SYNTHESIZING AND EVALUATING DATA-DRIVEN MOTION TRANSITIONS

By

Jing Wang

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements for

the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

December, 2005

Nashville, Tennessee

Approved:

Professor Bobby Bodenheimer

Professor J. Michael Fitzpatrick

Professor Benoit Dawant

Professor Randolph Blake

Professor Richard A. Peters

To my dearest parents and Zhihua

ACKNOWLEDGMENTS

I would like to thank my advisor and mentor, Bobby Bodenheimer, for his advice, encouragement, and patience. He provided me the opportunity to work in the Computer Graphics and Animation lab and guided and supported my work. I am grateful to members of my committe, Randolph Blake, J. Michael Fitzpatrick, Benoit Dawant, and Richard A. Peters for providing invaluable insights. Special thanks to my colleagues Chiristina de Juan and Besty Williams, who have never hesitated to help and encourage me.

I would like to thank my entire family for their enduring love and support. Thanks also to Lin Tang and Xia Li for their wonderful friendship. Most importantly, I wish to thank my loving boyfriend Zhihua for inspiring my passion and having faith in me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

Animation is a sequence of images that people perceive as a continuous movement. These images could be hand-drawn or computer generated. Computer animation starts with models of objects then generates from the models sequences of images. Computer animation is widely used in video games and special effects in films. It also has many applications outside of entertainment such as learning and training in virtual environment, education, and information visualization.

Computer animation in three dimensions involves constructing a virtual world where characters may move and interact. Human characters in such a virtual world are described as virtual humans and their movement and interaction as behaviors. Designing a rich repertoire of appealing behaviors for virtual humans is an important problem for virtual environments and video games. It is a challenging task because of the complexity and variety of human motion and the extensive experience of humans' observance of movements.

I.1 Goal

One approach to designing a repertoire of human motion is to collect segments of motion generated by motion capture, a technique for recording motion from a human performer, and pre-process it to form a structure that can be traversed in various orders to re-sequence the data in new ways. In such an approach creating visually pleasing transitions between motion streams is critical. Such transitions are the topic of the research described in the thesis. One goal of this research is to develop methods for determining optimal transition points. Secondly, we want to develop ways to produce visually compelling transitions without manual intervention.

How can we assess the quality of the motion transitions generated? We could develop a measurement formula and compare the output signal, but "visual appeal" is based on subtleties that are difficult to characterize; we could use machine learning to "learn what represents good motion", but it is often difficult to interpret the learned function; we could cross validate the data to assess the generalizability of the method, but it does not tell us about the visual quality; or we could empirically evaluate the result.

Empirically evaluating the plausibility of motions by conducting user studies is an effective way for determining the quality of the motions. The results of user studies can be used to measure some quantity of perceptual difference or be used to compare different methods for generating motions. Another goal of this research is to evaluate methods for synthesizing motions. We want to use empirical evaluation as a method to compare our methods for detecting transition points and creating transitions with other methods, and to measure the sensitivity of users to changes in motion transitions.

I.2 Research Contribution

Much animation research deals with the development of methods to specify, create, and evaluate visually pleasing motion. New motion can also be generated by concatenating existing motion clips in a motion library instead of capturing them. The proper selection of transition points, points at which the motion will change from one segment of captured motion to another segment, is important in such an approach. Because these transition points represent discontinuities in the motion stream, selection of good transition points can be crucial to the quality of the resulting motion. A cost function is used to calculate the cost of transitioning from one frame to another. The cost function is a function that takes two poses of a figure, or equivalently in our terminology, two frames of motion, and evaluates quantitatively how similar they are. That is, it is a metric in the configuration space of the figure. Several researchers [Lee et al. 2002; Kovar et al. 2002a; Arikan and Forsyth 2002] have proposed different cost functions and they are all parameterized through user-selected weights. In this work, we compute a set of optimal weights for the cost function proposed by Lee et al. [2002] using a constrained least-squares technique (Chapter IV). The weights are then evaluated in two ways: first, through a cross-validation study and second, through a medium-scale user study. The cross-validation shows that the optimized weights are robust and work for a wide variety of behaviors. The user study demonstrates that the optimized weights select more appealing transition points than the weights originally used by Lee et al. [2002].

