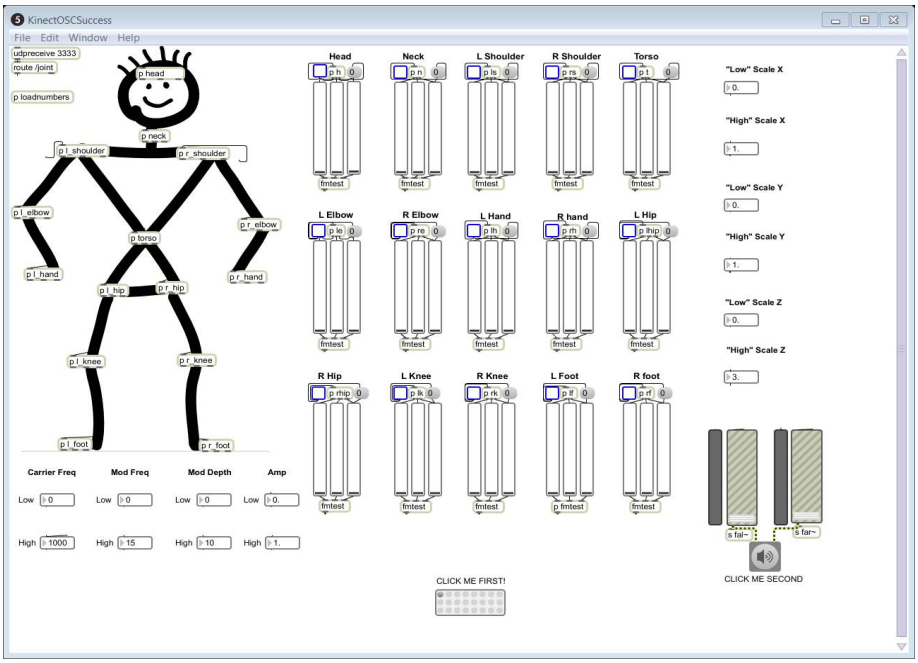

Figure 2.9. Final patch for the MAX application.

2.4 Results

To check out a demo of this application and generate musical notes as you perform a set of movements using the Kinect™ and our systems, please visit our project website [3].

Chapter 3

Tagging Moves into Human Interactions

3.1 Introduction

Next to verbal language what human beings often use as a mode of interaction is their body language. And hence is the interest in trying to recognize human gestures, expressions, micro-expressions and also to use them as cues for building interactive applications. Human-gesture driven applications has gained hung popularity in todays commercial market. From gesture to action to interaction, human motion has many application is video surveillance, video-event annotation, virtual reality, human-computer interaction, and robotics. Here we propose a system to automatically annotate a sequence of human movements with an interaction class.

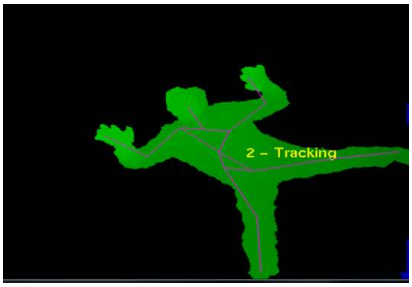

Figure 3.1. Silhouette from Skeleton Tracking

Recognizing human interactions has been a challenging task due to the overwhelmed dependency on low-level vision algorithms that include segmentation and tracking of salient image regions and extraction of object features.

In this work, we try to overcome some limitations of human interaction recognition by utilizing infrared range sensor quipped technology available commercially and inexpensively with the Kinect console.

3.2 K-10 Interaction Dataset

3.2.1 Description

In this work, we present a dataset of two-person interactions captured with the Kinect videogame Console interfaced with the computer. In the present day, motion capture is usually done with a marker-based system. A performer wears markers near each body joint to identify the motion by the positions or angles between the markers [23]. Acoustic, inertial, LED, magnetic or reflective markers, or combinations of any of these, are tracked, optimally at least two times the frequency rate of the desired motion to finally output a robust motion tracking for the participating agents. An example of a widely used database for such motion capture data is present at the CMU Graphics Lab Motion Capture Databases [24] and for building this database they have used a Vicon motion capture system consisting of 12 infrared MX-40 cameras, each of which is capable of recording at 120 Hz with images of 4 megapixel resolution. Motions are captured in a working volume of approximately 3m x 8m. The capture subject wears 41 markers and a stylish black garment. But the recent advances on imaging hardware and computer vision algorithms had led to the emerging technology

of markerless motion capture using a camera system. The commercial solution for markerless motion capture currently available in the market is Microsoft's Kinect videogame console. Using a Kinect to obtain Motion analysis data is affordable, portable and robust enough to evaluate the performance of Computer Vision Algorithms. To use the Kinect videogame console for Xbox 360, we had to first interface it with a PC. For that we have used the OpenNI ™ framework [4] and NITE Middleware from PrimeSense [5]. We then used our setup to record video clips of 2-person interactions. We made sure that the lighting conditions are optimal to give a robust tracking output and recorded both the image map and the depth map of the scene. The recorder records depth and image frames at 30 FPS [25]. Later these depth frames are used to extract the skeleton tracking data and record the joint position co-ordinates which are used in our algorithm for classification purposes. All our software systems for recording video, extracting skeletons at key-frames and playing the frames with skeleton tracking from stored information on joint positions is available on the project website [3]

3.2.2 Specifications

We worked with 10 classes of two person interactions. The clips were recorded with 6 participating agents. There are 25 different sets of agents performing the same interactions. In Table 3.1 we give the name of the Interaction classes and the number of clips we have for each class. We have tried to work with interactions where the human body orientation is either frontal or sub-frontal to the Kinect sensors and are un-occluded for most of the interaction sequence. The complete dataset is available on our project page.[3]

Interaction Class	Number of Clips
Approaching	57
Departing	55
Pointing	47
Pushing	55
Kicking	55
Punching	48
Exchanging objects	47
Walking Hand in Hand	25
Hugging	27
Shaking Hands	55

