A COMPARISON METHOD TO EVALUATE MOTION SPLICING TECHNIQUES

by

GEORGE THEKKANATH RAPHAEL

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTERS OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
DECEMBER 2010

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Venkat Devarajan for his support and valuable suggestions. I am heartily thankful to my guiding professor, Dr. Gutemberg Guerra-Filho, for his guidance, support and encouragement at every stage of development in the research work. I would like to thank Dr. Mingyu Lu for serving in my committee

Last but not the least, I would like to thank my family and friends who helped me throughout my life.

December 20, 2010

ABSTRACT

A COMPARISON METHOD TO EVALUATE MOTION SPLICING TECHNIQUES

George Thekkanath Raphael, M.S

The University of Texas at Arlington, 2010

Supervising Professor: Venkat Devarajan

The development of methods for the re-use and modification of motion data is an active area of research in the field of Graphics, Image processing and human-like Animation. Among these methods, we analyze and evaluate different automatic techniques to generate spliced motions by combining different actions recorded in separate sessions. Motion splicing allows capturing motions independently and combining them later to create a new natural looking motion. Even though there have been much research in motion editing techniques, we have found no generalized quantitative evaluation of these edited motions. In this thesis, we present a novel methodology to quantitatively evaluate the synthesized motion generated by different motion editing techniques. We implemented three splicing algorithms to perform a comparison study based on our evaluation methodology. The splicing algorithms considered are spatial body alignment, segmentation-based and naïve DOF replacement. We use 39 sets of motions in the Human Motion Database, specifically collected for testing splicing. The motion sets consist of two actions performed individually and in combination with each other. This captured combination motions served as the ground truth in our comparison. The motions synthesized using the above splicing techniques were then evaluated using our quantitative evaluation method. The spatial body alignment performs best, since it accounts for the correlation between human joints. Due to poor segmentation, the segmentation-based method generated too much jerkiness in the synthesized motion, which was demonstrated with a poor result in our evaluation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

Computer graphics has been a powerful tool in creating virtual worlds and 3D character models. For a long time, animators have been trying to make these characters look real and move realistically, so as to recreate real world scenarios. In order to achieve realism in what concerns the motion of 3D characters, motion capture techniques were introduced to provide real motion data for animators. *Motion capture* is the process of recording a live motion event from a subject by tracking over time, a number of key points (or markers) on the skin of the subject, translating it into usable mathematical terms and combining them into a single digital three dimensional representation of the motion. The motion of the captured subject can later be transferred to a virtual world with a 3D character, in a process known as Motion-based Animation.

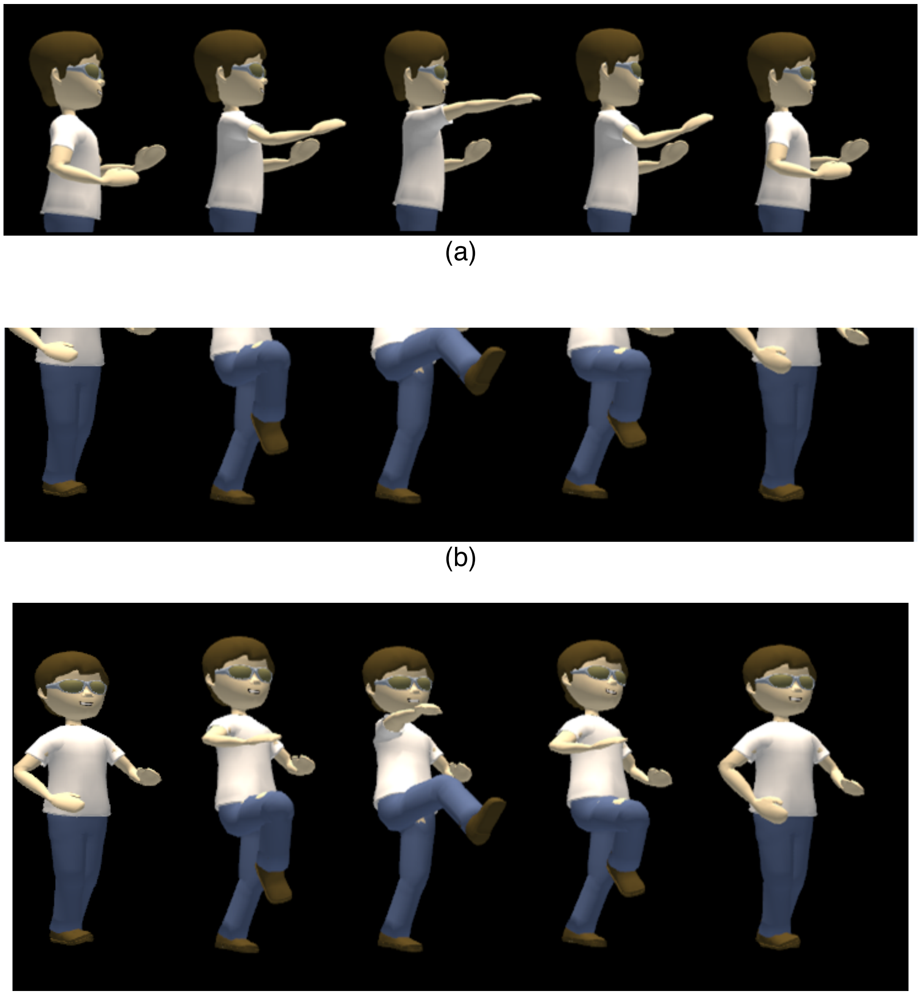
There are four major ways of generating animations: artistic (key frame animation), data driven, procedural, and hybrid approaches. All these approaches are the descendants of a primitive technique called rotoscoping. Rotoscoping was a technique to copy realistic motion from film footage onto cartoon characters. Among these different approaches, the artistic approach has been a successful technique in the game development industry. This technique involves an artist manually tweaking the various controllers attached to the character to create a frame-by-frame animation. The main advantage of this technique is that the animated character exactly fits the environment where it is supposed to act. In this case, the animator performs the validation of the generated motions. The main limitation of this technique is that it is time consuming and labor intensive.

More recently, artists have been aided by the introduction of motion capture. Motion capture uses sensors to observe and record the detailed motion of a set of objects or a subject.

Since motion capture records all the subtle nuances of real motion, it is a useful technique for creating realistic animations. Motion capture is less time consuming and cheaper than the artistic approach. Disadvantages of motion capture are that it requires skilled technicians for post processing and expensive equipment for data capture. Usually, eight infrared cameras are used for motion capture along with a set of optical markers. Furthermore, the captured motion is specific to a particular task in a particular environment.

Researchers are focused on editing existing motion capture data to synthesize new motion that can then be used in a different environment. Motion editing techniques can be categorized into retargeting [AF02], generalization [AG10], transitioning [KG02], and splicing [HK06]. Motion blending, warping and interpolation are some of the other fields supporting motion editing techniques. Thus, motion editing techniques have significant practical importance and several applications.

Motion splicing is the technique of combining different actions into a single motion. For example, given an action, where the subject punches and an action, where the subject kicks, a splicing method combines the punch action with that of the kick action to generate a single whole-body motion, where the subject simultaneously punches and kicks. Figure 1.1 illustrates the splicing procedure. Figure 1.1a shows the key frames in a punching action. Figure 1.1b shows the key frames in a kick action. Figure 1.1c shows the key frames in the spliced action.



(a)

(b)

(c)

Figure 1.1: Motion splicing of a simultaneous punch and kick. (a) Punching action.
(b) Kicking action (c) Spliced motion.

We have presented a novel technique to quantitatively evaluate and compare synthesized motions against actual captured motions. We used the Human Motion Database (HMD) **[**AG10**]** specially designed to provide data for the discovery of motion-editing methods and its quantitative evaluation. The database for the testing of motion splicing techniques contains a splicing dataset with 39 motion sets. Each motion set consists of two different actions performed individually and the combined simultaneous performance of these two actions. We used these motions as the ground truth for our evaluation.

In this thesis we implemented and performed detailed quantitative evaluations of the three most significant motion splicing algorithms to apply our quantitative evaluation methodology. The three motion splicing methods chosen based on our preliminary evaluation were *spatial body alignment* [HK06], *segmentation-based* [JL08] and *naïve degree of freedom replacement* [Per95]. Spatial body alignment is the method used to splice the upper body of one locomotive action to another. Segmentation-based method splices motion segments, whose partial motions are similar. Naïve DOF replacement method replaces a subset of the joints in one motion with the same joints in the other motion to form the spliced motion**.** The individual actions in the ground truth motion set are the input to these motion splicing techniques. Their synthesized spliced output is compared against the combined motion in the ground truth motion set.

Our quantitative evaluation methodology provides a measure of how well each algorithm splices based on their rotational and translational similarities. Another method for the evaluation of synthesized motions is the support vector machine (SVM) approach, in which the synthesized data is compared against a large set of training motion data, only to verify whether the synthesized motion can be classified as natural motion [IF04]. This method requires that the training data contain actions similar to the synthesized motion. Other limitation is the large computation time in the training and testing steps. An appealing aspect of our evaluation method over the SVM technique is that it compares the synthesized motion against the same exact captured motion instead of using a set of similar motions as the SVM technique.

The rest of the thesis contains details of the implementation of the above-mentioned splicing techniques and the application of our quantitative evaluation methodology to compare their results. In Chapter 2, we present an assessment on the mathematical model of the human body and explain the basics of motion capture. Chapter 2 also explains the kinetic and kinematic parameters associated with human movement along with the different approaches used in determining the body segment parameters.. The chapter ends with a summary of the approaches used in motion editing techniques like retargeting, blending, and splicing.

Chapter 3 describes each of the above-mentioned splicing techniques in detail.

Chapter 4 explains the details of our quantitative evaluation methodology, the various steps involved in comparing motions, and the significance of the ground truth data.

Chapter 5 presents our conclusions derived from our quantitative evaluation methodology and the significance of a future physics based approach in splicing motion data.

CHAPTER 2

BACKGROUND STUDY

In this chapter, we present an assessment of the mathematical model of the human body and explain the basics of motion capture. We present the kinetic and kinematic parameters associated with human movement along with the different approaches used in determining the body segment parameters. We explain center of gravity, moment of inertia, and various forces and torques linked with different body segments. We mention the significance of optical motion capture over key frame animation and different ways of representing the motion data. The chapter ends with a summary of the approaches used in motion editing techniques like retargeting, blending, and splicing.

2.1 Kinematics and Kinetics of Human Movement

Kinematics is concerned with describing and quantifying both the linear and angular positions of the bodies and their time derivatives. Kinematics is the preferred analytic tool that describes the range of motion of joints and provides input data for kinetic analysis. The movement of the body is captured using a set of markers attached to the body. The lengths of bones, the degrees of freedom associated with each joint, and their global position can define the whole body posture at any given time. Position, velocity, and acceleration are the primary variables associated with a given motion.

The position of a marker is represented in the 3D Cartesian coordinate system. The axis system can be right handed or left handed with either the *z*-axis pointing upwards (*z*-up format) or *y*-axis pointing upwards (y-up format). The number of independent parameters that uniquely define the location of a point is known as a degree of freedom (DOF). Figure 2.1 shows the 6 DOFs (t_x-position, t_y-position, t_z-position, R_x- rotation about the *x*-axis, R_y-rotation about the *y*-axis, R_z-rotation about the *z*-axis) of a joint in the 3D. The most common method for collecting kinematic data uses a motion capture system to record the motion of markers affixed to a moving subject. Analyzing these kinematic parameters gives us a better understanding of the characteristics of a motion.

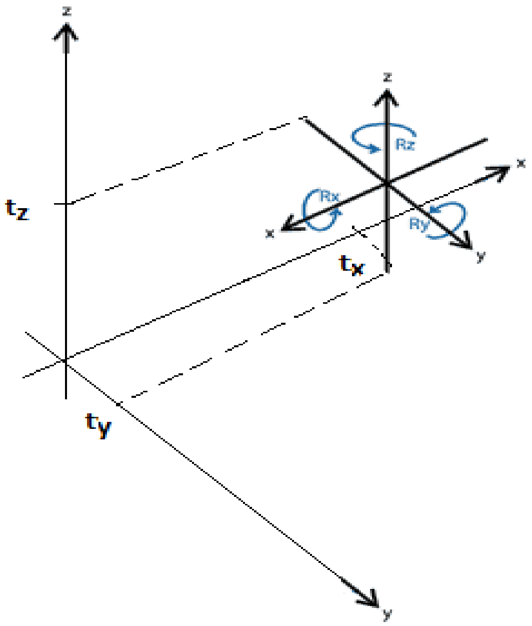


Figure 2.1: Six DOFs representing a joint.

On the other hand, kinetics deals with the causes of motion such as forces and torques. Kinematic data provides input values for inverse dynamics analysis to estimate the force and moment acting across different joints in the body. Forces and torques are computed by treating the whole body as a set of rigid objects connected at joints. In human movement, we are interested in two specific effects of force. They are the effects of force on rigid parts like bones and on deformable body parts like muscles.

Hanavan [HP64] modeled the body segments in the three-dimensional space as shown in the Figure 2.2, which later became one of the best mathematical models for the body segmentation. He was also able to derive principle mass moments of inertia from integral calculus, of various geometrically uniform dense solids relating to the shape of these segments.

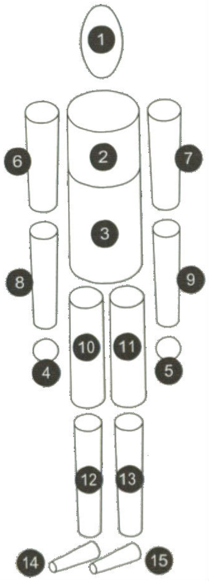


Figure 2.2: Mathematical model of the human body [HP64].

2.2 Body Segment Parameters

Inertial properties and physical characteristics of each rigid segment of the body have to be precisely determined before performing kinetic analysis. Inertial properties (mass, center of mass, moment of inertia) are difficult to determine for a living human being. Hence, studies were conducted on cadavers and properties were generalized. Dempster [DT55] created a table showing the segmental masses as proportions of the total body mass; the locations of the centers of gravity; lengths of radii of gyration as proportions of the segments lengths. The radii of gyration were used as an indirect means of calculating rotational inertia. Later, Clauser et al. [CM69] made slight modifications to Dempster's table and came up with the conclusions Figure 2.3 and Figure 2.4.