Blending is a basic way to create transitions between motions. It is one way to produce visually compelling and optimal transitions without manual intervention. However, generating high quality transitions using blending is still difficult and involves significant manual labor. An animator often needs to go back and forth to modify parameters for blending to obtain a pleasing transition. Some automatic systems simply pre-specify a fixed blend length for all motions. The blend length or duration of the transition is a critical component in the visual fidelity of a spliced animation stream. We develop two methods for determining an optimal blend length for motion transitions, the geodesic distance method and the velocity method (Chapter V). These methods are suited to different types of motion. The geodesic distance method works well for locomotions that are cyclic in nature, i.e., motions that are repetitive or predictable such as walking. The velocity method works well for activities which are not often predictable such as boxing and free-style dancing. In these motions a user does not have an a priori expectation of what the next pose will be. We call these motion "physical activities" as a way of categorizing them separately from cyclic motions, although obviously walking is a physical action. Although, not optimal, we decided this nomenclature was superior to terms such as "acyclic", which carry connotation we wish to avoid. These categories are necessarily loose. It is not the focus of this dissertation to try to categorize motions. Rather, we are trying to label broad sets of motion data that work well with these methods in a reasonable way. These methods are empirically evaluated through user studies. We find (1) that visually pleasing transitions can be generated using our optimal blend lengths without further tuning of the blending parameters; (2) by conducting user studies users prefer the transition methods we developed over a generic fixed-length blend. (3) that users prefer the geodesic distance method for locomotions and the velocity method for physical activities. To our knowledge, the present work is the first that is explicitly concerned with determining the optimal blend length of a transition.

We also ran a set of experiments to determine the sensitivity of subjects to changes in blend length of motion transitions, i.e., to estimate the psychometric function of the just noticeable difference (Chapter VI). The "just noticeable difference" is the amount by which something must be changed for the difference to be noticeable. We conducted the just noticeable difference study on two transition specifications, what we call the "start-end specification" and the "center-aligned specification". We found that people are very sensitive to changes in blend length of motion transitions on both transition specifications. More specifically, the studies show that people can differentiate between transition lengths that differ by -7 (i.e., shorter by 7 frames from a reference duration) or 10 frames (i.e., longer by 10 frames from a reference duration) for center-aligned transitions, and by 2 or -3 frames for start-end transitions.

The methods and experimental evaluation described later in this work give guidance to designers of animation systems who wish to incorporate automatic methods of determining transition points and varying blend lengths into their systems, such as video games. This work also opens a door for a number of other interesting psychophysical experiments that could be conducted on assessing motion transition methods. For example, one could conduct experiments using different transition methods to categorize motions, and determine how much, if any, overlap there is. One could determine classes of motions and transitions for which people are sensitive or insensitive to a transition method.

I.3 Overview

This chapter addressed the goal and contribution for this thesis. The remainder of this thesis is organized as follows. Chapter II provides some needed background and notation of this work. Chapter III presents recent research in motion editing, motion transition creation, and empirical evaluations. Chapter IV presents the weight optimization of a cost function for picking transition points. Chapter V presents the methods developed for creating motion transitions. Chapter VI presents the just noticeable difference study for motion transitions. Chapter VII concludes this thesis and presents the future work.

CHAPTER II

BACKGROUND

Traditionally, an animation was in the form of hand-drawn, 2D images. In modern 3D animation, 3D models for the character and environments are generated and animated, and then a sequence of images are rendered from the 3D models. A simple 3D animation might be just moving the camera or the relative motion of rigid bodies in the scene, while many sophisticated animation techniques have been developed to generate realistic animated scenes.