Table 3.1. K-10 Interaction Dataset Specifications

3.3 Our Approach

Firstly, we interface the Kinect console using OpenNI framework and NITE middleware with the computer. We then use our system to capture video clips of two-person interactions. Then these recordings are played and key frames for all classes of interactions are annotated manually. Finally we have our dataset of video clips containing image and depth maps for 10 different interactions. With the raw skeleton joint positions (in reference to the world co-ordinate system) extracted from the system, we delve into getting a spatial representation for these joints over time. Now, our skeleton recorder gives the x-y-z joint position for 15 joints [5] of the two interacting agents per frame. And we compute feature vectors over these joint positions (converted to viewport co-ordinates) and use Support Vector Machine classification schemes to recognize interactions. We have done experiments with fea-

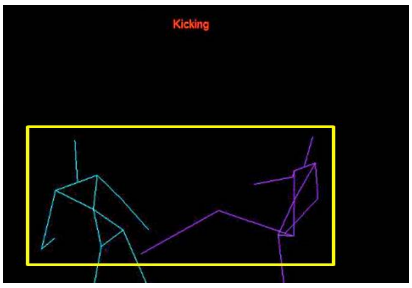

Figure 3.2. Interaction Example

tures like Euclidian distances between every pair of joints, correlation between every pair of joint-distances (both the participating agents), inverse correlation between every pair of joint-distances (both the participating agents), correlation between all joint-positions (both the participating agents), inverse correlation between all joint-positions (both the participating agents) and Euclidian Euler-angle distances between every pair of joints. Among all these, the Euclidian distance (position and Euler angle) feature between every pair of joints showed fair results. But, later we inferred, only the Euclidian distances between every pair of joints give much better classification results. And we think this is in relation with the not-so-confident capture of the joint orientation during skeletal tracking of participating agents [5]. So, we decided to make our classification schemes use the Euclidian distance between every pair of joint distances as feature vectors. To capture the temporal aspect of a set of moves that plays the vital role in human interaction recognition, we used a defined window over a set of frames and use that window of frames as a single feature-vector for our template-based classification.

We use one-VS-all SVM classification scheme for our interaction recognition algorithm. We have experimented with both linear and *rbf* kernel and found the later giving better performance.

3.4 Training and Testing

For training and testing we have used clips from the K-10 Interaction Dataset. Table 3.2 shows the number of clips for each of the interaction classes.

Interaction Classes	Number of (Frames)
Approaching	1018
Departing	814
Pointing	484
Pushing	322
Kicking	277
Punching	238
Exchanging objects	485
Walking Hand in Hand	322
Hugging	430
Shaking Hands	412

Table 3.2. Training & Testing Specifications

Even though we used our very own dataset to evaluate the results of this algorithm, since we use as features joint distances of 15 major skeletal joints of a human body, our system can essentially be used with any other conventional MoCap dataset that captures two person Interactions over a time frame. Again our system can be trained and evaluated with any other user-defined classes of two-person interactions where the user has a Kinect to record videos of those interactions or set of movements. We present all our software systems on our project page with which any user having a Kinect can define his own set of two-person interactions and train the classifier for automatically annotating them.

3.5 Canonical Time Warping

Template matching method for performing human motion analysis compares features extracted from the given image sequence to the pre-stored patterns dur-

ing the recognition process. A well-known advantage of using the template matching technique to do human motion analysis is its inexpensive computational cost. But one of the problems associated with this method is that it is relatively sensitive to the variance of the movement duration. And thus, many prefer to use the alternative state-space models that define each static posture as state and connect these states by certain probabilities. A motion sequence is then seen as a composition of these poses and a particular one considered as a tour going through certain states.

We tried to come around this problem by posing the difference in movement duration as an alignment problem. We used the method developed in [1] to align our training set clips with each other. This helped in better classification results. Zhou et al. showed the effectiveness of Canonical Time Warping in alignment of motion capture data of two subjects performing similar actions in [1]. Their results confirm that CTW provides qualitatively better alignment than state-of-the-art techniques based on DTW. We experimented with the alignment algorithm in the following two ways.

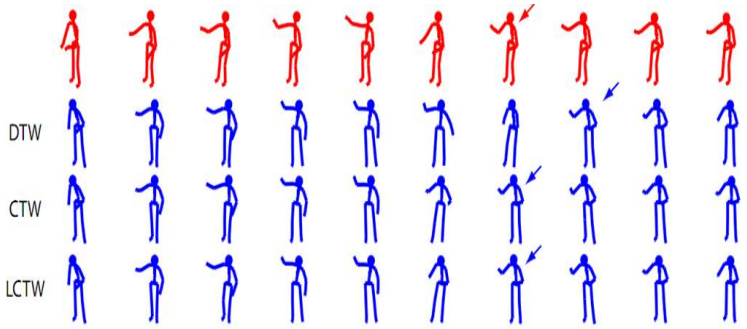

Figure 3.3. Canonical Time Warping Algorithm applied to Motion Alignment *Image Credit: [1].*

In one of our experiments, we aligned video clips from both the training and the test set with a base sequence. We choose the base sequence as the longest

available sequence for an interaction class (largest number of frames). For the training set, all the interactions from a certain class were aligned with the base sequence of that class. For the test set, each of the interactions were aligned with each of the base sequences and tested against the respective classifier. For example, a test clip "0001" is aligned with the clip "02Base" and tested against the "02 Class classifier" to see how confidently it is classified to the "02 class". Again the clip "0001" is aligned with the clip "03Base" and tested against the "03 Class classifier" to see how confidently it is classified to the "03 class". The most confident result decides the interaction class a test clip is entailed to.

In another experiment setup, we only align the training set clips. This experiment surprisingly gives us a better performance on the classification results. The reason, it appears, is when we try to align a clip like pushing with approaching; the CTW algorithm makes sure that the frames in the test clip that has more properties similar to the base sequence take the major length of the sequence. As a result, the classification performance drops slightly in the first set of experiments from the unaligned experiments while the later experiments perform way better.