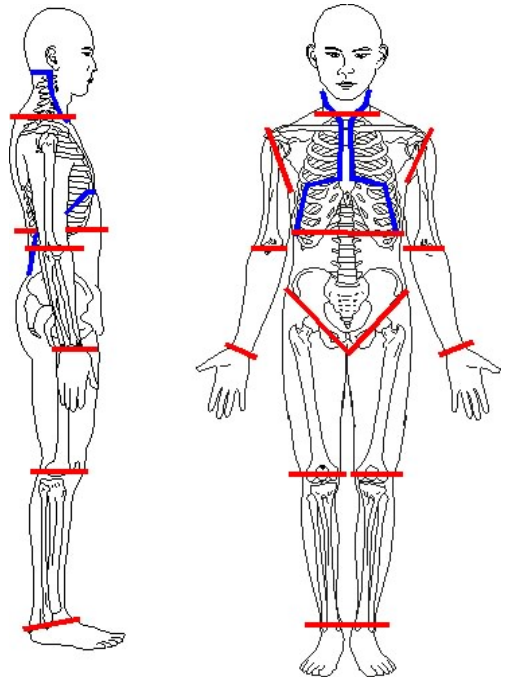
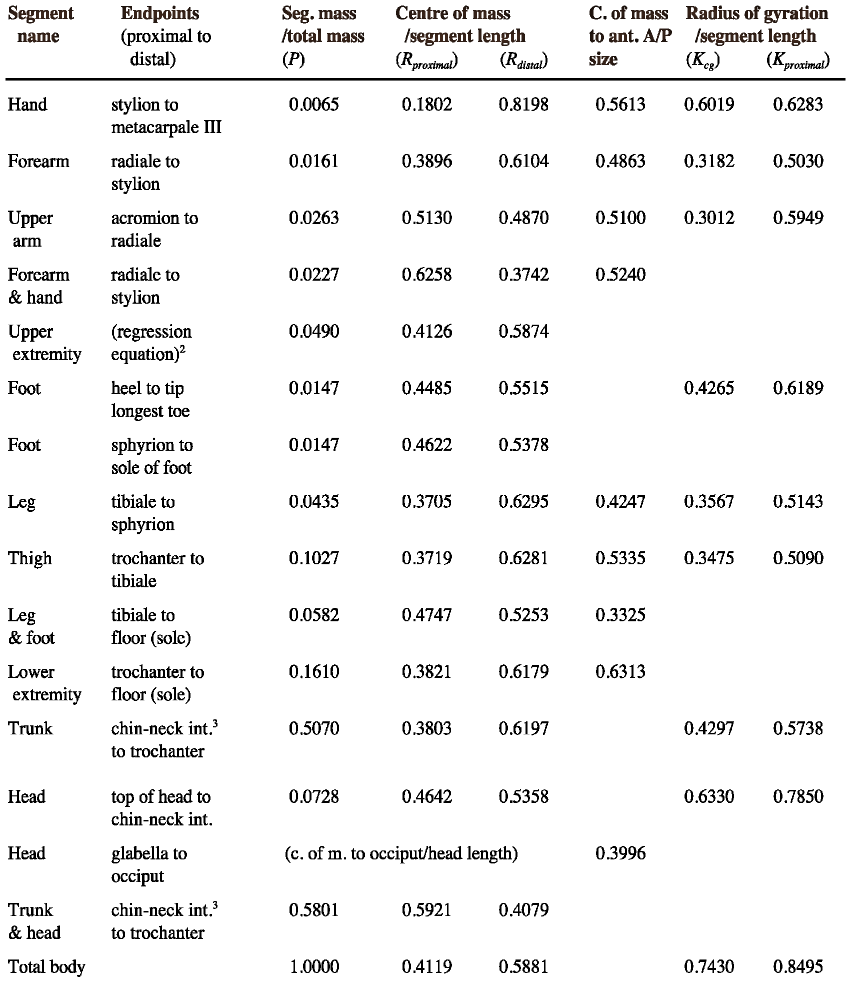


Figure 2.3: W.T Dempster's body segmentation [DT55].



Segment name	Endpoints (proximal to distal)	Seg. mass /total mass (P)	Centre of mass /segment length ($R_{proximal}$)	(R_{distal})	C. of mass to ant. A/P size	Radius of gyration /segment length (K_{cg})	($K_{proximal}$)
Hand	stylion to metacarpale III	0.0065	0.1802	0.8198	0.5613	0.6019	0.6283
Forearm	radiale to stylion	0.0161	0.3896	0.6104	0.4863	0.3182	0.5030
Upper arm	acromion to radiale	0.0263	0.5130	0.4870	0.5100	0.3012	0.5949
Forearm & hand	radiale to stylion	0.0227	0.6258	0.3742	0.5240		
Upper extremity	(regression equation)[2]	0.0490	0.4126	0.5874			
Foot	heel to tip longest toe	0.0147	0.4485	0.5515		0.4265	0.6189
Foot	sphyrion to sole of foot	0.0147	0.4622	0.5378			
Leg	tibiale to sphyrion	0.0435	0.3705	0.6295	0.4247	0.3567	0.5143
Thigh	trochanter to tibiale	0.1027	0.3719	0.6281	0.5335	0.3475	0.5090
Leg & foot	tibiale to floor (sole)	0.0582	0.4747	0.5253	0.3325		
Lower extremity	trochanter to floor (sole)	0.1610	0.3821	0.6179	0.6313		
Trunk	chin-neck int.[3] to trochanter	0.5070	0.3803	0.6197		0.4297	0.5738
Head	top of head to chin-neck int.	0.0728	0.4642	0.5358		0.6330	0.7850
Head	glabella to occiput	(c. of m. to occiput/head length)			0.3996		
Trunk & head	chin-neck int.[3] to trochanter	0.5801	0.5921	0.4079			
Total body		1.0000	0.4119	0.5881		0.7430	0.8495

Figure 2.4: Clauser's table for calculating body segment parameters [CM69].

The two most significant parameters required to calculate kinetic parameters are the center of gravity and moment of inertia (MOI). Finding the Center of Gravity and Moment of Inertia for different body segments is relatively more complex than for a rigid body. The center of gravity of the body segment in the *X* direction (X_{cg}) is computed as in equation 2.1.

$$X_{cg} = X_{proximal} + R_{proximal}(X_{distal} - X_{proximal}) \qquad 2.1$$

where X_{distal} and $X_{proximal}$ are the *x*-axis coordinates of the distal end of the segment (the one further from the body) and the proximal end of the segment (the one closer to the body), respectively, and $R_{proximal}$ is the distance from the proximal end of the segment to the center of the segment. In 3D, the moment of inertia is represented as a 3 x 3 tensor where the diagonal elements are called the principle mass moment of inertia (I_x, I_y, I_z) and the off–diagonal elements *(P_{xy}, P_{xz}, P_{yx}, P_{yz}, P_{zx}, P_{zy})* are called the mass products of inertia. The moment of inertia tensor is represented as in equation 2.2:

$$MOI = \begin{vmatrix} I_x & P_{xy} & P_{xz} \\ P_{yx} & I_y & P_{yz} \\ P_{zx} & P_{zy} & I_z \end{vmatrix}. \qquad 2.2$$

Although accurate body segment parameters are desirable, errors in these parameters may have little effect on the kinetic measurements due to the large magnitude of the reaction forces.

2.3 Forces and Torques

Force represents the action of one body on the other. The two major forces acting on a human body while performing an action are the gravitational force and the frictional force. Gravitational force acting on the body produces two forces: the weight of the body acting downward and the ground reaction force (GRF) exerted by the ground on the person. When a person is in a static rest pose, the weight of the body and the ground reaction force are equal. However, when the person starts exerting force by extending muscles, this equilibrium breaks and the ground reaction force increases above the body weight and throws the body out of equilibrium causing movements in the body. Other types of major forces acting on the body are the inertial force, centripetal force, centrifugal force, and the Coriolis force [BJ77].

A force in a body joint causes the joint to change its global position while the moment of force or the torque causes rotation in the joint. When the line of application of force does not pass through the axis of rotation, it produces a torque. Torque travels along the body and causes non-uniform changes through out the body. Analyzing the torque variation in the joints gives a detailed understanding of the correlation between the joints in the body. Applying different values of torques on the same joint and measuring the depth of travel gives a better understanding of the joints. This also helps when splicing various motions so as to get a measure of the change to be applied to various joints based on the values of torques at each joint. Pandya *et al.* [PM92] designed a model to predict the dynamic force resulting from multi joint motions. For a motion to be realistic, special attention should be given to the body balance and comfort control parameters while accounting for force and torques.

2.4 Key Frame Animations

Among the various methods of generating animations, key frame animations generated by artists are extremely controllable and environment compatible. Here, an animator assesses the required motion and generates key poses as well as break down poses. Standard animation tools to generate a complete motion clip later interpolate these poses. Interpolation (also known as tweeing) is the process of generating the intermediate poses between the artist-generated poses. The animator later edits these intermediate poses.

The main advantage of key frame animation is that it is extremely controllable. For every DOF in the body at every single frame, the animator manually sets the position. The crated animations are visually plausible and most probably follow physical limitations. The main limitation of this method is the time consuming labor involved in the production of any animation. For a single frame, the animator has to alter a large number of joints and also specify the time duration for a specific set of frames. The artist must have a good idea about the timing for the occurrences of different actions in a motion. For example, in a walking motion the time duration between the instant the heel touches the ground to the instant the toe leaves the ground is 60 percent of the overall walk cycle.

Animation systems have come up with helpful mechanisms to reduce the amount of animators' work. Hierarchical joint clustering [GJ96], joint trajectory controls [PM92], and skeleton manipulation [AG10] are some of the mechanisms currently available. Full body inverse kinematics [Gir87] helps the animators to get a better control over the physical nature of the motion. Figure 2.5 shows the interface from animation software.

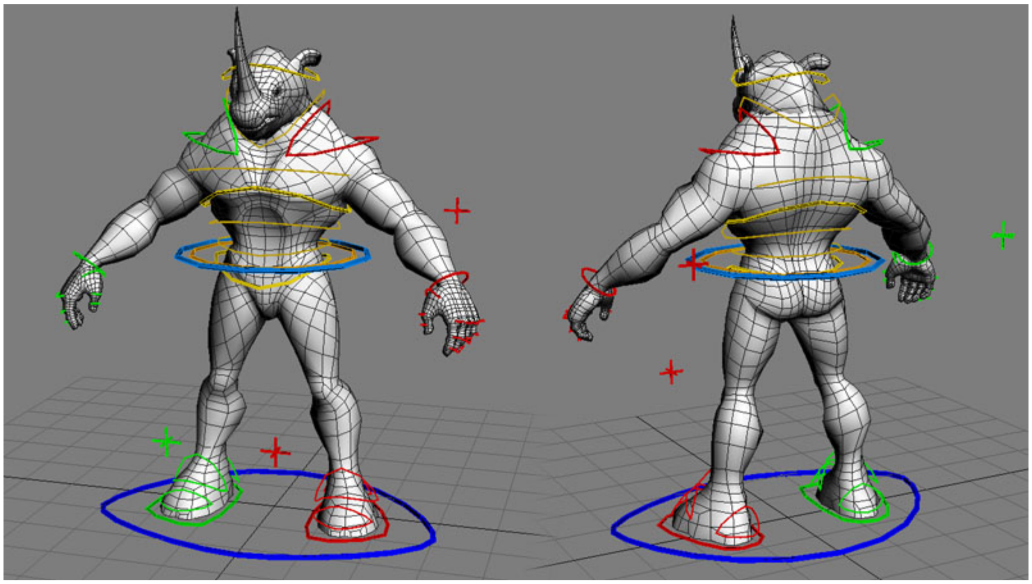


Figure 2.5: Key frame animation using animation software.

2.5 Optical Motion Capture

Motion capture is the process of capturing a motion performed by a subject. The motions are captured either using retro-reflective markers attached to the character performing the motion or using a motion capture suit that has the necessary sensors to record the joint angles while performing an action. For motion-based animation, the captured motion is then mapped into a virtual character with the same skeletal topology and similar limb proportions. Motion capture has a wide set of applications in fields such as robotics, animation, medicine, and kinesiology. Motion capture techniques allow the construction of a large database with human motions. In this thesis, we use the Human Motion Database [AG10] constructed in the Sensory-Motor Intelligence Lab at the University Of Texas at Arlington to support the evaluation of motion-based animation methods.

Requirements for skilled post-processing technicians and expensive capture equipment are some of the disadvantages of this system. Post-processing includes creating a skeleton, fitting the skeleton to the marker data, filling in occluded markers, smoothing noisy marker data, and solving for a character pose at every time sample.

The motion capture data is commonly represented in two different file formats. One format called the ASF/AMC format [*Gue*05], was developed by the Acclaim game company and the other called the BVH file format [*Gue*05] by Biovision Inc. Figure 2.6 shows a typical motion capture studio, where the subject is performing an action [*Mots*].

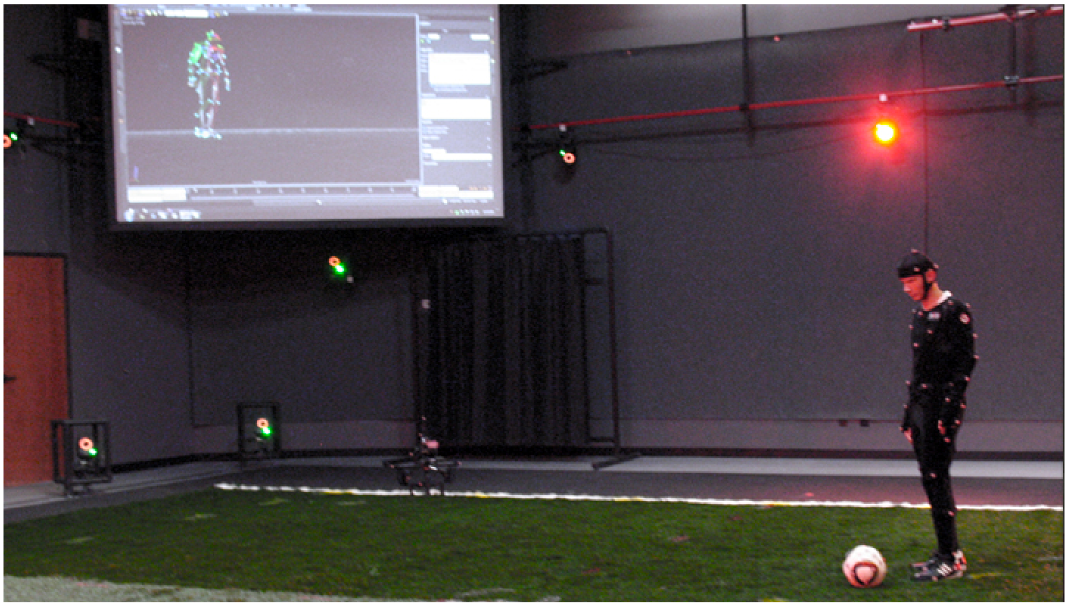
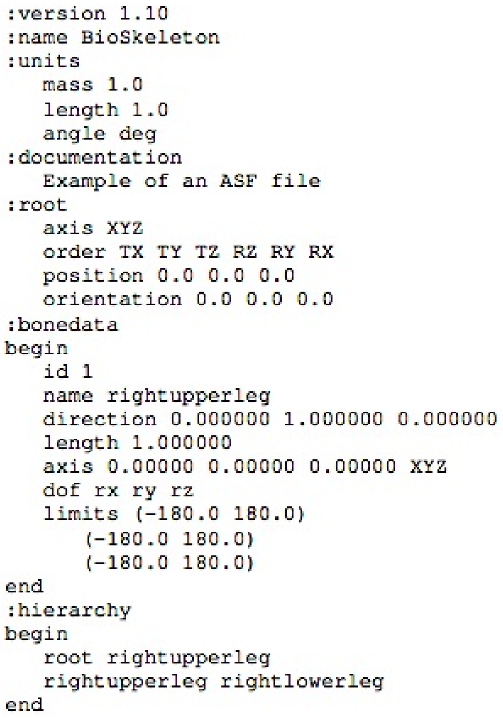


Figure 2.6: A typical motion capture studio

2.5.1 Acclaim File Format

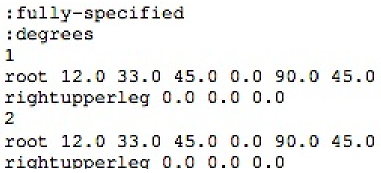
In the Acclaim file format, a single motion is recorded into two different files: the ASF file contains the skeleton information and the AMC file has the motion information for every single frame. For every bone in the skeleton, the ASF file has all the required information: name, identification number, degrees of freedom of the proximal joint, the limits on the range of movement of the proximal joint in every single DOF, and also the length and orientation to the distal joint or end site. The ASF file provides the information about the hierarchical ordering of the joints and all the information pertaining to the mechanical function of the skeleton, including units, multipliers, degrees of freedom, limits, and documentation. A typical sample of an ASF file is shown in Figure 2.7. The ASF file is divided into eight sections defined by the following keywords preceded by a colon: version, name, units, documentation, root, bonedata, hierarchy, and skin (sometimes).



```
:version 1.10
:name BioSkeleton
:units
   mass 1.0
   length 1.0
   angle deg
:documentation
   Example of an ASF file
:root
   axis XYZ
   order TX TY TZ RZ RY RX
   position 0.0 0.0 0.0
   orientation 0.0 0.0 0.0
:bonedata
begin
   id 1
   name rightupperleg
   direction 0.000000 1.000000 0.000000
   length 1.000000
   axis 0.00000 0.00000 0.00000 XYZ
   dof rx ry rz
   limits (-180.0 180.0)
      (-180.0 180.0)
      (-180.0 180.0)
end
:hierarchy
begin
   root rightupperleg
   rightupperleg rightlowerleg
end
```

Figure 2.7: A sample ASF file.