Many of the principles of traditional animation can be applied to 3D computer animation [Lasseter 1998]. These principles, such as squash and stretch and timing, were developed in the 1930's at the Walt Disney studios. They were developed to make animation, especially character animation, more realistic and entertaining. This chapter gives an overview of the background of character animation and notation of mathematics involved in this thesis.

II.1 Character Animation

II.1.1 Hierarchical Model

A major part of animation is motion control. One common way to produce the motion of an animated character is to use an articulated model such as the one shown in Figure II.1. The animation of a character is described by the movement of multiple, hierarchical, articulated joints. The articulated joints are connected by rigid lengths in a hierarchy for a character. The hierarchy is then combined with a 3D geometric model of the human or creature to produce animation. The 3D geometric model is normally a polygonal mesh. The animation specifies the trajectories or orientations of the joints in the skeleton. The transformation of a parent joint propagates down to all of its sub-joints. For example, a rotation of a hip joint rotates the entire leg, just as in human movement. The number of

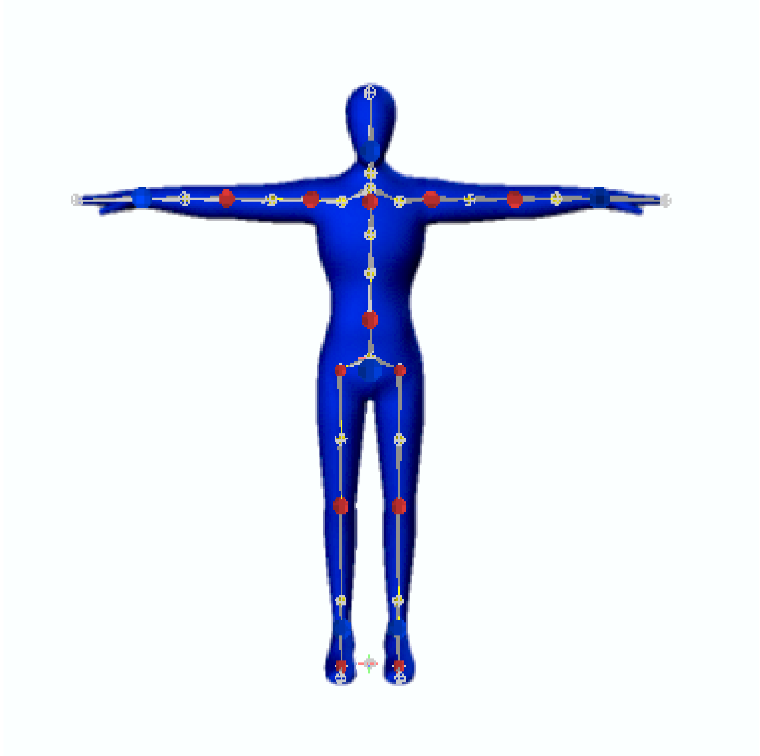


Figure II.1: An articulated human model, both the structure of joint hierarchy and the geometric model are shown.

degrees of freedom (DOF) of a joint is the number of parameters of the joint which may be independently varied. For example, moving a simple rigid object such as a sphere requires six degrees of freedom. Some of the human joints have three DOFs but most have one or two. For example, the shoulder joint is usually modeled with three DOFs, but the elbow joint is usually modeled with only two DOFs. The root of the skeleton normally has three translational DOFs and three rotational DOFs.

II.2 Motion Generation

There are roughly three categories for producing 3D character animation: keyframing, procedural methods, and motion capture. In keyframing, the animator specifies key values for the animated DOFs and the computer interpolates between these values. Procedural methods specify motions algorithmically. Motion capture is the process of recording motions of a performer and mapping them to a virtual character. These three techniques will be discussed in this section.