We furnish the results from all the experiments we performed to give a better overview on the classification performance.

3.6 Results

In this section we furnish our results for interaction recognition. Among all our experiments we cite the results from these three set of experiments:

- **Experiment 1** deals with unaligned training and test set frames with Euclidian joint distances calculated among every joint pairs as features.

Frames with a window size 3 is considered to construct a feature vector and a rbf-kernel SVM is used for building one-VS-all classifier.

- **Experiment 2** deals with aligned training and test set frames with Euclidian joint distances calculated among every joint pairs as features. Frames with a window size 3 is considered to construct a feature vector and a rbf-kernel SVM is used for building one-VS-all classifier.

- **Experiment 3** deals with aligned training set and unaligned test set frames with Euclidian joint distances calculated among every joint pairs as features. Frames with a window size 3 is considered to construct a feature vector and a rbf-kernel SVM is used for building one-VS-all classifier.

3.6.1 Confusion Matrices

Here we present the Confusion Matrices from the above mentioned 3 set of Experiments. All uses Euclidian joint distances calculated among every joint pairs as features. Frames with a window size 3 was considered to construct the feature vectors and an (rbf-kernel SVM) one-VS-all classifier was built for each classes.

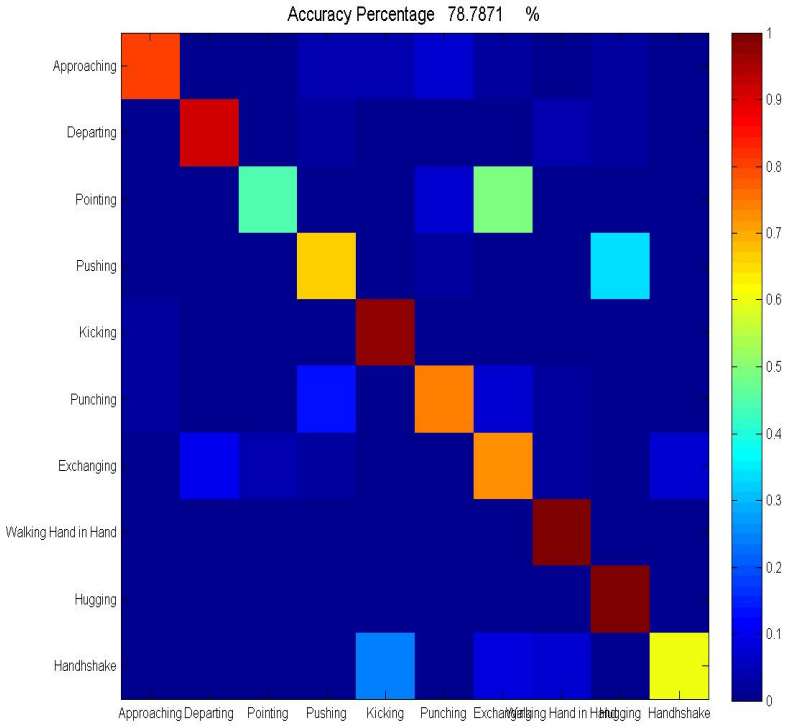

Figure 3.4. Confusion Matrix for Interaction Recognition with 10 classes in Experiment 1 **Average Accuracy: 78.79%**

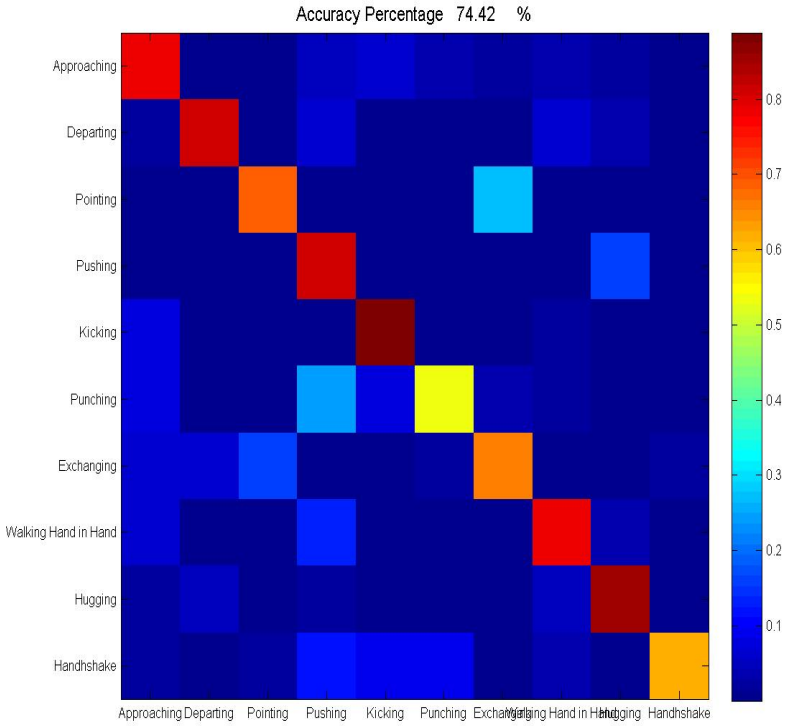

Figure 3.5. Confusion Matrix for Interaction Recognition with 10 classes in Experiment 2. **Average Accuracy: 74.42%**