The AMC file on the other hand contains the motion data information for every frame. The frame number is declared followed by the bone name and data for every channel. A sample AMC file is shown in Figure 2.8. All the data in the AMC file is relative to the ASF file. The fields are sequenced in the same order as the DOF fields in the ASF file.



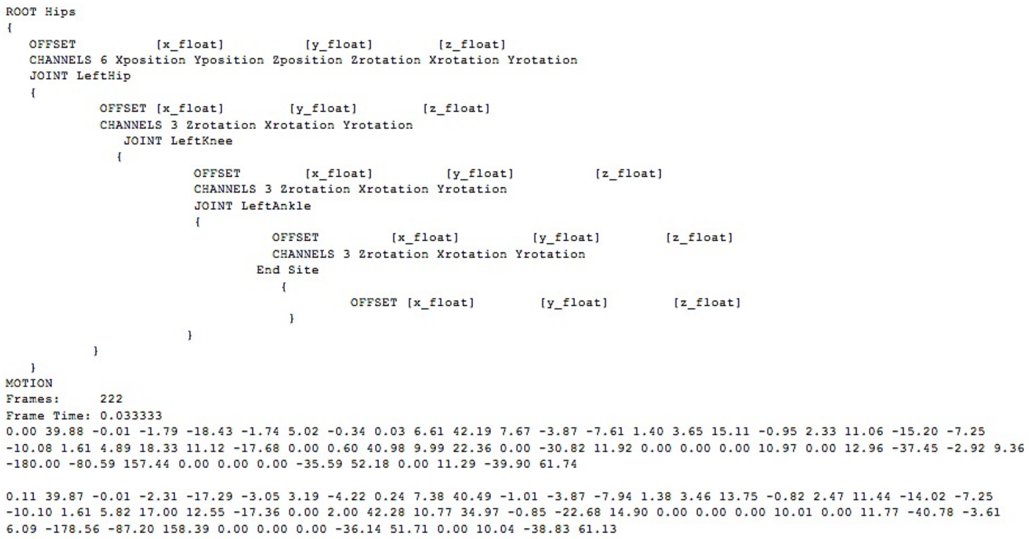
```
:fully-specified
:degrees
1
root 12.0 33.0 45.0 0.0 90.0 45.0
rightupperleg 0.0 0.0 0.0
2
root 12.0 33.0 45.0 0.0 90.0 45.0
rightupperleg 0.0 0.0 0.0
```

Figure 2.8: A sample AMC file.

2.5.2 Biovision Format

A BVH file has two parts: the header section defining the initial pose with the skeleton and the data section containing the motion information. In a header section, the skeleton formation starts with the root and chains of joints going from the root to a particular end site are represented as a single block separated by braces. Each joint has an offset value and the channel information. An offset value specifies the offset location of the current joint with respect to its parent joint. Channel information specifies the number of channels and then a list of that many labels indicating the type of each channel.

The BVH file reader (Figure2.9) must keep track of the channel (the dimensions required to specify a joint) count and the types of channels encountered as the hierarchy information is parsed. Later, when the motion information is parsed, this ordering will be needed to parse each line of motion data. The data section contains the information about the root location (global location of the whole body) and the joint angle rotations. The root position has six values, three of them represent the position of the root in the global *xyz* coordinates and the other three the global orientation as Euler angles. The data segment also contains the information about the orientation of every joint in their respective local coordinate system. The channel information provides the rotation information for calculating the global position from the Euler angles. Figure 2.9 shows a sample BVH.



```
ROOT Hips
{
   OFFSET          [x_float]          [y_float]        [z_float]
   CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
   JOINT LeftHip
   {
            OFFSET [x_float]        [y_float]        [z_float]
            CHANNELS 3 Zrotation Xrotation Yrotation
               JOINT LeftKnee
                {
                        OFFSET        [x_float]         [y_float]          [z_float]
                        CHANNELS 3 Zrotation Xrotation Yrotation
                        JOINT LeftAnkle
                        {
                                  OFFSET         [x_float]         [y_float]        [z_float]
                                  CHANNELS 3 Zrotation Xrotation Yrotation
                                End Site
                                   {
                                          OFFSET [x_float]        [y_float]        [z_float]
                                    }
                        }
           }
   }
MOTION
Frames:     222
Frame Time: 0.033333
0.00 39.88 -0.01 -1.79 -18.43 -1.74 5.02 -0.34 0.03 6.61 42.19 7.67 -3.87 -7.61 1.40 3.65 15.11 -0.95 2.33 11.06 -15.20 -7.25
-10.08 1.61 4.89 18.33 11.12 -17.68 0.00 0.60 40.98 9.99 22.36 0.00 -30.82 11.92 0.00 0.00 0.00 10.97 0.00 12.96 -37.45 -2.92 9.36
-180.00 -80.59 157.44 0.00 0.00 0.00 -35.59 52.18 0.00 11.29 -39.90 61.74

0.11 39.87 -0.01 -2.31 -17.29 -3.05 3.19 -4.22 0.24 7.38 40.49 -1.01 -3.87 -7.94 1.38 3.46 13.75 -0.82 2.47 11.44 -14.02 -7.25
-10.10 1.61 5.82 17.00 12.55 -17.36 0.00 2.00 42.28 10.77 34.97 -0.85 -22.68 14.90 0.00 0.00 0.00 10.01 0.00 11.77 -40.78 -3.61
6.09 -178.56 -87.20 158.39 0.00 0.00 0.00 -36.14 51.71 0.00 10.04 -38.83 61.13
```

Figure 2.9: A sample BVH file.

2.6 Motion Databases

All the motions required for experimentation in this thesis were obtained from the Human Motion Database constructed by the Sensory-Motor Intelligence Laboratory (SMILE Lab) at the University of Texas at Arlington. This is a well-organized motion database focused on providing data for motion editing techniques like retargeting, generalization, motion indexing, transitioning, and splicing. The database has sufficient data for studying, analyzing, and testing motion splicing techniques. The database contains different motion sets with two motion files where the subject performs two different actions individually and a third motion file where these two actions are performed simultaneously. For example, there is a motion set containing a motion file where the subject performs the waving action and a motion file where the subject is walking. Then there is the third motion file where the subject is waving while walking [AG10].

2.7 Motion Editing Techniques

There has been a great deal of research on motion editing techniques. In order to accommodate a given motion to a particular environment, modifications need to be made on the current motion to synthesize a new motion. One way to produce a physically plausible motion is to generate them directly from physical simulation. Badler *et al.* [1993] treated the whole body as a mechanical linkage system and applied inverse dynamics in animating human locomotion. Boulic *et al.* [BT90] attempted kinetic generalization of empirical walking data to generate locomotion. Bruderlin *et al.* [BC89] used a combination of kinematic considerations and dynamic motion control for locomotion. Liu and Popovic [LP02] demonstrated that motion could be synthesized entirely from conservation laws.

In contrast, motion capture produces a rich variety of highly realistic motions. Motion capture by itself does not have much control over the motion. Therefore, much effort has been invested in developing editing methods. Some of the motion editing techniques incorporated physical principles; Tak *et al.* [TS02] used an iterative algorithm based on Kalman filters to enforce kinetic constraints. Few other researchers tried to implement motion editing based on physics [WP95, YN00]. Shin *et al.* [LS02] came up with an idea of physical touch up on edited motions to improve physical plausibility.

Since the motion is represented as discrete points in time, signal processing has a great influence in motion synthesis. Kim *et al.* [KP03], Amaya *et al.* [AB96], and Jones and Boltz [JB89] demonstrated the significance of signal processing in analyzing motions.

A single motion file may not contain all the required actions for generating a desired animation. Subsets of motion frames from different motion files have to be concatenated in an orderly sequence to generate a complete animation. Kovar *et al.* [KG02] presented a novel method of creating realistic controllable motion from a large database by encapsulating connections among the database. Arikan *et al.* [AF02] presented a framework to generate new motions by cutting and pasting example motions. In both these methods, researchers find an optimal point where a transition can be made between two motion clips. Another set of algorithms deal with ways to linearly blend two frames [MO97]. Safonova *et al.* [SH05] showed how two frames could be linearly interpolated. Lee *et al.* [LC06] used reinforcement learning and dynamic programming to determine how a character can best perform a desired action for any given character state. Srinivasan *et al.* [SH04] pre-computed mobility maps from a motion transition graph to aid in runtime generation of locomotion. Pullen *et al.* [PB02] developed methods for combining partial motions with complete motions.

CHAPTER 3

MOTION SPLICING TECHNIQUES

In this chapter, we describe the three splicing techniques considered for comparison using our quantitative evaluation methodology: spatial body alignment [HK06], segmentation-based algorithm [JL08], and naïve DOF replacement [Per95].

Spatial body alignment splices whole-body motion data between two different actions performed while in locomotion. A temporal alignment is performed between the two motions and a path with least resistance for frame replacement is determined. A spatial alignment between the matched upper frames of both the motions is performed and a spliced motion is generated.

The segmentation-based algorithm consists of segmenting the entire motion based on the overall body kinetic energy. Other similar partial motions replace partial motions associated with segments and a new spliced motion is synthesized.

The naïve DOF replacement of joints from one motion to another transfers the values associated with one joint from one motion to the other motion. The exact same motion associated with the joint in one motion is replicated on another motion.

As a part of our comparative study we implemented these three motion splicing techniques. Thus we provide some insight in to each of these methods.

3.1 Splicing Methods Background

Splicing techniques are widely used in the gaming industry to instantly generate actions that simultaneously combine two actions. Heck *et al.* [HK06] presented a splicing method for actions performed with locomotion. In their approach, they used Dynamic Time Warping [KG02] to best match poses and spatial alignment for keeping the body correlation. Ashraf *et al.* [AW00] addressed upper and lower body correlation and a smooth way of blending them separately.

Perlin [*Per*95] accomplished splicing by naïve DOF replacement in the joint angles between two frames. Ikemoto *et al.* [IF04] used naïve degree of freedom replacement to transplant pieces of motion.

Jang *et al.* [JL08] used partial motion clips to find the similarity in the motions and similar segments to perform a naïve DOF replacement. Pullen *et al.* [PB02] were able to synthesize a new motion by interpolating animator specified key frames.

The three methods we use for testing are spatial body alignment [HK06], segmentation-based algorithm [JL08], and naïve DOF replacement [Per95]. We selected these techniques because they are among the most significant and novel methods of motion splicing. The spatial body alignment method considers the correlation between joints. The segmentation–based algorithm is based on the kinetic parameters and accounts for balance in the spliced motion. The naïve DOF replacement is a method capable of splicing any example motions (running, punching) to create complex motions (running *and* punching) with little effort.

3.2 Spatial Body Alignment Technique

Heck *et al.* [HK06] presented a simple and efficient technique for splicing together the upper body action of one motion and the lower body locomotion of another motion in a manner that preserves the fidelity of the original source motions. The approach was fundamentally based on the idea that interactive applications often divide character control to motion and locomotion while preserving the correlation between them. They proposed an efficient method to synthesize spliced motions that appropriately preserve the correlation present in the original motions. The algorithm allows the upper body motion to be decoupled from the lower body locomotion by preserving the details such that the upper body motion can be precisely attached to another lower body locomotion. For example, consider a motion where the person is walking along a straight line with a suitcase and a second motion where a person walks along a curved path. This algorithm is best suited to generating a motion, where the person walks with a suitcase in a curved path. The rest of this section explains the different steps in implementing this algorithm.

Correlation between different parts of the body is one of the most important constraints while generating visually realistic motions. In every day of our life, we see hundreds of different actions being performed in different styles. Our brains get well trained in differentiating between real and artificial motions. Hence, any motion that is uncorrelated can be easily spotted. There have been a large number of studies about the correlation of body parts. Figure 3.1 shows how different parts of body are correlated while performing a wave action. The graph in the figure shows the variation of the second derivative of position of joints at different instances in time. This means that there is a similarity in the pattern of movement of joints while performing an action. The number and degree of correlation varies greatly depending on the exact motion, but the importance of these correlations is the driving force for this algorithm.

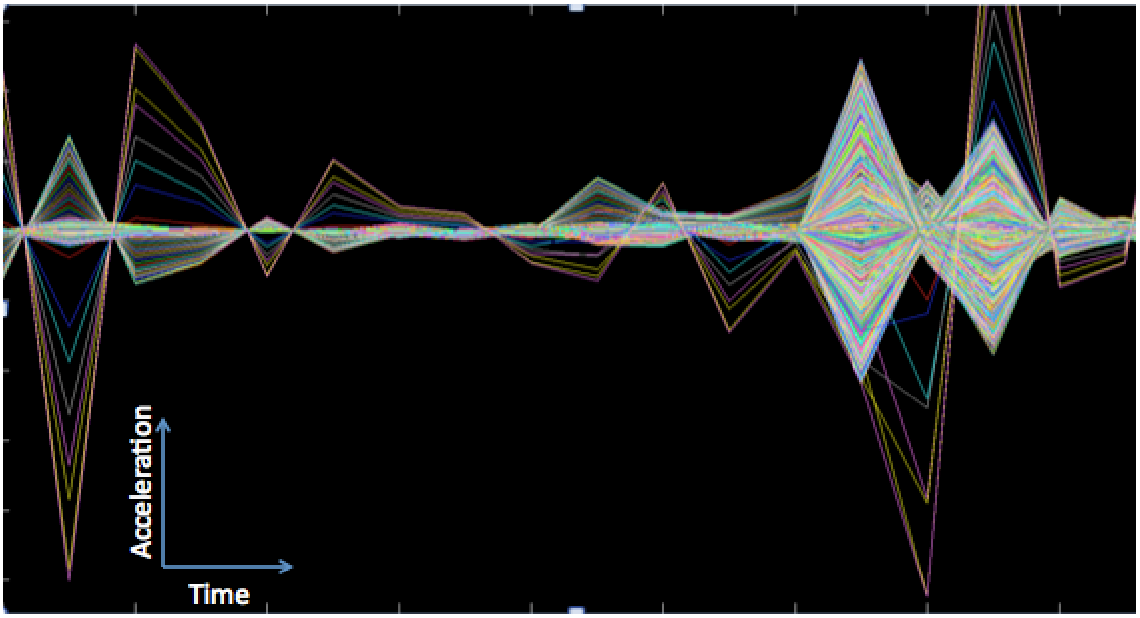


Figure 3.1: Joint correlation graph for the waving action. Each wave in the graph represents a single DOF associated with a joint in the body. The pattern of zero crossing in the graph shows why the correlation in joint movement is important while splicing motion.

The splicing algorithm works as follows. Suppose we have motion *A* and motion *B* with possibly the same locomotion performed by the lower body and different actions performed by the upper body. The algorithm decouples the upper body motion from the lower body motion of both motion *A* and motion *B* to create four individual motion sets: A_{Lower}, A_{Upper}, B_{Lower}, B_{Upper}. Then the algorithm uses dynamic time warping **[**KG02**]** to find the best match between the lower body motions A_{Lower} and B_{Lower}. Once the time alignment has been found using the lower body point cloud (*x*, *y*, and *z* global location of the lower body joints in the Cartesian coordinate system), the upper body motions A_{Upper}.and B_{Upper} are matched according to the time alignment obtained from the lower body motions. The spatial alignment translates and rotates the upper body of motion *B* to match that of the motion *A*.