II.2.1 Keyframing

Keyframing is the oldest type of animation technique. It is still the dominant type of animation used in film. Keyframing requires the animator to outline a motion by specifying key positions for the object being animated. Keyframing was originally used in a traditional 2D animation such as *Tom and Jerry*, where all the frames of the animation had to be drawn by hand. The drawing or painting is usually done by more than one person. The lead animators will draw the important frames, or the *keyframes*. Associate and assistant animators are responsible for drawing the intermediate frames, or the *in-betweens*.

With computer animation, the term *keyframe* has been generalized to apply to any variable whose value is specified at important frames. The computer calculates the in-betweens by interpolating. Computer software has been developed for generating keyframing animation. The software provides graphical interfaces for animators to model, animate, and render the animation. Keyframing gives the animator a fine level of control of the animation. However, it requires intense manual labor to generate a finished product and is thus very time consuming. For example, the animated film *Shrek* took almost three years to produce.

II.2.2 Procedural Methods

Procedure methods generate motions by following steps in an algorithm. The algorithm can be based on laws of physics, or an approximation of those laws. In particular, dynamic simulation is a type of procedural method. Dynamic simulation uses physical properties associated with the graphical character to generate physically accurate motion. Dynamic simulations are well suited to generate simulations of fluids and phenomena such as water and fire because the motions are dominated by physical laws. However, achieving realistic and appealing character animations using dynamic simulation is extremely difficult because it is hard to find the underlying control strategy for character motion. Moreover, the simulated motion is often lacking in detail, which conveys the mood and individuality

Figure II.2: Runner in a park: All the objects in this image were animated using dynamic simulation. Image courtesy of the Graphics, Visualization and Usability Center, Georgia Institute of Technology.

of the character. Figure II.2 shows an example of dynamic simulation; all the objects were animated using dynamic simulation, the running, the child on the swing, and the clothing.

The advantage of procedural methods is the convenience of high level control. After a control system is built, the user can create animation by giving high-level commands such as walk, run, or jump. Moreover, dynamic simulated clothing, hair, and muscle and their interaction with the surfaces of the figure has shown to contribute significantly to the appearance of an animation as we can see in the animated film *The Incredibles*.

II.2.3 Motion Capture

Motion capture is the evolution of a technique known since the 1930's called rotoscoping, a method used in 2D traditional animation, which traces motions from moving video into an animation. Motion capture has grown to be very popular for generating realistic motion and has become a viable option for computer animation production. The nice thing about motion capture is that it allows you to do scenes that would normally be expensive or impossible. We believe that, as the technology develops, motion capture will become one of the basic tools of animators.

In motion capture, sensors attached to various parts of the performer's body communicate with a recording device. The sensors report position and rotational information. The motion capture process maps the data from each sensor on the performer to that sensor's corresponding node in the virtual skeleton. An articulated, hierarchical rigid-body model is then constructed by analyzing the captured input data. There are two popular types of systems for motion capture, optical and magnetic. Magnetic motion capture systems utilize sensors (receivers) attached to the joints of the performer's body to measure the spatial relationship to the transmitter source. The advantage of the magnetic approach is the lack of occlusion problems normally associated with optical systems. One of the limitations of magnetic system is that the tracker's sensitivity to metal can result in irregular output. Figure II.3 shows a performer in a typical configuration of magnetic motion capture sensors. Optical motion capture systems utilize video cameras to record the positions of reflective markers that are attached to joints of the performer's body. The marker images are matched from the various camera views (three video cameras at least) to compute their 3D positions. Figure II.4 shows an optical motion capture apparatus. Compared to magnetic system, optical system has much larger range and the data obtained from optical system has greater accuracy. However, optical system is normally more expensive. Also, as mentioned earlier, optical systems suffers from occlusion problems where one or more markers were hidden by actions of the performer.