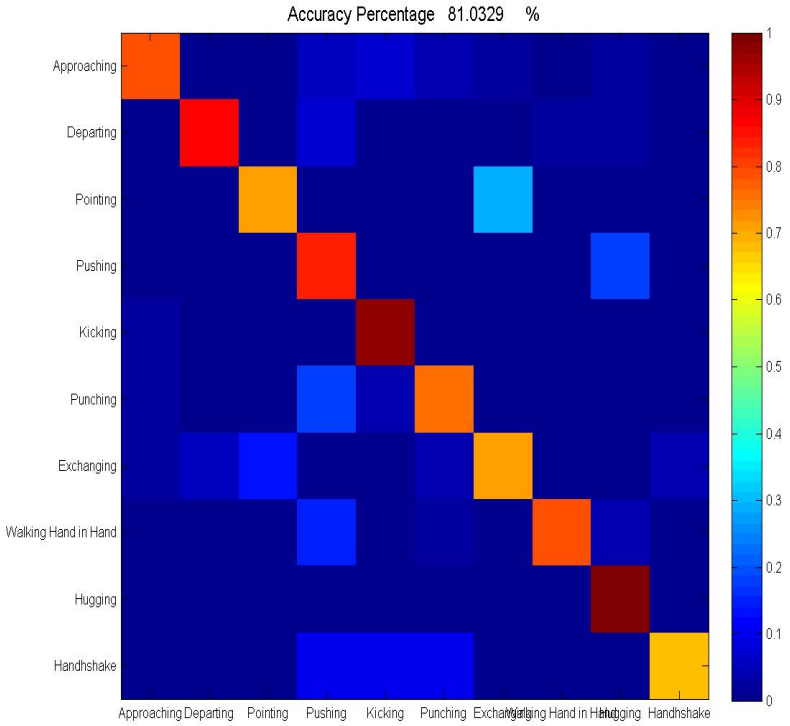

Figure 3.6. Confusion Matrix for Interaction Recognition with 10 classes in Experiment 3. **Average Accuracy: 81.03%**

3.6.2 Precision Recall Curves

Here we present the PR curves from the Experiment 3. These PR curves are for the individual interaction classifiers. A better sense of the classification accuracies can however be made from the confusion matrices, where it shows which class gets confused with another.

Experiment 3

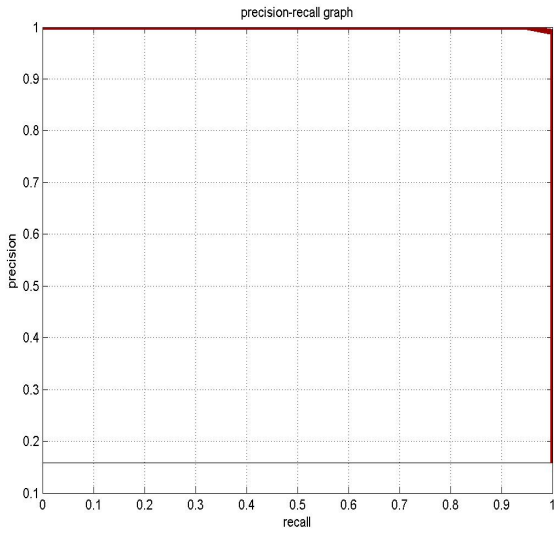

Figure 3.7. PR curve for Approaching Classifier.

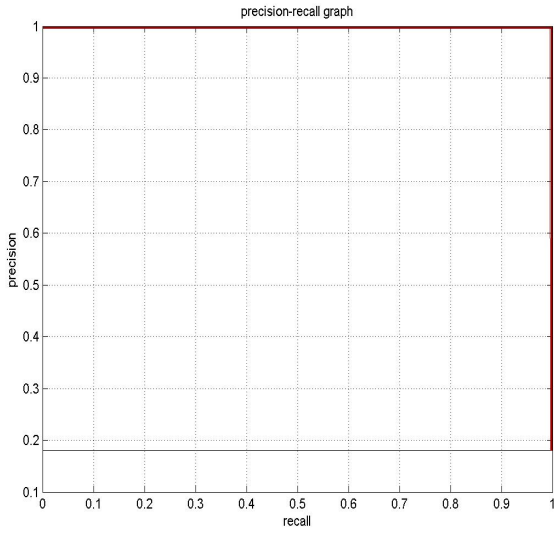

Figure 3.8. PR curve for Departing Classifier.

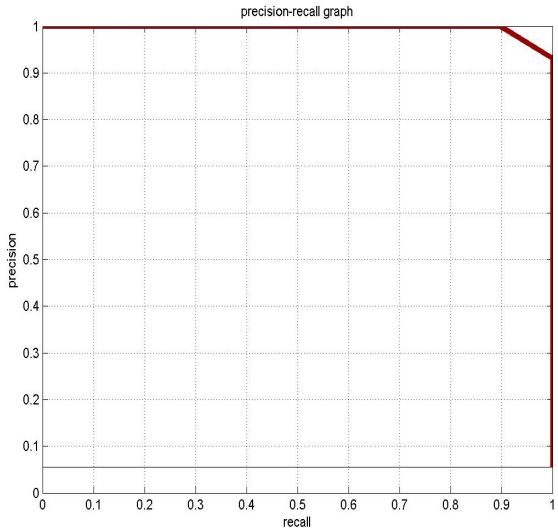

Figure 3.9. PR curve for Pointing Classifier.

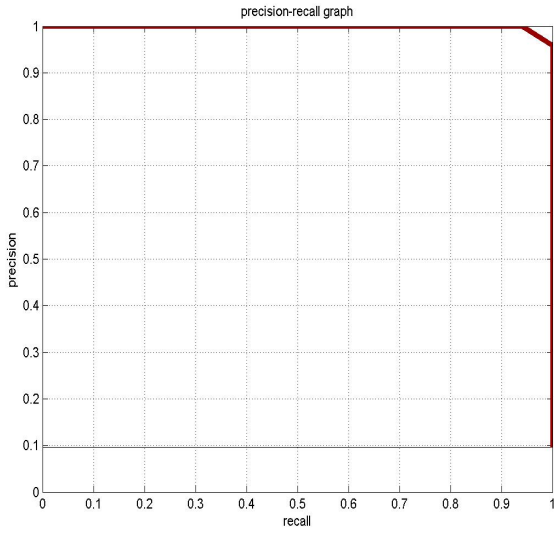

Figure 3.10. PR curve for Pushing Classifier.