Using the pelvis as the pivotal point between the upper body and lower body, the algorithm finds the best orientation for pelvis of the spliced motion using Horn's method **[**Hor87**]**. Finally, in order to maintain the posture of the character in the spliced motion similar to the original action (B_{Upper}), the pelvis is tweaked again to best match the shoulder alignment of the spliced motion to the shoulder alignment of motion *B*. The best orientation is computed using shoulders, spine, pelvis, and the hips. Each of the above steps is presented in detail in the following subsections.

3.2.1 Temporal alignment

Motion capture data is organized such that the global position and orientation of the skeleton's root and the joint angle rotations of all other joints are provided at every frame. Hence, motions are represented as sequences of discrete points in time. Let motion $\mathrm{M} = \{m_1, m_2, \dots \dots m_N\}$ where m_i is the motion at frame *i* and *N* is the total number of frames. Let m_i = [$P(i)$, $Q_1(i)$, $Q_2(i)$, …, $Q_k(i)$], where *P(i)* is the global root position at frame *i,* $Q_1(i)$ is the global orientation of the root at frame *i*, $Q_j(i)$ is the orientation of j^{th} joint at frame *i,* and *k* is the total number of joints including the root. The global root position is represented in the 3D Cartesian coordinates and orientations are represented as Euler angles. Dynamic time warping consists of two steps; in the first step, a distance matrix is calculated between the frames in two motions and in the second step a least resistance path is computed.

3.2.1.1 Distance Calculation for motion

There are different ways to calculate the distance between two frames in different motions. 3D point cloud-based distance calculation [KG02], quaternion-based distance calculation [HG10], and dynamics-based distance calculation [KP03], where rotational velocity of the joints is the measure of distance are some of them. The distance calculation used in this thesis is the one by Kovar *et al.* [KG02], where the distance is calculated as the difference in the 3D location of the joints in the two frames.

The computation of the distance between two frames involves three steps. In the first step, a window of frames is selected around each frame in both motions. Secondly, we convert the frame to a point cloud. Finally, the optimal sum of the squared distances of the corresponding points is computed. Instead of using the entire point cloud, Heck *et al.* **[**HK06**]** only used three points from the lower bodies: root, left knee, and right knee. This largely reduces the computation necessary for the time alignment. The distance matrix *Dist* is found using Equation 3.1:

$$Dist = \min_{\theta, x_0, z_0} \sum_i W_i \left\| X(i) - Tr_{\theta, x_0, z_0} X'(i) \right\|^2, \qquad 3.1$$

where W_i is the weight that may be chosen to taper off towards the end of the window, *X(i)* is the point cloud values of the every single joint in motion *A*, *X'(i)* is the point cloud values for every single frame in motion *B*, and Tr_{θ, x_0, z_0} is the linear transformation matrix computed in the 3D space to align *X* and *X'*.

The linear transformation rotates a point *p* about the *y*-axis (vertical axis) by *θ* degrees and then translates it by (x_0, z_0). The index i is over the number of points in each point cloud. Since the captured motion is in the *y*-up format, the point cloud computed has to translate along the plane formed by the *x*-axis and *z*-axis and rotated along the *y*-axis. This transformation is obtained by the following closed form equations:

$$= arctan \frac{\sum_j W_j (x_j z'_j - x'_j z_j) - \frac{1}{\sum_j W_j} (\bar{x}\overline{z'} - \overline{x'}\bar{z})}{\sum_j W_j (x_j x'_j + z_j z'_j) - \frac{1}{\sum_j W_j} (\bar{x}\overline{x'} + \overline{z'}\bar{z})}, \qquad 3.2$$

$$x_0 = \frac{1}{\sum_j W_j}\left(\bar{x} - \overline{x'}\cos(\theta) - \overline{z'}\sin(\theta)\right), \qquad 3.3$$

$$z_0 = \frac{1}{\sum_j W_j}\left(\bar{z} + \overline{x'}\sin(\theta) - \overline{z'}\cos(\theta)\right), \qquad 3.4$$

where $\bar{x} = \sum w_j x_j$, W_j is the weight assigned to the j^{th} joint.

3.2.1.2 Creating the Time Warp Curve

Using the distance function in the above equation, we calculate a cost matrix *C* using the following equations:

$$\text{(first row) } \mathrm{C}(1, \mathrm{c}) = \sum_{j=1}^{c} D(x_1, y_j), c \in [1, M], \qquad 3.5$$

$$\text{(first column) } \mathrm{C}(\mathrm{r}, 1) = \sum_{j=1}^{r} D(x_j, y_1), r \in [1, N], \qquad 3.6$$

and

$$C\,(r, c) = min\,\{C\,(r-1, c-1), C\,(r-1, c), C\,(r, c-1)\} + Dist(r, c), r \in [1, N], c \in [1, M]), \qquad 3.7$$

where *C (r, c)* is the cost of the r^{th} row and c^{th} column. M is the total number of frames in the first motion. N is the total number of motion in the second motion.

Using the cost matrix *C*, we create a grid where the elements correspond to warping costs between frames from motion *A* to frames in motion *B*. We select a least resistance path following three properties: continuity, causality, and slope limit. Continuity is that each cell on the path must be adjacent to another cell on the path. Causality means that the path is monotonically going in the forward direction. The slope limit property allows a maximum of consecutive horizontal or vertical steps in the path. All the initial frames before the first match frame are removed in the spliced motion. Figure 3.2 shows a typical dynamic time warp matching, where the rows represent the frames of motion *A* and columns represent the frames of motion *B*.

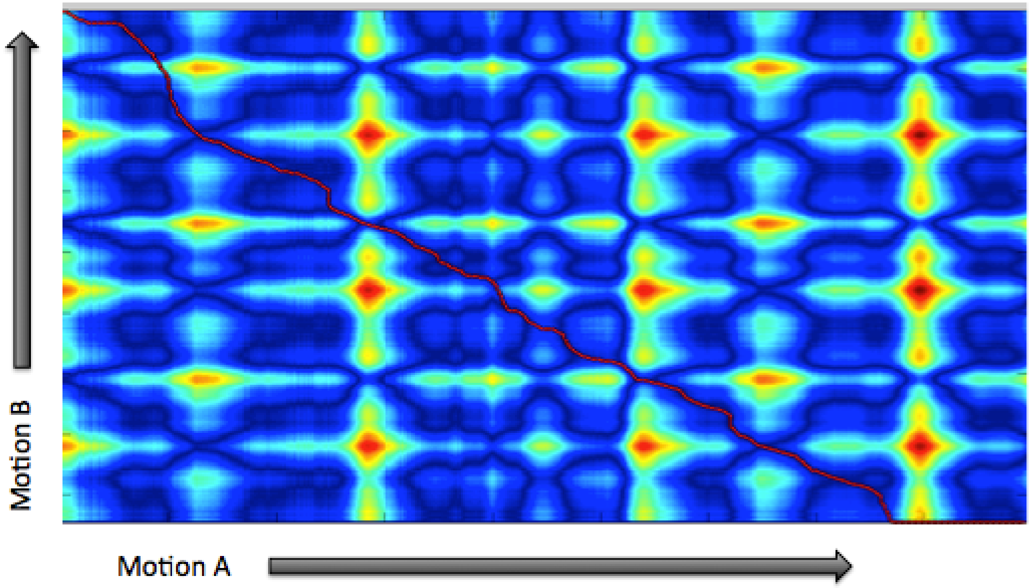


Figure 3.2: Dynamic time warping performed on two motions. The red line represents the matching between closely aligned frames in the two motions.

Dynamic time warping is based on the well-known dynamic programming method. The frame correspondences generated by dynamic time warping could be used directly to form a piecewise linear time warp curve.

3.2.2 Spatial Alignment

The temporal alignment dictates the corresponding frame numbers to be matched in motion *A* and motion *B*. Now B_{Upper} has to replace A_{Upper} and form a new motion (A_{lower} B_{Upper}). To generate a realistic motion, a local orientation for the pelvis has to be calculated in such a way that the upper body would turn, bend, and twist in relation to the lower body, while compensating for the movement of the root, effectively stabilizing the upper body. The easiest way of performing this calculation is to assign the pelvis with the local orientation of the upper body (B_{Upper}) or to retain the pelvis orientation of the lower body (A_{lower}). Implementing both these methods either destroys the coordination in the movement or disorients the upper and lower body. Heck *et al.* **[**HK06**]** used Horn's method **[***Hor*87**]** to find the best orientation on the pelvis using A_{upper} as reference. More specifically, for every pair of corresponding frames, a local orientation was computed using the point clouds derived from the spine and the two shoulder joints.

Horn [*Hor87*] discovered a method to find the rotation that best aligns two point clouds X_l and X_r in 3D space. The method finds a unit quaternion q that maximizes the equation q^TSq, where S is the 4 x 4 matrix whose elements are the sums of products of coordinates measured in both point clouds, X_l and X_r. The matrix contains all the information required to solve the least square problem for the best rotation:

$$S = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix}, \tag*{3.8}$$

where $S_{xx} = \sum_i^n x_{l,i}\, x_{r,i}$, $S_{xy} = \sum_i^n x_{l,i}\, x_{r,i}$, $S_{yx} = \sum_i^n x_{l,i}\, x_{r,i}$, $x_{l,i}$, $y_{l,i}$, $z_{l,i}$ are the Cartesian coordinates of point x_i in the left point cloud X_l, $x_{r,i}$, $y_{r,i}$, $z_{r,i}$ are the Cartesian coordinates of point x_i in the right point cloud X_r, and n is the number of points in both point clouds.

The sum of the diagonal of matrix S is zero. The unit quaternion that maximizes the equation q^TSq is the eigenvector corresponding to the largest positive eigenvalue of the matrix S. An alternate way of computing the unit quaternion is using Ferrari's method **[***Hor*87**]** and calculating the coefficients of the quartic equation obtained from matrix S. After computing the unit quaternion $q = q_0 + iq_1 + jq_2 + kq_3$, the rotation matrix R can be calculated using the following equation:

$$R = \begin{bmatrix} (q_0^2 + q_x^2 - q_y^2 - q_z^2) & 2(q_xq_y - q_0q_z) & 2(q_xq_z + q_0q_y) \\ 2(q_yq_x + q_0q_z) & (q_0^2 - q_x^2 + q_y^2 - q_z^2) & 2(q_yq_z - q_0q_z) \\ 2(q_zq_x - q_0q_y) & 2(q_zq_y + q_0q_z) & (q_0^2 - q_x^2 - q_y^2 + q_z^2) \end{bmatrix}. \tag*{3.9}$$

The method suggests that the best translation is the difference between the centroid of one set of measurements and the scaled and rotated centroid of the other measurement. Hence, while computing the spatial alignment both the point clouds obtained from motion *A* and motion *B* are initially translated to origin and then the best orientation is found by Horn's method.

3.2.3 Posture Transfer

Temporal alignment and spatial alignment generate the complete spliced motion. However, there are some actions in which the inclination of the spine with respect to the lower body posture is important. For example, if a person is carrying a box, depending on the weight of the box, the person would be leaning in the opposite direction to compensate for the box's weight. Spatial alignment aligns B_{upper} to A_{upper} such that the position of the shoulders in the spliced motion is at the position that best matches the position of the shoulders in the A_{upper} thus losing the originality of the motion B. Now the box looks lighter as the spine inclines lesser than the original action.

In order to preserve the posture in the original action, a posture transfer step is added to the algorithm to automatically correct this issue. Posture transfer is similar to spatial alignment. The only difference is the selection of a different point cloud. Here, the best aligning rotation is found between the spliced motion and B_{upper}. The point cloud is formed using the joint locations of the two shoulders, pelvis, and the hips. Figure 3.3 shows the spliced motion before and after the posture transfer.

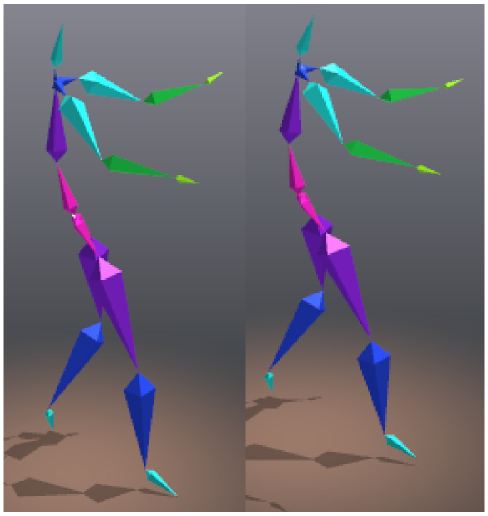


Figure 3.3: Spliced motion before and after posture transfer.

3.2.4 Limitations of the Spatial Body Alignment Technique

The most important limitation of this method is the assumption that both original actions perform locomotion in the lower bodies. For example, a motion in which the shoulder motion is the only meaningful motion of the action cannot be spliced by this method. Another potential drawback of this motion splicing method is that it does not explicitly take into account the physics of the original motions. The inability to account for the physics can cause the physical characteristic of objects in the scene to look different. The spliced motion in which the person carries a box is a good demonstration of the above limitation. When a person carrying a box is spliced with another motion, the subject's spine does not incline in the exact same way as in the original action. Hence, the box appears to be lighter than it is. This is what changing the physical characteristics of the objects in the animation means.

3.3 Segmentation-based Technique

This technique synthesizes new motion from existing motion data. The captured motion data involving the complete skeleton is partitioned into partial motions involving lesser number of DOFs and later spliced to generate a complete motion. Segmentation, principal component analysis, and clustering techniques are used to find the best matches between partial motions while combining the different partial motions. Realism and physical plausibility of the combined motions are verified using the Support Vector Machines **[**IF04**]**. The key advantage of this technique is the reduction in database requirements and instantaneous generation of motion. This splicing technique allows any group of joints in one motion to be replaced with a similar set of joints from the other motion. Since the segmentation and clustering is based on the kinetic parameters, the spliced motion would closely follow the kinetic nature of the base motion to maintain the balance of the body

Every joint has a set of related and unrelated joints by which it can be defined. Each action has a set of essential joints involved in the performance of the action. If the subject is performing a punch action, depending on the speed and force associated with the punch the number of joints involved in this action varies. Taking this in to account, if there is a motion in which a particular set of joints move similarly, the set of unrelated joints can be replaced with data from another motion file to generate a new motion involving the original motion. The similarity measure is used to generate a rich variety of natural looking and physically plausible motions without much effort.

Assume every motion is a simultaneous collection of distinguishable coordinated motions. This implies that the natural look of the motion comes from the coordinated body movement that must also be replicated in synthesized motions. The combination of partial motions can be defined as: Let motion A and motion B be natural motions where A_{partial} and B_{partial} are similar to each other. We can produce the analogous combinations (A_{partial}, B_{rest}) and (B_{partial}, A_{rest}), where A_{rest} and B_{rest} are the partial motions complementary to A_{partial} and B_{partial}, respectively. Figure 3.4 illustrates the set of related and unrelated joints involved while performing a wave action. The stick figure in blue is a single frame of the wave action. The segmentation-based method splits the action into two sets of partial motions, named as *'partial'* and *'rest'* as illustrated in the Figure 3.4, and later replaces the partial motion from one motion with that of the other motion to generate the spliced motion.