Motion capture provides an easy way of generating many human motions. The process automatically captures the subtle details of human motion that convey the personality and the mood of a character. For this reason, motion capture is very popular in video games and special effects in films. However, problems exist when using motion capture to record

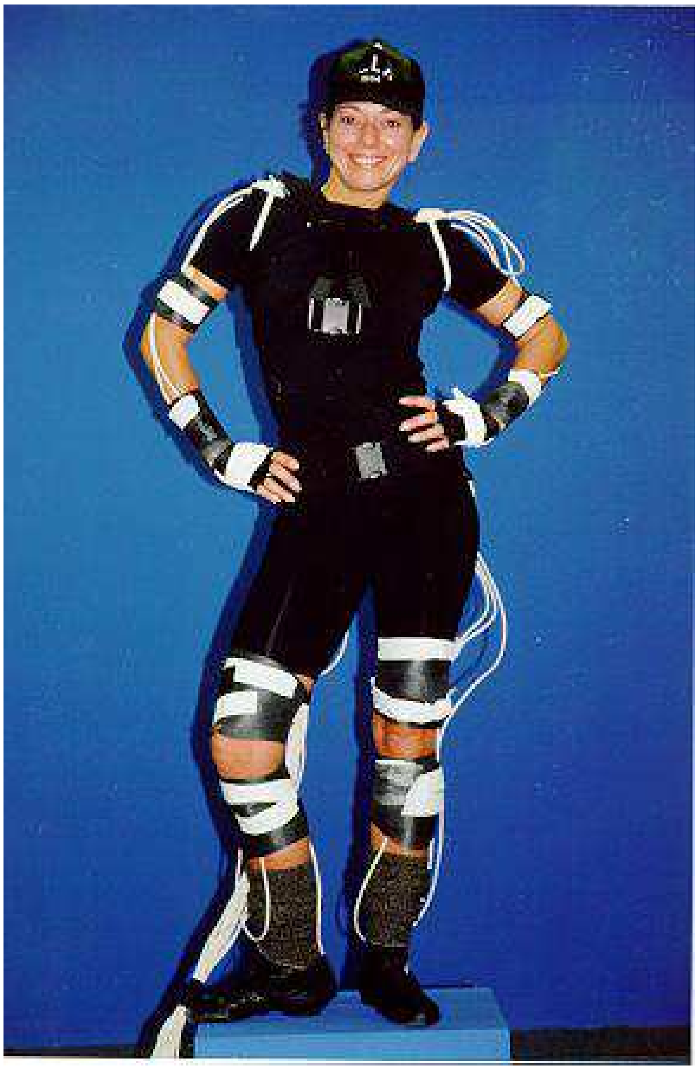


Figure II.3: A performer in a typical configuration of magnetic motion capture sensors. Image courtesy Bobby Bodenheimer.

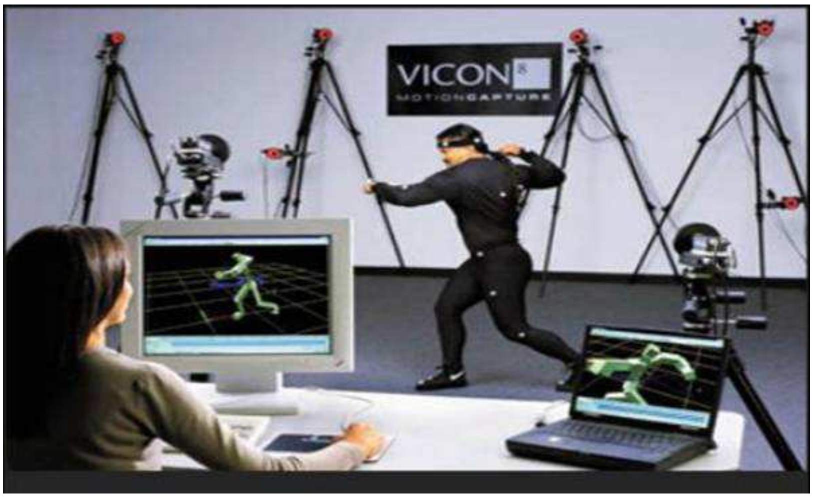


Figure II.4: An optical motion capture apparatus. Image courtesy of Vicon Motion Systems.