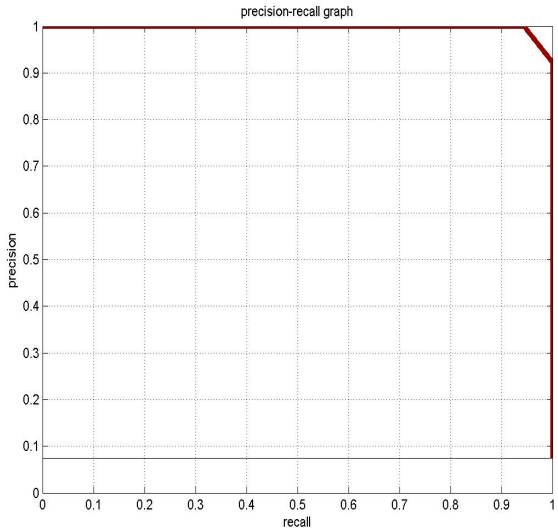

Figure 3.11. PR curve for Kicking Classifier.

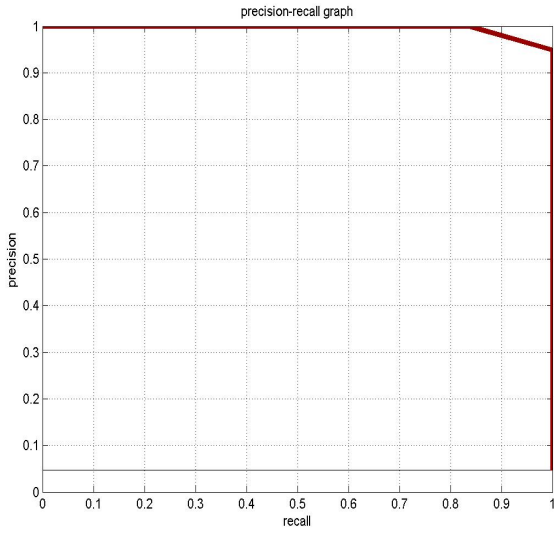

Figure 3.12. PR curve for Punching Classifier.

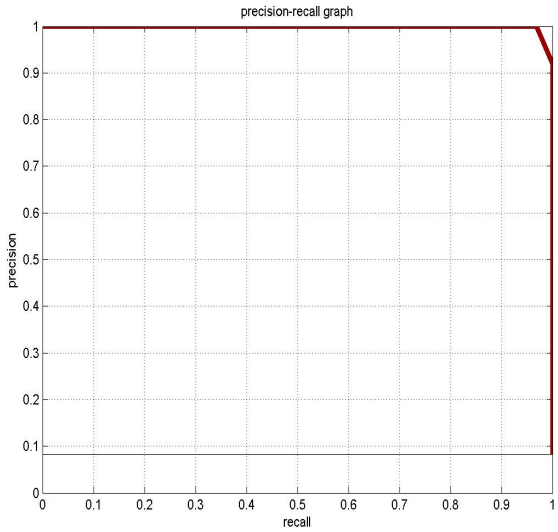

Figure 3.13. PR curve for Exchanging Classifier.

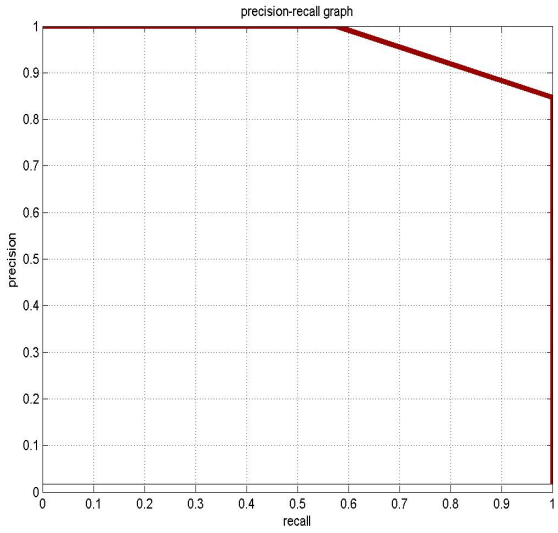

Figure 3.14. PR curve for Walking Hand in Hand Classifier.

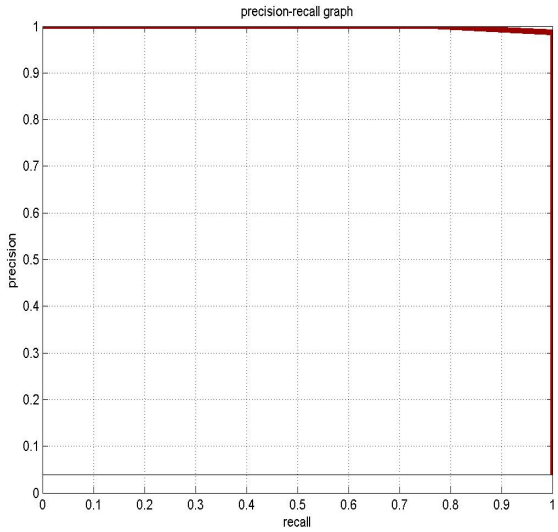

Figure 3.15. PR curve for Hugging Classifier.

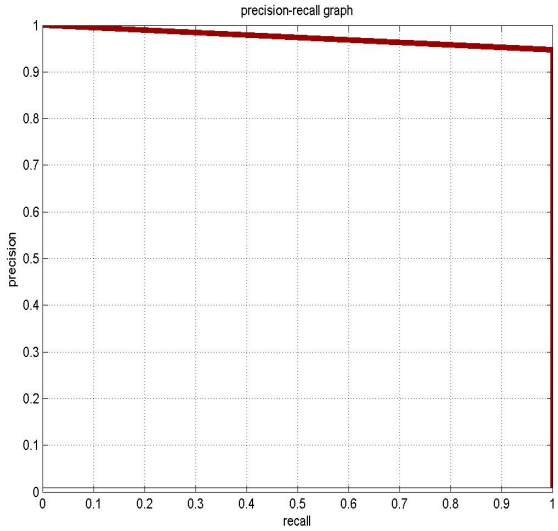

Figure 3.16. PR curve for Shaking Hands Classifier.