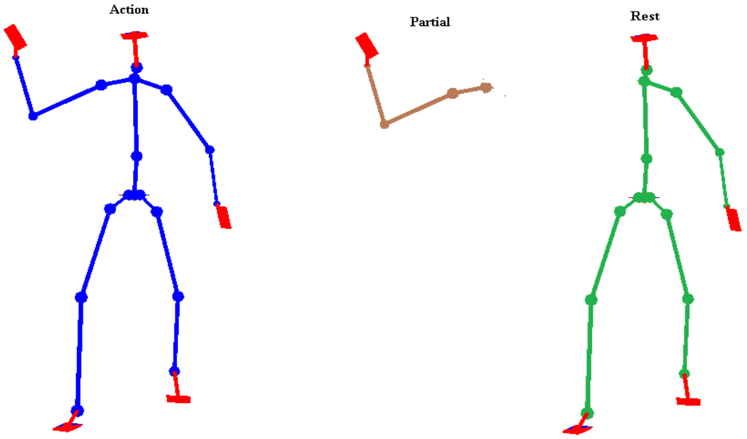


Figure 3.4: Partitioning of a motion into partial motions.

The input motion data is divided into motion segments along the time axis. Each segment is further subdivided into partial motions corresponding to joint groups having reduced data set. Finally, these partial motions are classified into groups with similar characteristics using a hierarchical clustering technique.

3.3.1 Motion Segmentation

Motion beats are the distinctive rapid directional changes in any basic movement corresponding to the regular rhythmic unit of time for motion. Rhythmic motion synthesis refers to the capture of motion beats from basic movements and their transitions.

Rapid directional changes can be detected by analyzing the second derivative of each motion signal at every frame. The dominant period of a signal is given by $y(t_i) = (t_i)_{i=1}^{N}$, where each t_i is the moment of zero crossing for frame *i* : $t_i = \phi + \tilde{k}_i T + n_i, i = 1, \dots\dots, N$, where *T* is the unknown period of beats, ϕ is a random phase with a uniform distribution over an interval [0, *T*), $\tilde{k}_i$ for 1 ≤ *j* ≤ *N* are positive non-repeating integers, and n_i for 1 ≤ *i* ≤ *N* are zero-mean additive white Gaussian noises in the interval [−*T*/2, *T*/2).

Not every moment of zero crossing is a motion beat. Dominant frequency of a sample signal can be analyzed by constructing *N*-1 cosine curves each spanning a time interval of ($t_{\tilde{k}-1}$, $t_{\tilde{k}}$), where $t_{\tilde{k}}$ is the upper extreme value in the sinusoidal input signal. Assuming the beat period to be not less than the inter frame time and the highest possible frequency being the reciprocal of the inter frame time, performing a power spectral density analysis on the sample input signal at twice the highest possible frequency (Nyquist sampling theorem) would provide the dominant frequency of the candidate beat:

$$Pow_y(f) = \frac{1}{\bar{M}} \left|\sum_{j=1}^{\bar{M}} y(\hat{t}_j) e^{2i\pi f \hat{t}_j}\right|^2, \qquad 3.10$$

where $Pow_y(f)$ is the power at different values of frequency *f*, $\hat{t}_j$ is the sampling rate based on the Nyquist sampling criteria, $\bar{M}$ is the number of terms in $\hat{t}_j$, and *y(t)* is the cosine functions of every joint in a given time period. The most dominant frequency can be given by $\tilde{f} =$

arg $max_{(f)}\ Pow_y(f)$. The dominant period can be computed from the dominant frequency as $\tilde{T} = \frac{1}{f}$.

The dominant period for multiple joints is computed by the power spectral density analysis of the superimposed single joint signals. The superimposed signal is $\tilde{S} = \sum_{j=1}^{K} \tilde{S}(\mathrm{t}_i)$ = $y_1(t_j)$ + ... + $y_K(t_j)$, where K is the number of joints. Finally, the reference beat r_{ref} is given by $\mathrm{r}_{\mathrm{ref}}$ = ˜+ $k\tilde{T}$, for k = 1, ..., N, where˜is the phase of the multiple joints and $\tilde{T}$ is the dominant period of the multiple joint signal. The power spectral density for multiple joints is given by the following equation:

$$Pow_y(f) = \frac{1}{\tilde{M}} \left| \sum_{j=1}^{\bar{M}} S(\hat{t}_j) e^{2i\pi f \hat{t}_j} \right|^2. \qquad 3.11$$

The actual beats are not perfectly coincident with the reference beats other than in an ideal case. While most candidate beats except some outliers are likely to be clustered around the reference motion beats. The representative of each cluster is chosen as the actual beat using a Huber M-estimator [*Hub*81]. For each reference beat $\mathrm{r}_{\mathrm{ref}}$, we estimate the representative $\hat{b}_{ref}$ from the candidates within the window [$\mathrm{r}_{\mathrm{ref}} - \tilde{T}/2$, $\mathrm{r}_{\mathrm{ref}} + \tilde{T}/2$] as follows:

$$\hat{b}_{ref} = \arg\max \sum_{j=0}^{\bar{M}} W_{ref}(t_j) \rho(t_{ref} - b), \qquad 3.12$$

where $\rho(x)$ is a Huber function for a tuning constant α:

$$\rho(x) = \begin{cases} \frac{x^2}{2} & if\ |x| \le \alpha \\ k|x| - \frac{k}{2} & if\ |x| > \alpha \end{cases}, \qquad 3.13$$

and W(x) is a window function:

$$W_{ref}(x) = \begin{cases} 1 & if\ x\ \left[r_{ref} - \frac{\hat{T}}{2},\ r_{ref} + \frac{\hat{T}}{2}\right]. \\ 0 & otherwise \end{cases} \qquad 3.14$$

The value $\hat{b}_{ref}$ is computed numerically with the downhill simplex minimization algorithm [PT99]. The reference beat r_{ref} is used as an initial guess for the solver. If no directional changing beat was identified in the window [$r_{ref} - \tilde{T}/2$, $r_{ref} + \tilde{T}/2$] for some '*ref*', we designate the reference beat r_{ref} itself as a motion beat $\hat{b}_{ref}$.

Jang *et al.* [JL08**]** segments the input motion at local extreme points of angular kinetic energy given by the following equation:

$$- \quad , \qquad 3.15$$

where p is the total number of joints, I_j is the moment of inertia of the j^{th} joint, and $_k$ is the rotational velocity. The i^{th} motion segment in the set of n segments can be represented as S_i *(i = 1,……,n)*. The matching of segments uses the derivative of the angular kinetic energy. Figure 3.5 shows a motion segmented at local extremes of the angular kinetic energy.

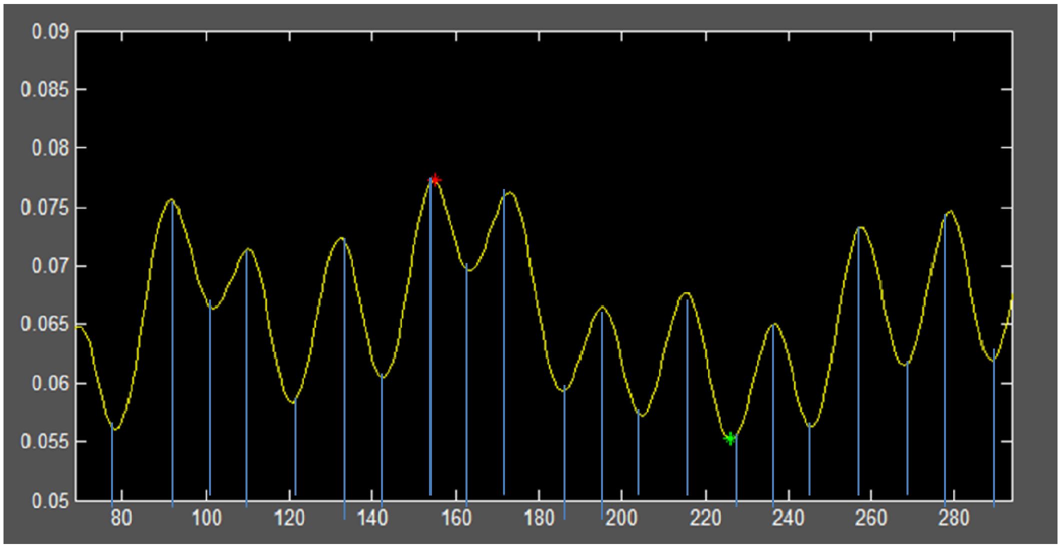


Figure 3.5: Angular kinetic energy segmented at local extreme points.

3.3.2 Partial Motions

We divide the motion segments into partial motions by selecting different joint groups (possibly a single joint). The most common way of partitioning the body is as three groups: legs, arms, and the rest of the body. By using this approach, the body can be partitioned into *m* individual joints or groups. As the number *m* increases more motions can be synthesized from the segments, which help to enrich the database. As the number of joint group increases, the combinations with which the spliced motion can be generated increases. Hence, the above method can enrich the database with a variety of new motions.

In order to make the number of frames in each segment the same, every segment is reduced to 10 equally spaced frames. The joint angles are interpolated using Spherical linear interpolation (SLERP) and the root location (the character's global position) is interpolated using cubic spline curves. The skeleton of the Human Motion Database has 66 DOFs where the lower body consists of 30 DOFs and the upper body consists of 36 DOFs. After sampling each segment into 10 samples, each segment has now 660 DOFs which are later reduced using principle component analysis.

Spherical linear interpolation is a method to calculate the intermediate angle between a pair of angles. In order to precisely define the orientation of a joint in space, Euler Angles can be used. Even though they may be used to compute rotations, they are originally designed to provide a clear and graphical description of the spatial orientation of a rigid object in space. Euler Angles are a set of angles between a rotated system - the one to be described and a static system - the space the object is situated in. In order to calculate intermediate angles, Euler angles are converted into quaternions. Quaternions provide a straightforward way to represent, illustrate, and compute rotations. This is due to the fact that quaternions can be used as an easier representation than rotation matrices and a rotation can be smoothly computed by multiplying a quaternion with another.

When a joint moves from one place to another, the path this joint makes and the points situated along the path are found by interpolating between two points using linear interpolation (LERP). LERP draws a straight line between the two points and computes the intermediate points. If P_0 and P_1 are the two points in space and $t_{inter} \in [0,1]$ is the interpolation parameter, LERP can be computed using the following formula:

$$LERP\ (P_0, P_1;\ t_{inter})\ =\ (1 - t_{inter})\ P_0\ +\ t_{inter}\ P_1. \qquad 3.15$$

A straight line is however not useful to animations since a rotating joint is supposed to move along a smooth curve. For this, one can use spherical linear interpolation (SLERP).

SLERP is performed on the surface of a unit sphere. SLERP of two unit quaternion's q_1 and q_2 can be formulated as

$$SLERP\ (q_1, q_2;\ t_{inter})\ =\ q_1\ ({q_1}^{-1} q_2)^{t_{inter}}. \qquad 3.16$$

We convert from joint angles represented as Euler angles to quaternions and then convert back to Euler angles.

At each frame in a motion, the root location specifies the global position of the skeleton. In order to calculate the intermediate points, the root location has to be interpolated using the smooth best-fit curve that travels through all the discrete points. A cubic spline assigns weight to each data point. The weights are the coefficients of the cubic polynomials used to interpolate the data. These coefficients bend the line so that it passes through each of the data points without any erratic behavior or breaks in continuity. The cubic spline has a better endpoint behavior where it relates the first frame M_1 to $2M_2$ and M_3 and the last frame M_n to $2M_{n-1}$ and M_{n-2}. One other strength of the cubic spline curve is its ability to correlate data that doesn't follow any specific pattern without a single polynomial's extreme behavior.

Using the above two methods, SLERP and cubic spline, we will be able to make the motion continuous. Later we pick 10 equidistant frames to represent the entire motion. For further reduction in the computation time, the motion at a given frame is reduced to a lower dimensional space using Principal Component Analysis. Principle component analysis is a statistical technique used to identify patterns in data by expressing the data in such a way that highlights its similarities and differences. Once the pattern in the data has been identified, then it can be easily compressed. In this splicing algorithm, PCA reduces the number of dimensions in the motion data to 10, thereby providing a high computational advantage while keeping the detailed variation in the motion data.

In order to reduce the dimensionality to lower dimensions, initially the given data set centroid is translated to the origin by subtracting all the values from the mean. Then a covariance matrix is computed from the original high dimensional data. Eigenvalues and eigenvectors are computed from the covariance matrix. After this step, the appropriate eigenvalues and eigenvectors are selected and the lower dimensionality data can be computed. Figure 3.6 shows the importance of PCA in reducing the dimension of the motion segments.

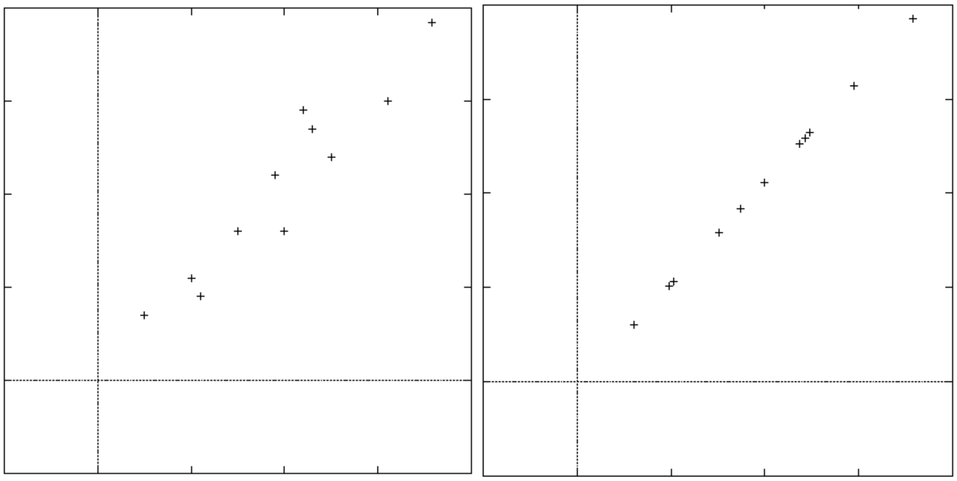


Figure 3.6: The dimensionality reduction in a sample data set by PCA.

Safonova and Hodgins [SH04] introduced a method to reduce the dimensionality of human motion data using PCA. Solving this problem for complex characters such as humans is hard due to the high dimensionality of the search space. The dimensionality is an artifact of the problem representation because most dynamic human behaviors are intrinsically low dimensional.

The next step in the segmentation-based splicing algorithm is to group similar motion segments into clusters. The partial motions generated are clustered into groups using the hierarchical clustering method [GJ96]. The clustering technique uses the complete link method to measure the distance between clusters. The complete link method uses the least similar pair of motion segments in the two clusters to measure the distance between the clusters. The distance between two motion segments is computed by calculating the norm between the vectors formed by principle component coefficients of the two segments. Motion segments that are logically similar are then stored together in the same cluster.

3.3.3 Motion synthesis using Partial Motions

In order to synthesize the spliced motion, a partial motion segment is selected as the base for comparison. If there is a second partial motion segment similar to the base motion, then the base motion can be replaced with other motion generating a new motion. The most important constraint is the length of the partial motions, both the base partial motion and the replacement partial motion should have pretty much equal number of frames. If the partial motions are too different in size, then the synthesized motion looks unnatural. Once the partial motions are selected, conventional motion transition techniques are used to generate smooth transitions between adjacent motion sequences. Foot skate cleanup must be performed if the foot skate problem occurs. An alternate way to avoid foot skate problems is to make the root a joint of the base partial motion.

3.3.4 Limitations of the Segmentation-based Technique

A major limitation of the segmentation-based technique is the large computation time required for the generation of spliced motion. Motions must be segmented and partial motions generated beforehand. Furthermore, as the number of joint groups increases, the computation becomes more time consuming. Segmenting the body at the local extreme points of angular kinetic energy of the motion is not an appropriate segmentation technique, as it does not take into consideration the significant group of joints performing the action. For example, if the subject performs a punch action, then the segmentation is performed at every local extreme point of total kinetic energy. This means that the punch action is segmented wherever the velocity of the hand is the maximum or minimum. Hence, a single punch action is segmented at five different points in a time frame creating four segments. In order to exactly replicate the same action in the spliced motion, there should be four consecutive segments in the base motion that accommodates the above action segments.