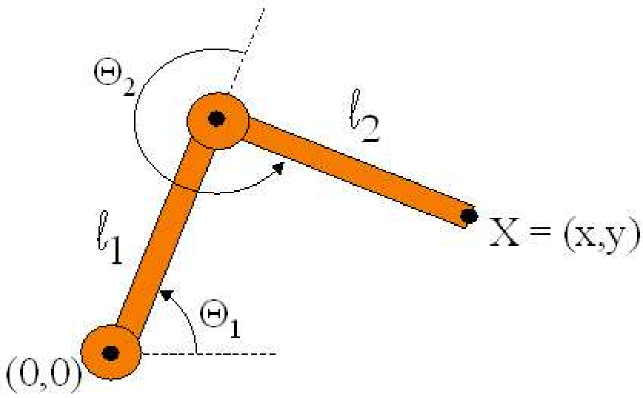


Figure II.5: Forward kinematics solves the position of the end effector X, given the angles θ_1 and θ_2 of each of the joints. Inverse kinematics solves the angles θ_1 and θ_2 for given desired position of the end effector X.

motion. Motion capture data is often noisy and contains gross errors that were introduced in the capture process. Another limitation of motion capture data is that it requires a recording session. Once the motion is finished recording, it is the motion we have. It is hard to modify and re-use in new contexts, and small changes to the data might destroy the realism of the motion.

Additionally, motion capture allows one to construct a large library of raw motion, but processing that motion into a finished product such as a video game is still a labor intensive process. Minimizing the amount of manual intervention in processing motion capture data has been the focus of much recent research. Extending the use of these motion libraries is the motivation for much of the work described in this thesis.

Inverse Kinematics

Inverse kinematics is often used in motion capture system to support editing the data. Forward and inverse kinematics are terms to describe mapping from the space of inputs to the space of outputs of mechanical systems. Forward kinematics solves the position of the end effector X, as shown in Figure II.5, given the angles θ_1 and θ_2 of each of the joints. On the other hand, inverse kinematics solves the angles at all of the joints for given desired position of the end effector. Inverse kinematics is important for 3D character animation. It allows the animator to treat a 3D character's limbs as a kinematics chain. For instance, the animator can manipulate the arm by moving the hand (the end effector). In motion capture process, raw motion capture data are usually made up of points representing sensors moving frame-by-frame in 3D space. Inverse kinematics will be often used to produce desired joint angle trajectories given raw motion capture data. It solves for the angles of the body joints and compensates for the fact that the sensors are often offset from the actual joint's center.

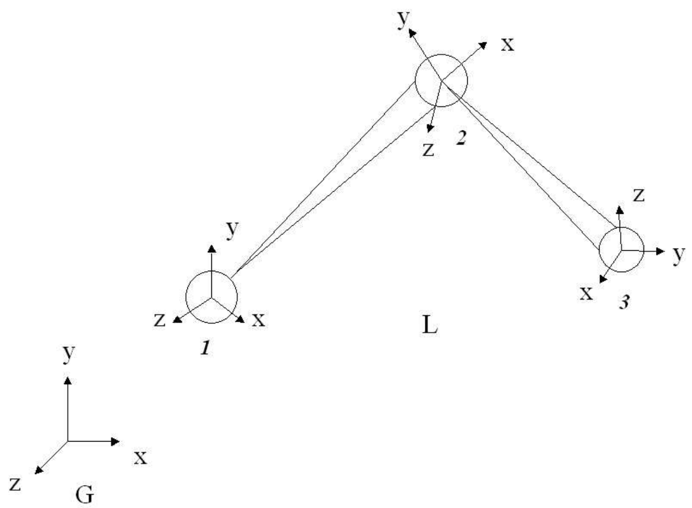


Figure II.6: A joint with local coordinate system. G: global coordinate system. L: local coordinate system.