3.6.3 Performance Evaluation

In this section we present the performance accuracies for all our three sets of Experiments. We present the recognition accuracy for each interaction class in each table. We also computed the performance considering the fact that some interactions have overlapped characteristics. Like in *pushing*, there will be always *departing* interaction present, which is not wrong. So, we tried to observe how much the accuracy changes if we consider the correct decision (ground truth, in our case human annotated interaction labels) to be in Top-1 predicted class, Top-2 predicted classes, Top-3 predicted classes, Top-4 predicted classes and Top-5 predicted classes where the prediction values are ranked by their probabilities to belong to a certain class. In the following tables, *rank levels* indicate how many classes we considered for the right decision.

One noticeable thing is that for the class *Shaking Hands* the accuracies do not increase with the rank. We found that that the right decision for this class is either in rank 1 or in rank later than 7. This can be reasoned as this class is confused with about 5 different classes with noticeable probabilities.

Interaction Class Name	Rank 1 Acc. %	Rank 2 Acc. %	Rank 3 Acc. %	Rank 4 Acc. %	Rank 5 Acc. %
Approaching	80.98	91.34	96.09	99.32	100.00
Departing	91.67	95.49	98.19	98.42	100.00
Pointing	44.57	72.28	86.41	98.37	100.00
Pushing	65.63	96.88	99.22	100.00	100.00
Kicking	97.32	100.00	100.00	100.00	100.00
Punching	73.75	92.50	97.50	100.00	100.00
Exchanging objects	73.09	92.83	95.07	98.21	100.00
Walking Hand in Hand	100.00	100.00	100.00	100.00	100.00
Hugging	100.00	100.00	100.00	100.00	100.00
Shaking Hands	60.87	67.39	67.39	67.39	67.39
Average	**78.79**	**90.87**	93.99	96.17	96.74

Table 3.3. Performance Accuracies *(Experiment 1)*

Interaction Class Name	Rank 1 Acc. %	Rank 2 Acc. %	Rank 3 Acc. %	Rank 4 Acc. %	Rank 5 Acc. %
Approaching	77.79	92.65	96.48	99.08	99.85
Departing	81.35	86.11	96.83	99.20	100.00
Pointing	69.03	88.49	96.90	97.35	97.79
Pushing	81.82	96.02	96.02	96.02	97.73
Kicking	88.83	90.86	91.88	91.88	92.89
Punching	54.09	68.85	69.67	69.67	69.67
Exchanging objects	65.92	83.89	86.14	88.76	89.89
Walking Hand in Hand	77.78	93.65	93.65	93.65	93.65
Hugging	86.05	86.05	86.05	86.04	86.05
Shaking Hands	61.54	61.54	61.54	61.54	61.54
Average	**74.42**	**84.81**	87.52	88.32	88.91

Table 3.4. Performance Accuracies *(Experiment 2)*

Interaction Class Name	Rank 1 Acc. %	Rank 2 Acc. %	Rank 3 Acc. %	Rank 4 Acc. %	Rank 5 Acc. %
Approaching	79.46	92.69	96.60	99.15	100.00
Departing	86.03	90.54	97.97	99.32	100.00
Pointing	70.65	92.39	100.00	100.00	100.00
Pushing	82.81	100.00	100.00	100.00	100.00
Kicking	97.98	100.00	100.00	100.00	100.00
Punching	75.00	92.50	93.75	95.00	95.00
Exchanging objects	71.75	92.83	95.52	97.76	99.10
Walking Hand in Hand	79.25	100.00	100.00	100.00	100.00
Hugging	100.00	100.00	100.00	100.00	100.00
Shaking Hands	67.39	67.39	67.39	67.39	67.39
Average	**81.03**	**92.83**	95.12	95.86	96.15

Table 3.5. Performance Accuracies *(Experiment 3)*

Now, we would like to cite the results as reported by the work [19] that is closest to our work. They worked with the interaction classes: (1) approaching, (2) departing, (3) pointing,(4) standing hand-in-hand, (5) shaking hands, (6) hugging, (7) punching, (8) kicking, and (9) pushing. They report an overall average accuracy of 78%. The accuracies of each of the interaction class classication are 100, 100, 67, 83, 100, 50, 67, 83, 50% respectively. We report a final accuracy for these classes as 79, 86, 71, 79, 67, 100, 75, 98, 83 % respectively and an average overall accuracy as 81.03%.