3.4 Naïve Degree of Freedom Replacement

Naïve DOF replacement is one of the simplest and easiest methods in synthesizing spliced motions. It is one of the best techniques used in real time motion splicing due to its less computation requirements. The method can create a rich database of motions clips from existing motion data.

Naïve DOF replacement is the technique for generating motions by replacement of any particular joint or groups of joints from one motion with that from another motion. Naïve DOF replacement has been widely used in the animation industry. Many motion-splicing algorithms [PB02, Per95, IF04] currently available are entirely developed by modifying this method. All these algorithms incorporate additional blocks that would either find matching clusters or automatically check for physical plausibility of the synthesized motion.

Pullen and Bregler [PB02] were the first to use this method. They used frequency bands to decompose the sampling data and the information in these bands to create the synthetic data, one frequency band at a time. Perlin *et al.* [*Per*95] invented a powerful interactive method in synthesizing motion using signal generation, modification and conventions for fitting things together. Ikemoto *et al.* [IF04] developed a rule-based approach to naively replace DOFs in a motion to generate new motions. This technique used support vector machines to identify which motions looked real and which were not based on a set of training data.

A naïve DOF replacement technique should be applied to joints or joint groups having a low joint correlation. There are actions involving the whole body, like leaning forward or bending down, where all the joints in the body rearrange themselves to keep the body balanced. If such a motion is being spliced without giving much care to the related joints, the spliced motion is going to be both physically incorrect and visually unrealistic.

A motion M is represented as $M = \{m_1, m_2 \dots m_n\}$, where n is the number of frames in the motion, m_i is the pose at frame i, which is given by $m_i = [P(i), Q_1(i), Q_2(i), \dots, Q_k(i)]$, where $P(i)$ is the global location of the root at frame i, K is the total number of joints in the body, and is the Euler angle of the joint $j \in \{1, 2, \ldots, K\}$. Using naïve DOF replacement, the partial motion M_A concerning a subset of joints can be replaced by the partial motion M_B concerning the same subset of joints to generate a new motion. For example, if M_A is a walking motion and M_B is a waving motion, the joint angles of the right arm in the motion M_A can be replaced by the joint angles of the right arm in motion M_B to create a motion M_C that resembles a motion, where the character walks and waves. The spliced motion M_C would be {   }. If the appropriate joints are not selected when splicing a turn with a wave, the lower body can keep spinning, while the upper body stays in a particular direction and keeps waving. Hence, the animator should clearly define the joints to be replaced. Figure 3.7 shows examples of spliced motions.

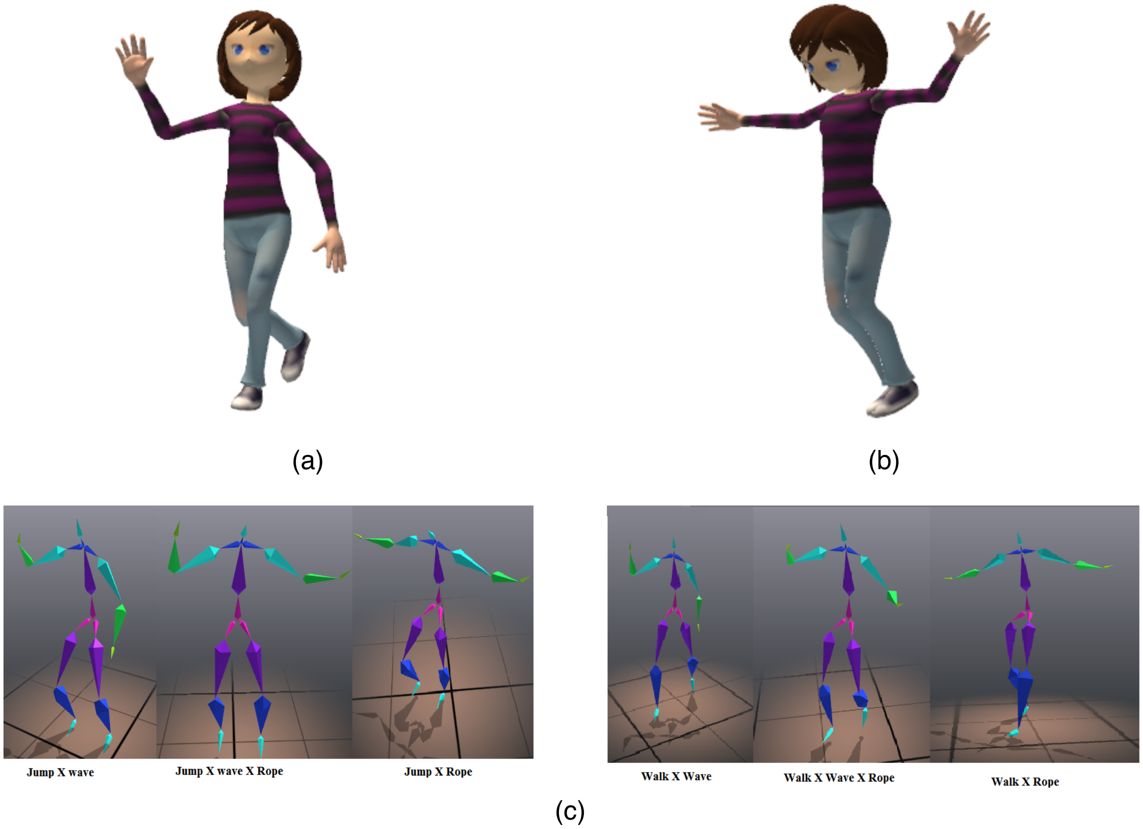


Figure 3.7: Motions generated using Naïve DOF replacement (a) Motion A “Walk and Wave” (b) Motion B “Jump Rope” (c) Spliced motions from motion A and motion B.

The following are some general guidelines to generate good slicing motions. To achieve good motion splicing, the initial poses of the base motion and the replacement motion should be similar enough to perform a transplant. Generally replacing only the hands and the head can synthesize realistic motion. In order to avoid the foot skate problem, the root and the lower body should be derived from a single motion. Motion clips with annotated actions can be helpful while choosing the appropriate frame. Replacement joints can be filtered based on certain constraints like the balance control and comfort control parameters.

3.4.1 Limitations of Naïve DOF Replacement

Even though this is a simple technique, the physical plausibility of the spliced motion may be a big concern. Hence, the animator has to carefully verify for any flaws in the spliced motion. If the motion is captured at different speeds or has different skeletons, the spliced motion might look unreal. Proper attention must be taken while transitioning between the spliced motions. The orientation of the transplanting joints must be pretty close to that of the base motion.

CHAPTER 4

QUANTITATIVE EVALUATION METHODOLOGY AND EXPERIMENTAL RESULTS

This chapter describes our quantitative evaluation methodology, the various steps involved in comparing motions, and the significance of the ground truth data. The chapter also describes the results obtained from applying our methodology to the comparison of three motion splicing techniques.

The quality of the synthesized motion is one of the most important features of any motion editing technique. Unfortunately, there is no standard method to measure motion quality. We present a novel quantitative evaluation methodology where the motion synthesized using various splicing techniques is compared to ground truth motion data. This is the first time such an approach has been presented. Our method uses one to one comparison between the synthesized data and the ground truth. For the purpose of comparison, we used ground truth data from the Human Motion Database constructed by the Sensory-Motor IntelLigencE Laboratory at the University of Texas at Arlington

4.1 Motion Analysis and Validation Background

Edited motions have to be tested and validated by some standard methods. Kwon *et al.* [KS05] used footprint pattern to validate locomotion. Kim *et al.* [KP03] analyzed the beat pattern in a motion to segment a motion and validate the data. Safonova *et al.* [SH04] showed how to analyze the physical correctness of interpolated motion. Jang *et al.* [JL08] examined the static balance of motion generated for testing physical validity. Ikemoto *et al.* [IF04] used Support Vector Machines to validate the synthesized motion. The output of this SVM technique is only related to how natural the synthesized motion is. Often the comparison results are inaccurate due to the discontinuities in the training data. This technique requires a large computation time for training and testing.

The Figure 4.1 shows a typical support vector machine motion classification method. In Figure 4.1, the green region represents the points closer to the training dataset and the points in the red region represent the corrupt data points or the points that are not related to the training data set. The white curve represents the hyper plane that separated the training data from the corrupt data.

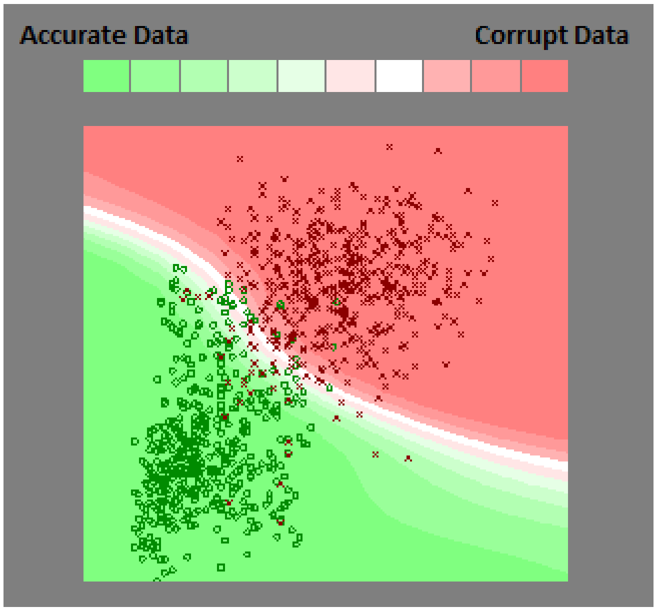


Figure 4.1: Support vector machine used in motion analysis

4.2 The Splicing Dataset of the Human Motion Database

The Human Motion Database (HMD) constructed at the SMILE laboratory contains motion information for different phenomena associated with several aspects of human movement such as a praxicon (a lexicon of human actions), a cross-validation set, generalization, transitioning, co-articulation, splicing, and interactions. This database is primarily focused on assisting motion-editing techniques. The database contains motions that are performed by 50 different subjects. The set of motions in the motion splicing dataset includes actions being performed individually a number of times in the same manner and in combination with other actions. For example, a splicing motion set contains a motion where a subject waves while standing, a motion where the subject drinks water while standing, and a third motion where the subject drinks and waves at the same time while standing. In a similar way, we have 39 splicing motion sets that are combinations of two or more actions. This captured combination motions served as the ground truth in our comparison. Table 4.1 gives the list of all the individual motions and Table 4.2 provides a list of the motions being performed in combination with other actions. The individual actions in Table 4.1 are used as the input for the three splicing techniques and the output-synthesized motion using these techniques are compared with the corresponding motions in Table 4.2.

Table 4.1: List of All Individual Actions in the Splicing Dataset.

Bump	Carry
Chair walk backwards	Clap
Crazy sign alternate	Crazy sign left reverse
Crazy sign left	Crazy sign right reverse
Crazy sign right	Crazy sign together
Drink	Extend arm back
Fan	Flail
Flap	Flex biceps alternate
Flex biceps left	Flex biceps right
Flex biceps together	Hand to ear
Hands behind head	Hit low
Hold hands	Hold suitcase
Incline head	Jog
Jump low	Jump
Kick	Knife left
Nod	Ok sign
Punch	Raise leg
Raise thigh	Raise umbrella
Reach	Rope
Rotate chair	Scratch nose
Sit	Skip
Step on	Strike left A
Strike left	Strike right A
Strike right	Sword down
Touch	Turn
Walk	Wave

Table 4.2: List of All Combination Actions in the Splicing Dataset.

Chair walk backwards X scratch nose	Clap X jump
Clap X turn	Crouch X hands behind head
Dodge X punch	Drink X wave
Fan X ok sign	Incline head X hand to ear
Jog X flail	Jump X kick
Jump X nod	Jump X rope X nod
Kick X punch	Kick X turn
Raise leg X hit low	Raise thigh X extend arm back
Raise umbrella X hold suitcase	Reach X jump low
Rope X nod	Rotate chair X wave
Sit X fan	Sit X ok sign
Sit X punch	Skip X rope
Step on X touch	Strike right A X strike left A
Strike right X strike left A	Strike right X strike left
Sword down X knife left	Turn X nod
Turn X punch	Walk X bump
Walk X carry	Walk X crazy sign alternate
Walk X crazy sign together	Walk X flap
Walk X hold hands	Walk X nod
Walk X wave	

4.3 Synthesized Motions

4.3.1 Experimental Results for the Spatial Body Alignment Technique

Motions were synthesized according to this technique and the results are presented below. The following figures show the motions generated using the input motions from Table 4.1 and the results were compared against the motions in the Table 4.2.

Carrying a suitcase: The Figure 4.2 shows the spliced motion where a person is carrying a suitcase. The synthesized motion is a good example to demonstrate the spatial body alignment splicing technique. The motion is generated from two input motions, motion *A* is subject walking in a slow pace carrying a suitcase and motion *B* is the subject walking in normal pace. We generated a spliced motion where the person carries a suitcase while walking in normal pace.

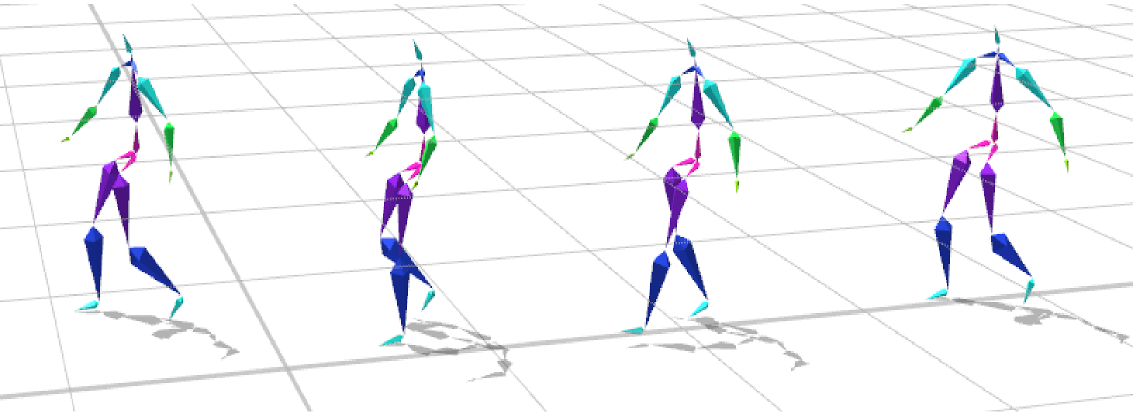


Figure 4.2: Subject Walking While Carrying a Suitcase.

Carrying a box: Figure 4.3 demonstrates the spliced motion generated from the motions, where the subject is carrying a box, while standing at a fixed place and a motion, where a person is walking. The synthesized motion of a person walking with a box looked realistic in some frames and not in the others. This happens due to the wrong selection of points in performing a time alignment. The time alignment is being computed using the lower bodies of the two motions. As one of the two lower bodies is not moving, a temporal misalignment happens at different frames. This is a typical example to demonstrate where the algorithm fails.