3.6.4 Example Frame Sequences

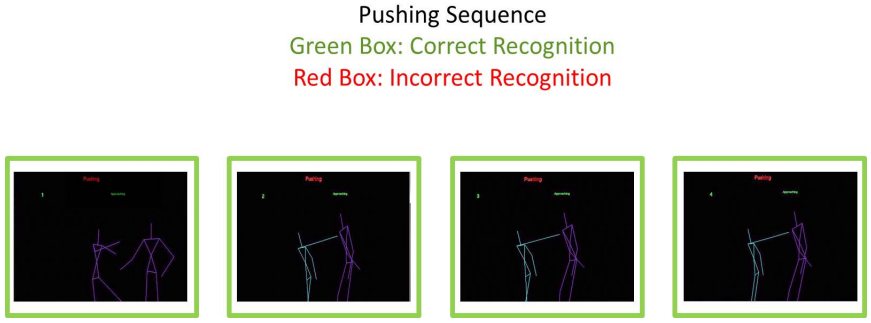

Figure 3.17. Pushing Sequence

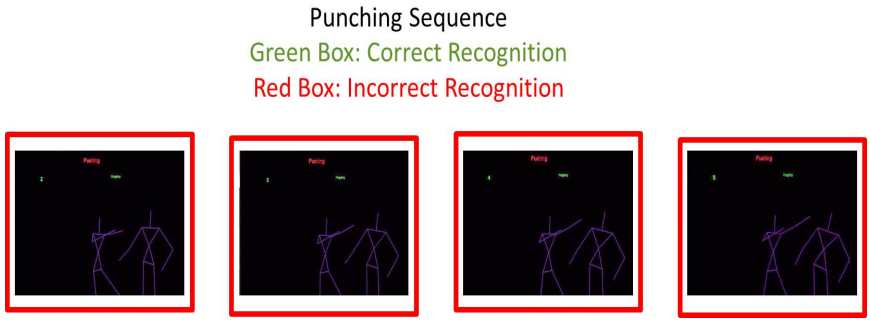

Figure 3.18. Punching Sequence

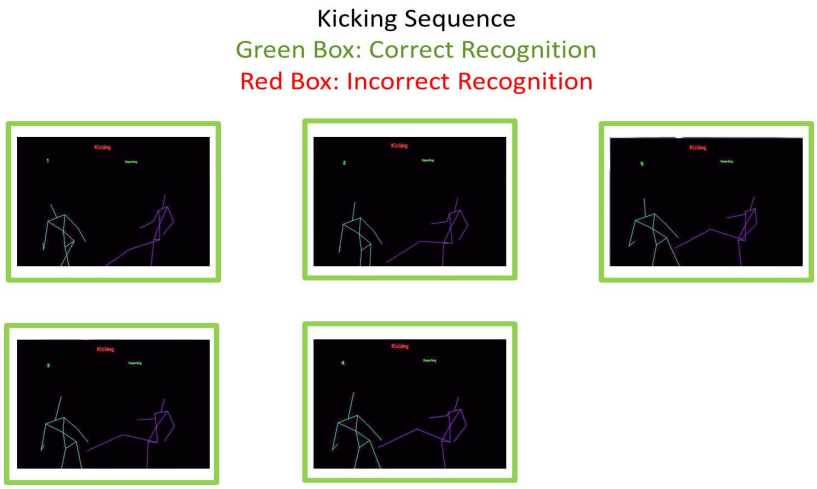

Figure 3.19. Kicking Sequence

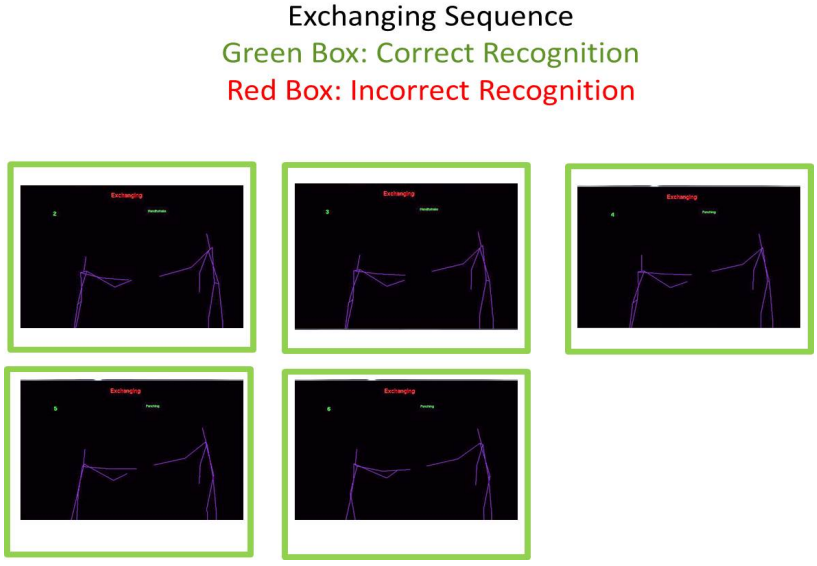

Figure 3.20. Exchanging Sequence

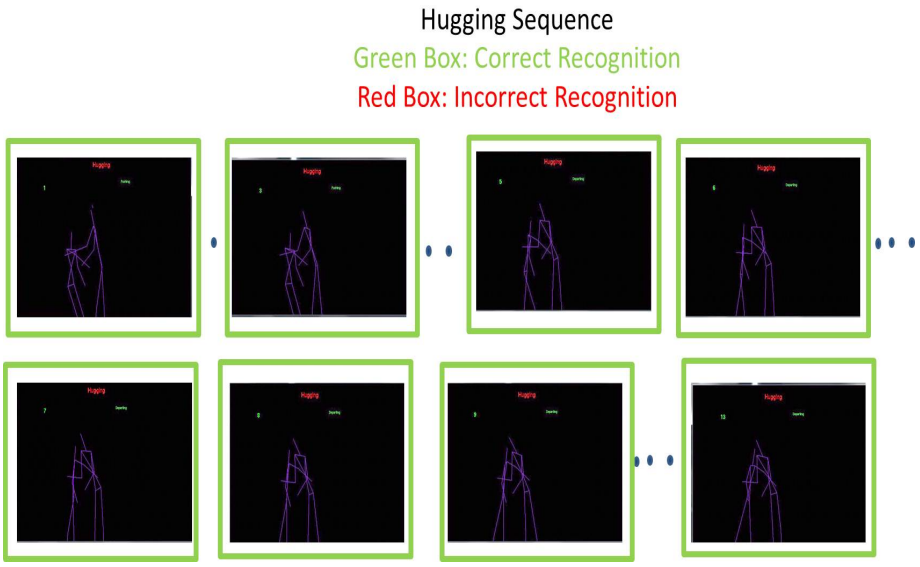

Figure 3.21. Hugging Sequence

Figure 3.22. Handshake Sequence

Chapter 4

Conclusion

This work is a human motion analysis study that essentially uses human motion for two different tasks: human computer interaction and human-human interaction. We use human motion to generate musical signatures in one of our HCI applications. The idea behind this project had been to use a markerless motion capture system , the Kinect videogame console, to capture human motion easily, portably and use it to control computer mediated applications like generating musical notes in a real time way. We had also used some recorded motion capture data to train an interaction classifier and used alignment algorithms to come around the relative sensitivity of template-based action recognition methods to the variation of movement duration. Our use of the CTW algorithms for alignment before the classifier learning improves its performance. The framework proposed here can be used for any MoCap dataset with two persons interacting over a time period. And also, we can define new classes of interactions and train new classifiers using this framework. All required software systems are available on the project page.