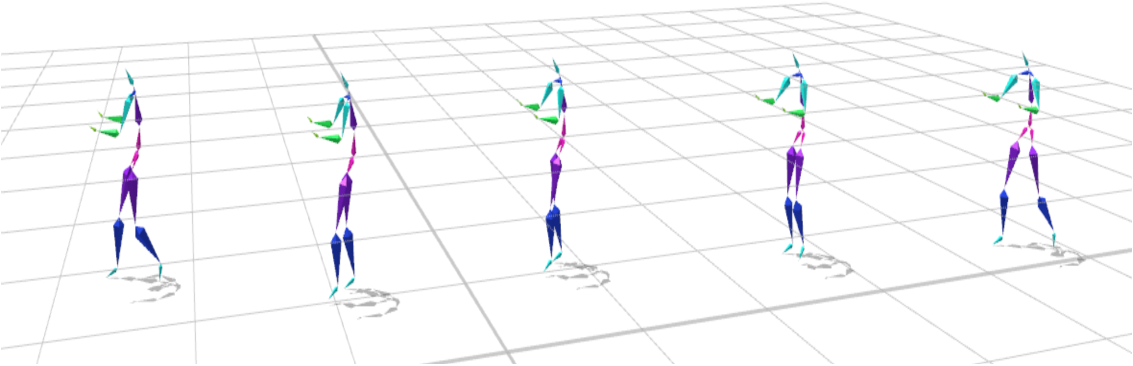


Figure 4.3: Spliced Motion for Walk And Carry.

4.3.2 Experimental Results for the Segmentation-based Technique

Jumping and reaching: Splicing the jump action and the reach action generates the motion depicted in Figure 4.4. The subject is jumping in the first motion and, in the second motion the subject is reaching the hand up while standing. The algorithm best replaces the arms' partial motion of the first action with the arms' partial motion of the second motion, where the similar base partial motion is the lower body motion. In generating this action, the lower body is used as the reference. Therefore, when the lower body motions in the two input motions are similar, their upper body joints are replaced. The best matches between the lower bodies are found, when both legs are on the ground in the jumping action with that of the reaching action.

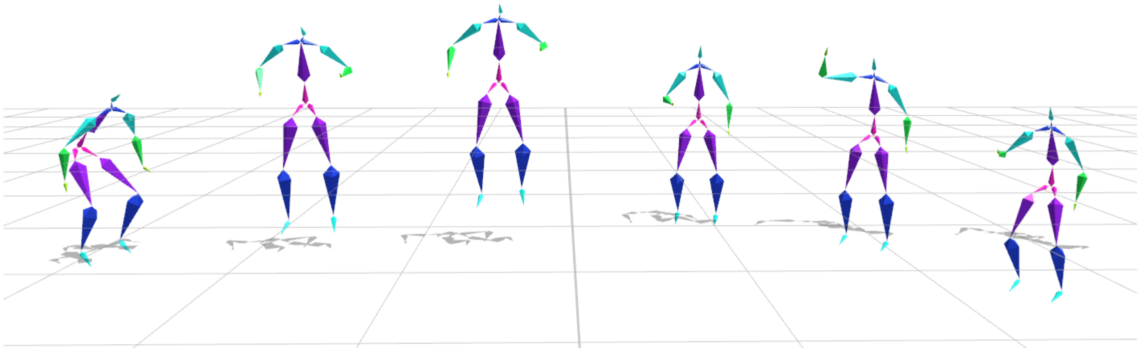


Figure 4.4: Spliced Motion for Jump And Reach.

Jumping and kicking: Figure 4.5 shows the spliced motion synthesized by splicing the kick action with the jump action. Again in this motion, the kick is performed, while the person is standing in the same place. This motion is a typical example to demonstrate the limitation of this technique on accounting for the balance of the synthesized motion. When the person tries to kick, the entire body accommodates the posture to maintain the balance. In this motion, the reference partial motions are the upper body and the leg motion is replaced. The synthesized motions look unrealistic, as the rest of the body does not compensate for balance, while performing the kick action.

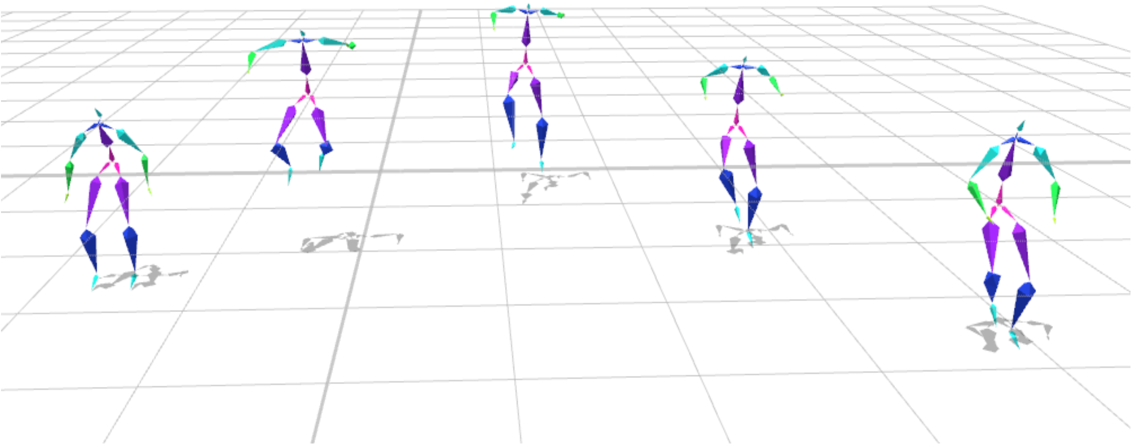


Figure 4.5: Spliced Motion for Jump And Kick.

Jog and Flail: The input motions are jogging and flailing while standing. The reference partial motions are the lower body of the two actions. As the jogging motion comes to an end, the legs' movements are slower and hence there is a better match between the lower bodies of jogging and standing. This allows the actions of the upper body to be swapped. However, the number of frames generated using this technique is limited in any motion. This method does not produce splicing in every motion frame. The method just produces motion for a certain number of frames in the entire motion.

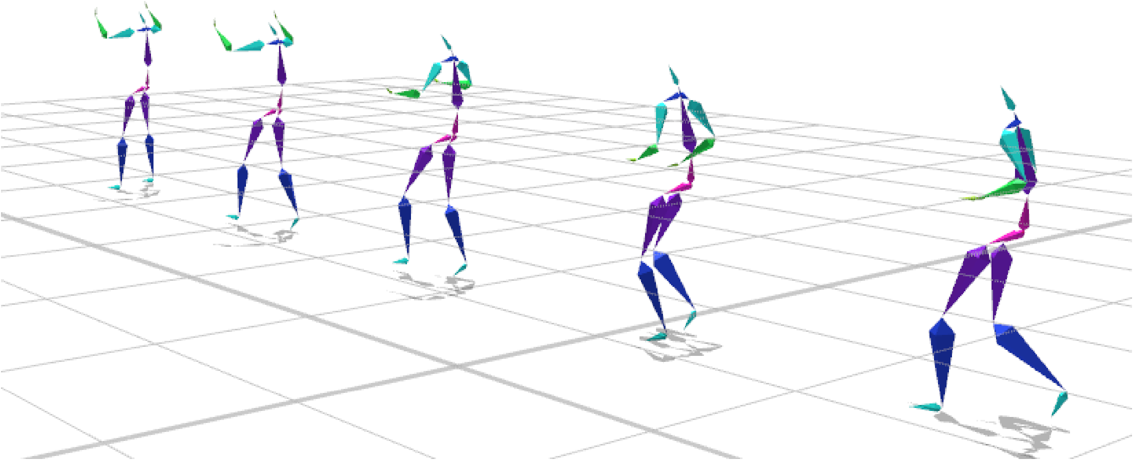


Figure 4.6: Spliced Motion for Jog And Flail.

4.3.3 Experimental Results for Naïve Degree of Freedom Replacement

Sword down knife left: This splice motion consists of the subject moving a knife using the left hand and moving a sword up and down with the right hand. The motion is synthesized from the individual motions where the person moves the knife alone and the person moves the sword up and down. The naïve DOF replacement technique splices the motion effectively. The left arm joints are replaced in the sword down action to synthesize this motion. This is a good example to demonstrate where the Naïve DOF replacement performs well.

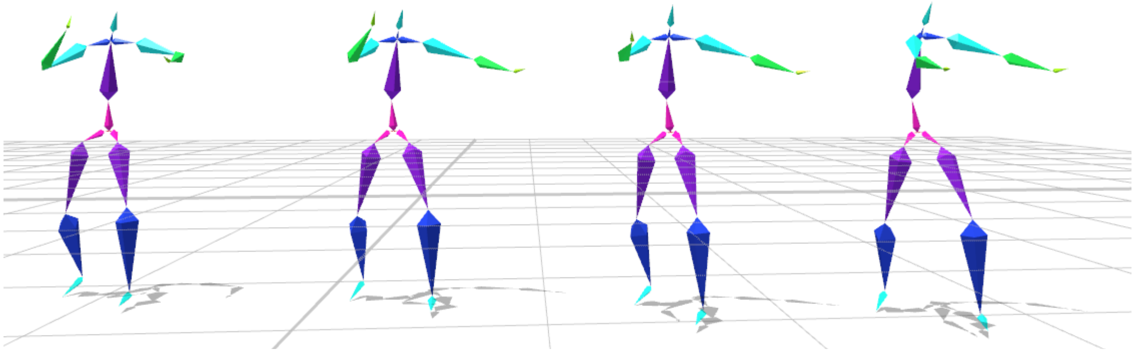


Figure 4.7: Spliced Motion for Sword Down And Knife Left.

Raise leg and reach low: The synthesized motion of raise-leg and reach-low is generated from the actions, where the person raises a leg to a certain point and reaches for the same point. The purpose of the splicing motion is for the subject to reach for his raised leg. The naïve DOF replacement just replaces the hand motion in the raise-leg motion. Hence, the purpose of touching the leg is not achieved, but the two actions are simultaneously performed. In Figure 4.8, the maximum point of raising the leg is achieved in the 4th frame and the lowest point of reach is achieved in the fifth frame. This is one of the limitations of the Naïve DOF replacement method as the purpose of the simultaneous actions is neglected.

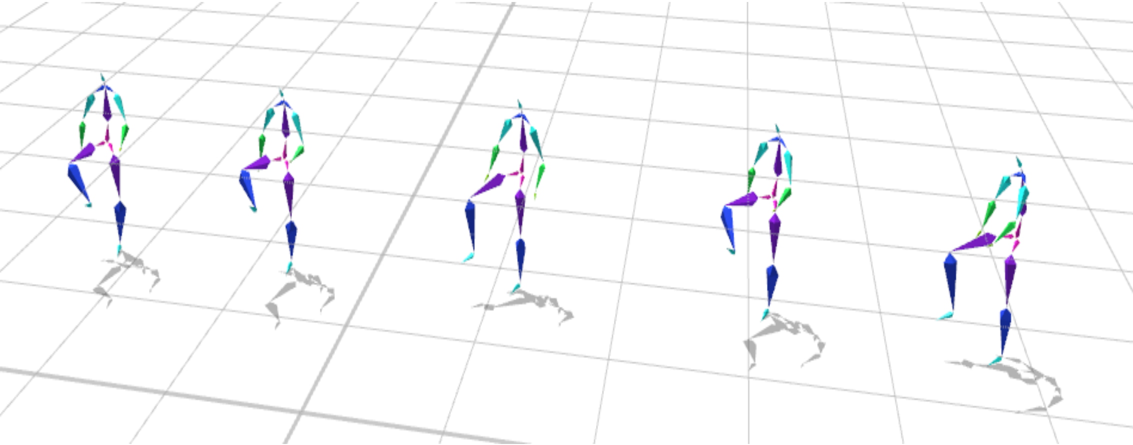


Figure 4.8: Spliced Motion for Raise Leg And Reach Low.

Turn and kick: Figure 4.8 displays the spliced motion generated from the motion of a person turning back and forth 90 degrees and the motion, where a person stands and kicks. This is an example to demonstrate the limitations that naïve DOF has on accounting for the balance of the motion. In the kick motion, the entire body rearranges itself to get a better balance. Therefore, by just replacing the leg motion in the turn actions produces frames that are unrealistic as shown in Figure 4.9. Usually, when a person tries to kick, his spine inclines back to keep the center of mass within the body and to maintain the body balance. However, when applying naïve DOF replacement on these individual motions, the balance of the body is not maintained.

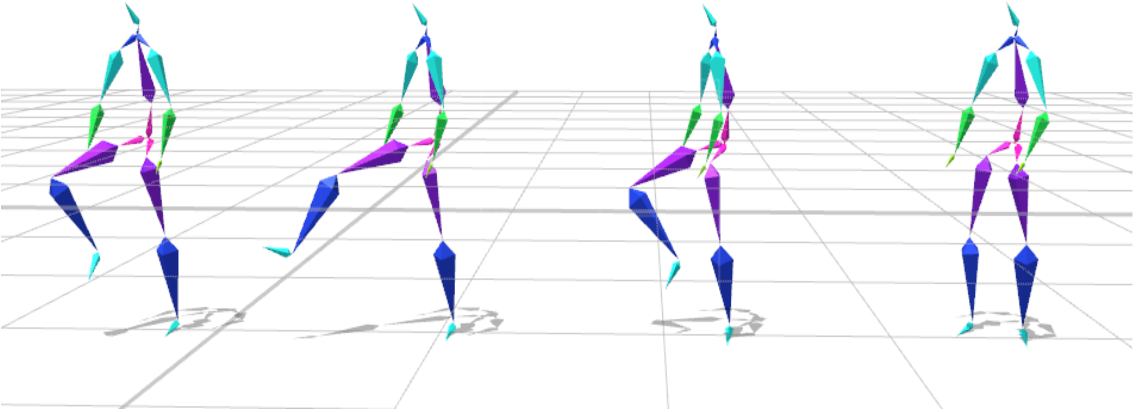


Figure 4.9: Spliced Motion for Turn And Kick.

4.4 The Quantitative Evaluation Methodology

The quantitative evaluation methodology is a three-stage process. During the first stage of the evaluation, each degree of freedom of the combination motion (*M_G*) that is the ground truth is time aligned to each degree of freedom of the spliced motion (*M_S*). In the second stage of the evaluation, the aligned frames are used to compute the time normalized spliced motion. In the third stage, the motion distance for all aligned frames between ground truth and synthesized motion is computed for rotational data (joint angle, angular velocity, and angular acceleration) and translational data (joint position, linear velocity, linear acceleration). Next, each step is explained in detail.

Initially, we perform the time-alignment between motion M_G during a given period of time ranging from frame t_G^0 to frame t_G^1 and motion M_S during a given period of time ranging from frame t_S^0 to frame t_S^1. For each DOF *j* in the skeleton model, we find the time displacement function $\Delta_j = DTW\left(M_G^j(t_G^0 \dots . t_G^1\ ), M_S^j(t_S^0 \dots . t_S^1)\right)$, where $M_G^j(t)$ is the value assumed by DOF *j* at frame *t*. Dynamic time warping finds the optimal matching displacement between M_G^j and M_S^j such that $M_G^j(t)$ corresponds to $M_S^j(t + \Delta_j(t))$. Note that M_G is the reference motion and DTW only uses rotational data for the matching. While computing the best path, we introduce an error factor that limits the number of consecutive horizontal and vertical frame matches. The computed error value is directly proportional to the number of consecutive horizontal and vertical frames being matched. Using the error factor and the cost matrix previously computed for DTW, we compute a new cost matrix:

$$C_{new}(r,c) = C_{old}(r,c) + \left(\frac{n * err}{k-n}\right), \qquad 4.1$$

where $C_{new}(r,c)$ is the computed cost of the r^{th} row and the c^{th} column of the cost matrix $C_{old}(r,c)$ (refer to subsection 3.1.1.2). The error factor *err* is carefully selected by the user, depending on the type of data being modified. The *err* value selected in our comparison is 5.4 times the slope limit (maximum number of consecutive horizontal and vertical steps allowed in the path). The maximum number of allowed steps is 6 (k). The multiplication factor n is given using the following equation:

$$n = \begin{cases} n+1 & if\ i\ (path(t) = i\big(path(t-1)\big) or\ j\ (path(t) = j\big(path(t-1)\big) \\ 0 & otherwise \end{cases} \qquad 4.2$$

Once the time alignment is found for each DOF, the metric $D(M_G, M_S, \Delta)$ between two motions M_G and M_S according to time alignment Δ is defined as:

$$D(M_G, M_s, \Delta) = \sum_{t=t_G^0}^{t_G^1} \sum_{j=1}^{k*3} \left| M_G^j(t) - M_s^j(t + \Delta_j(t)) \right|, . \qquad 4.3$$

where *k*3* is the number of DOFs (joint angles or 3D joint points). This metric considers rotational data (joint angles) for the 0th order derivative (position). We consider a total of six different versions of this metric with all combinations between two different motion representations (rotational data and translational data) and three derivative orders (position, velocity, and acceleration). Any metric is computed as follows: Initially, the time alignment Δ_j is found for all DOFs. Given the time alignment Δ_j, the motion M_S is warped according to Δ_j into a warped motion $M_{S'}$ such that $M_S^j{}'(t) = M_S^j(t + \Delta_j(t))$ for all DOF *j*.

Once the warped motion is obtained, we can generate the time normalized motion from the spliced motion for every frame in the ground truth. We generate the normalized motion using the paths computed for each DOF in the earlier DTW step. For every DOF, a frame of the ground truth corresponds to a frame of the spliced motion. The normalized spliced motion is generated using the values associated with this corresponding frame. The normalized motion should have the same length of the ground truth data to allow a one-to-one matching of frames.

After generating the normalized motion, we may compute the translational data (3D Cartesian coordinates for all joints) for each frame. Given the rotational data and translational data for motion M_G and warped motion M_S', we then compute the first and the second order derivatives. Equation 4.3 gives the distance between the two motions.

For example, let's consider translational data and the first derivative (velocity). First, we compute the time alignment, second we perform the motion warping, third we compute the translational data for both M_G and M_S', then we find the linear velocity and finally, we compute the distance for all the frames between $t_G^0 \text{ and } t_G^1$. For translational data, we use Euclidean distance instead of calculating the absolute difference between values. For a pose associated with a particular frame *t*, the distance is the sum of the Euclidean distances between the corresponding joints in the two different motions. In order to avoid the need for spatial alignment (translation and rotation), we considered only the first six degrees of freedom associated with global position and global orientation to be zeroes.

The experimental results of our quantitative evaluation are shown in Figure 4.10. Each bar in the graph represents the average error values for each motion splicing technique. The green bar shows the error obtained using the spatial body alignment method [HK06]. The red bar represents the segmentation-based method [JL08]. The blue bar corresponds to the naïve DOF replacement technique.

Our quantitative evaluation shows that the spatial body alignment method for splicing upper body action with locomotion is the best method for splicing motion. The reason for this might be that the spatial body alignment method accounts for the correlation of the body movement.

On the other hand, the segmentation-based method is the worst because of the large jerkiness in the spliced motion. We are reasonably certain that this jerkiness is caused by the improper segmentation of the motion. The entire motion is segmented at local maximum and minimum of the total body kinetic energy, which divides a single action into multiple artificial parts. For example, the kick action is segmented into four different segments, since there are three minimum and two maximum values of velocity. When splicing this action on the other motion, if there were only two consecutive segments matching the segments in the kick, the action gets trimmed at undesirable frames creating jerkiness in the spliced motion.

Naïve DOF replacement is a technique that works better for splicing motions, where the primary joints don't overlap. For example, in the case of an action, where the person waves and an action, where the person drinks using different hands, the naïve DOF replacement best splices these motions, as there are no additional constraints. However, in the case of an action, where the person jumps and kicks, this technique splices the motion, which looks unrealistic, as there are absolutely no parameters accounting for the balance and the comfort parameters in a motion.

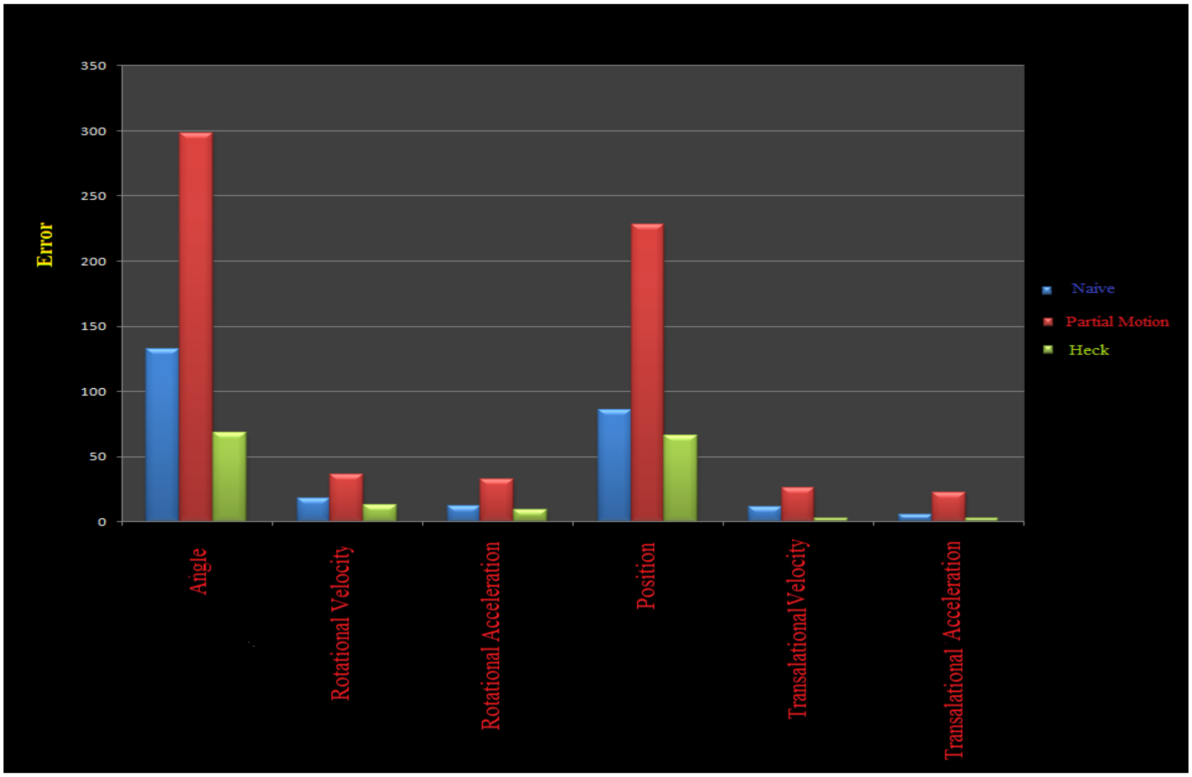


Figure 4.10: Experimental Results of Our Quantitative Evaluation.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Considering the current trend in video games, automatic surveillance, robotics, the need for quality interactive human motion synthesis will only grow. Still, there are many unsolved problems in the motion synthesis area.

This thesis presents a novel way of quantitatively evaluating synthesized motion obtained using different motion-editing techniques. This is the first time a quantitative evaluation is performed on a motion editing technique. There haven't been any real effort in evaluating the synthesized motion, till now the motions have been validated either visually or using a set of training data rather than using the exact ground truth data.

In this thesis, we have implemented three significant splicing techniques and the motions generated using these methods were compared to the ground truth data in the Human Motion Database. The ground truth data consists of a diverse set of motions captured to test any splicing motion. Using 39 sets of motions in the ground truth data and the normalization technique mentioned earlier, quantitative motion evaluation now has an elegant solution. The results of our quantitative evaluation showed that in the spatial body alignment method in which the correlation of the joints is taken in to account, provides the most realistic results. Segmenting a motion at local maximum and minimum of total kinetic energy can generate spliced motions with lesser similarity to the ground truth. Naïve DOF replacement is a good splicing technique, if the joints are carefully replaced. Implementing additional constraints can potentially improve this method.

We suggest further research in motion perception that can help fix various artifacts in motion generation. We also suggest more subjective research to determine the minimum acceptable artifacts in motion synthesis. Better physical models could be developed that take into consideration the style with which the subject performs an action. Future work on ways to combine example-based approaches with the physically based approaches has great potential to create realistic motions with the desirable physical as well as stylistic properties. Modeling the human body is one of the missing links in the motion editing techniques. A perfect physical modeling of the body is necessary to perfectly tweak the joints with respect to other joints and maintain a high correlation through out the whole body.

REFERENCES

[AB96] Amaya K., Bruderlin A., Calvert T.: Emotion From Motion. *Proceedings of Graphics Interface '96*, (1996), 222–229.

[AF02] Arikan O., Forsyth. D. A.: Interactive Motion Generation From Examples. *Proceedings of ACM* Siggraph*'02*, (2002), 483–490.

[AG10] Biswas A., Guerra-Filho G.: A Human Motion Database: The Cognitive And Parametric Sampling Of Human Motion. *Master's Thesis at University Of Texas*, (2010).

[AW00] Ashraf G., Wong K. C.: Generating Consistent Motion Transition via Decoupled Framespace Interpolation. *Computer Graphics Forum* 19.3, (2000), 447-56.

[BC89] Bruderlin A., Calvert T. W.: Goal-Directed, Dynamic Animation of Human Walking. *Proceedings of Siggraph, Vol. 23, No. 3, July 1989*, (1989), 233-242.

[BJ77] Beer F. P., Johnston, Jr.: Vector Mechanics For Engineers: Statics And Dynamics".3rd Ed. *Toronto. McGraw-Hill*, (1977).

[BP93] Badler N.I., Phillips C.B., Webber B.L.: Simulating Humans: Computer Graphics And Animation And Control. *Oxford Univ. Press, New York*, (1993).

[BT90] Boulic R., Magnenat-Thalmann N., Thalmann D.: A Global Human Walking Model with Real-Time Kinematic Personification. *In The Visual Computer, Vol.6, No.6*, (1990), 344-358.

[CM69] Clauser, C. E., McConville, J. T., Young, J. W.: Weight, volume and center of mass of segments of the human body. *AMRL-TR-69-70. Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio*, (1969).

[DT55] Dempster, W. T., Space requirements of the seated operator. *WADC-55-159, AD-087-892. Wright-Patterson Air Force Base, Ohio.* (1955).

[*Gir*87] Girard M., Interactive Design of 3D Computer-Animated Legged Animal Motion. *IEEE CGM,Vol. 7, No. 6, Jun. '87*, (1987), 39-51.

[GJ96] Gose, R., Johnsonbaugh, Jost S.: Pattern Recognition and Image Analysis. *Prentice Hall*, (1996).

[Gle97] Gleicher M., Motion editing with spacetime constraints. *In Proceedings of the Symposium on Interactive 3D Graphics (SI3D '97)*, (1997), 139–ff.

[*Gue*05] Guerra-Filho G.: Optical Motion Capture: Theory and Implementation. *Journal of Theoretical and Applied Informatics (RITA), Brazilian Computing Society, 12(2)*, (2005), 61-89.

[HG10] Harnish B., Guerra-Filho G.: A Comparison And Evaluation Of Motion Indexing Techniques. *Master's Thesis at University Of Texas,* (2010).

[HK06] Heck, R., Kovar L., Gleicher M.: Splicing Upper Body Actions With Locomotion. C*omputer Graphics Forum 25.3, (2006)*, 459–466.

[HP64] Hanavan, E. P.: A Mathematical Model Of The Human Body. *AMRL-TR-64-102, AD-608-463. Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio, (*1964).

[Hor87] Horn K. P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A 4*: (1987), 629–642.

[Hub81] Huber P. J.: *Robust Statistics*. *John Wiley & Sons, New York*, (1981).

[IF04] Ikemoto L., Forsyth D. A.: Enriching A Motion Collection By Transplanting Limbs. *In Proceedings of ACM Siggraph/Eurographics Symposium on Computer Animation* (2004).

[JB89] Jones M. R., Boltz M.: Dynamic Attending and Responses to Time. *Psychological Review 96, 3*, (1989), 459–491.

[JL08] Jang, W.S., Lee W.K., Lee, J.: Enriching a motion database by analogous combination of partial human motions. *In the Proceedings of Visual Computing*, *24(4)*, 2008, 271–280

[KG02] Kovar L., Gleicher M., Pighin F.: Motion Graphs. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21, 3 (July)* , (2002), 473– 482.

[KP03] Kim T., Park S. I., S. Y. Shin.: Rhythmic-Motion Synthesis Based On Motion-Beat Analysis. *In Proceedings of ACM SIGGRAPH '03*, (2003), 392–401.

[KS05] Kwon T., Shin S. Y.: Motion modeling for on- line locomotion synthesis. *In Proceedings of ACM Siggraph/Eurographics Symposium on Computer Animation, (2005)*.

[LC06] Lee K. H., Choi M. G., J. Lee.: Motion Patches: Building Blocks For Virtual Environments Annotated With Motion Data. *In Proceedings of ACM Siggraph '06*, (2006), 898–906.

[LP02] Liu C. K., Z. Popovic.: Synthesis Of Complex Dynamic Character Motion From Simple Animations*. In Proceedings of ACM Siggraph 2002, Annual Conference Series*, (2002), 329- 334.

[LS02] Lee J., Shin S. Y.: General Construction Of Time-Domain Filters For Orientation Data. *IEEE Transactions on Visualization and Computer Graphics, 8(2):* (2002), 119–128.

[*Per*95] Perlin K.: Real Time Responsive Animation With Personality. *IEEE Transactions on Visualization and Computer Graphics 1, 1 Mar. '95*, (1995), 5–15.

[PB02] Pullen, K., and Bregler, C. Motion Capture Assisted Animation: Texturing and Synthesis. *ACM Transactions on Graphics (Proc. SIGGRAPH 2002) 21, 3 July '02*, (2002), 501–508.

[PM92] Pandya A. K., Maida J. C., Aldridge A. M., Hasson S. M., Woodford B. J.: The Validation of a Human Force Model to Predict Dynamic Forces Resulting from Multi-Joint Motions. *Technical Report 3206 NASA, Houston, Texas* (1992)

[PT99] Press W. H., Teukolsky S. A., Vetterling W. T., Flan- Nery, B. P.: Numerical Recipes in C – The Art of Scientific Computing, *Second ed. Cambridge University Press*. (1999).

[SH05] Safonova A., Hodgins J. K.: Analyzing The Physical Correctness Of Interpolated Human Motion. *In Proceedings of the 2005 ACM Siggraph/Eurographics Symposium on Computer Animation (SCA '05)*, (2005), 171–180.

[SH04] Safonova A., Hodgins J. K., Pollard N. S.: Synthesizing Physically Realistic Human Motion In Low Dimensional, Behavior-Specific Spaces. *In Proceedings of ACM Siggraph '04*, (2004), 514–521.

[TS02] Tak S., Song O., Ko H.: Space Time Sweeping: An Interactive Dynamic Constraints Solver. *In Computer Animation, June '02*, (2002), 261–270.

[WP95] Witkin A., Popovic Z.: Motion Warping. *In Proceedings of Annual Conference Series*, *ACM Siggraph, Aug '95,* (1995), 427-433.

[YN00] Yamane K., Nakamura Y.: Dynamics Filter – Concept And Implementation Of On-Line Motion Generator For Human Figures. *In the 2000 IEEE International Conference on Robotics & Automation, April '00, (2000),* 688–695.

[Mots] "Motion Capture Studio Services - Performance Capture " *Motion Capture Animation Studio in Dallas, TX | MOCAP*. Web. 11 Dec. 2010. <http://motusdigital.com/motion-capture/>.

BIOGRAPHICAL INFORMATION

George Thekkanath Raphael completed his Bachelors of Technology in Electrical Engineering from the University of Calicut, India. After that he joined the University of Texas at Arlington as a graduate student of Electrical Engineering department. During the course of his graduate studies he worked under the guidance of Dr. Gutemberg Guerra Filho and Dr Venkat Devarajan.