As a future work possibility, we plan to work on designing a better clas-

sifier using other Machine Learning approaches like Multiple Instance Learning with Boosting. The idea behind using MIL Boost is to do feature classification on joint distances as well as key frames for an interaction class. By doing boosting on bags of joint distances, we hope to learn both the joints playing important role in an interaction as well as the major frames in a sequence of moves that serves as a signature for the interaction.

There is also scope of future work in experimenting with the state space methods for interaction recognition on K-10 or a similar interaction dataset. Using probabilistic models to model connection among each state of static postures, it would be interesting to see how different the classification performance is. And as always applicable to algorithms using intrinsic non-linear models choosing a proper number of states and dimensions of a feature vector will control the overfitting and underfitting problems.

Our results show that the skeletal tracking algorithm available through the Kinect hardware and OpenNI and NITE libraries can boost some of the action recognition algorithms that uses template matching methods by its robust performance.

Bibliography

[1] ZHOU, F. and F. DE LA TORRE FRADE (2009) "Canonical Time Warping for Alignment of Human Behavior," in *Neural Information Processing Systems Conference (NIPS)*.

[2] JONES, K., K. JONES, and K. BARKER *Human movement explained*, Physiotherapy practice explained.

[3] CHATTOPADHYAY, D., T. VALLIER, T. BERG, and M. SCHEDEL (2011) "Multimodal Tagging of Human Motion Using Skeletal Tracking With Kinect™," .
URL `http://tamaraberg.com/kinect/`

[4] PRIMESENSE (2010) "OpenNI™," .
URL `http://www.openni.org/`

[5] PRIMESENSE (2010) "NITE™," .
URL `http://www.primesense.com/`

[6] HONING, H. "Computational Modeling of Music Cognition: A Case Study on Model Selection," *Music Perception: An Interdisciplinary Journal*, **23**(5), pp. pp. 365–376.

[7] WINKLER, T. (1995) "Making Motion Musical: Gesture Mapping Strategies for Interactive Computer Music," *Computer*, p. 261264.

[8] PARADISO, J. A., K. HSIAO, J. STRICKON, J. LIFTON, and A. ADLER (2000) "Sensor systems for interactive surfaces," *IBM Syst. J.*, **39**, pp. 892–914.

[9] GRIFFITH, F. M., N. (1998) "LiteFoot: A floor space for recording dance and controlling media." in *Proceedings of the 1998 International Computer Music Conference, International Computer Music Association*, pp. 475–481.

[10] PARADISO, J. A., K. HSIAO, A. Y. BENBASAT, and Z. TEEGARDEN (2000) "Design and implementation of expressive footwear," *IBM Systems Journal*, **39**(3.4), pp. 511 –529.

[11] LEE, E., U. ENKE, J. BORCHERS, and L. DE JONG (2007) "Towards rhythmic analysis of human motion using acceleration-onset times," in *Proceedings of the 7th international conference on New interfaces for musical expression*, NIME '07, pp. 136–141.

[12] WECHSLER, F. W., R. and P. DOWLING (2004) "EyeCon: A Motion Sensing Tool for Creating Interactive Dance, Music, and Video Projections," in *Proceedings of the AISB 2004 COST287-ConGAS Symposium on Gesture Interfaces for Multimedia Systems*, AISB '04, pp. 74–79.

[13] AGGARWAL, J. and Q. CAI (1997) "Human motion analysis: a review," in *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pp. 90 –102.

[14] SHOTTON, J., 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," .

[15] JUNEJO, I., E. DEXTER, I. LAPTEV, and P. PE ANDREZ (2011) "View-Independent Action Recognition from Temporal Self-Similarities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **33**(1), pp. 172 –185.

[16] JUNEJO, I. N., E. DEXTER, I. LAPTEV, and P. PÉREZ (2008) "Cross-View Action Recognition from Temporal Self-similarities," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pp. 293–306.

[17] ALLIN, S., N. BAKER, E. ECKEL, and D. RAMANAN (2010) "Robust Tracking of the Upper Limb for Functional Stroke Assessment," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **18**(5), pp. 542 –550.

[18] RAMANAN, D., D. FORSYTH, and A. ZISSERMAN (2005) "Tracking people and recognizing their activities," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, p. 1194.

[19] PARK, S. (2004) "A hierarchical bayesian network for event recognition of human actions and interactions," in *Association For Computing Machinery Multimedia Systems Journal*, pp. 164–179.

[20] PARK, S. and J. AGGARWAL (2004) "Event semantics in two-person interactions," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, pp. 227 – 230 Vol.4.

[21] BLEIWEISS, A., D. ESHAR, G. KUTLIROFF, A. LERNER, Y. OSHRAT, and Y. YANAI (2010) "Enhanced interactive gaming by blending full-body tracking and gesture animation," in *ACM SIGGRAPH ASIA 2010 Sketches*, SA '10, pp. 34:1–34:2.

[22] L. XIA, C.-C. C. and J. K. AGGARWAL (2011) "Human Detection Using Depth Information by Kinect," in *Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR (HAU3D)*, Colorado Springs, USA.

[23] WIKIPEDIA "Motion Capture Methods and Systems," .

[24] CMU "CMU Graphics Lab Motion Capture Databases," .

[25] PRIMESENSE (2010) "NITE™," .
URL `http://www.openni.org/documentation`