II.3 Representation of Motion and Interpolation

As mentioned in Section II.1.1, one common way to represent the motion of a character is to use an articulated model. The motion of a skeletal model is specified by a global translation and orientation, and local orientations of the joints. Figure II.6 shows a joint with local coordinate systems. There are several ways to describe rotations in three dimensions. Euler Angles and quaternions are the most common ways and are discussed in the following sections.

II.3.1 Euler Angles

Euler Angles are used to represent rotations in many fields. According to Euler's rotation theorem [Murray et al. 1994], any rotation may be described using three angles at most. Euler angles represent rotation in 3D Euclidean space as a product of three successive 2D coordinate rotations ϕ, θ, ψ about the x-axis, the y-axis and the z-axis. Many different orderings of these axes of rotation can be used to represent a 3D orientation. Elementary rotation matrices around x, y and z axis are given by

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{II.1}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{II.2}$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{II.3}$$

As a representation of rotations, Euler angles have drawbacks. One drawback is that a single rotation can be represented by several different sets of Euler angles. We may also encounter so-called "gimbal lock", the loss of one degree of rotational freedom. Moreover, Euler angles are difficult to interpolate smoothly, or in a coordinate-independent way.

II.3.2 Quaternion Calculus and Exponential Map

Quaternions can also be used to represent rotation [Shoemake 1985]. The quaternion was invented by William Hamilton, an Irish mathematician, as an extension of complex numbers. A quaternion is composed of four elements. It is now often interpreted as (s, $\mathbf{v}$) where s is a real number and $\mathbf{v}$ is 3D vector. Quaternion addition is defined by

$$q_1 + q_2 = (s_1 + s_2, \mathbf{v_1} + \mathbf{v_2}) \tag{II.4}$$

Quaternion multiplication is not commutative but is associative. Quaternion multiplication is defined as

$$q_1 \cdot q_2 = (s_1, \mathbf{v_1}) \cdot (s_2, \mathbf{v_2}) = (s_1 s_2 - \mathbf{v_1} \cdot \mathbf{v_2}, s_1 \mathbf{v_2} + s_2 \mathbf{v_1} + \mathbf{v_1} \times \mathbf{v_2}) \tag{II.5}$$

The quaternion $(1,(0,0,0))$ is the multiplicative identity; that is, $(s,\mathbf{v}) \cdot (\mathbf{1},(\mathbf{0},\mathbf{0},\mathbf{0})) = (\mathbf{s},\mathbf{v})$. The inverse of a quaternion is denoted by q^{-1}, which is defined as

$$q^{-1} = (s, -\mathbf{v}) \,/\, \|q\| \tag{II.6}$$

where $\|q\|$ is the norm of the quaternion and given by $\|q\| = \sqrt{s^2 + \|\mathbf{v}\|^2}$. A unit quaternion is a quaternion q for which $\|q\| = 1$. A unit quaternion can be represented as

$$q = \cos\theta + \hat{u}\sin\theta \tag{II.7}$$

where $\hat{u}$ is a 3D vector having length 1. A unit quaternion represents the rotation of the 3D vector $\mathbf{x}$ by an angle 2θ about the axis $\hat{u}$. When plotted in 4D space, these unit quaternions lie on a sphere of radius one. Rotation of a vector $\mathbf{x}$, represented as (0, ($\mathbf{x}$)) by quaternion q is given by

$$\mathbf{x}^{'} = q \cdot \mathbf{x} \cdot q^{-1} \tag{II.8}$$

The exponential form of quaternion is defined as

$$q = \exp(\hat{u}\theta) = \cos\theta + \hat{u}\sin\theta \tag{II.9}$$

which is consistent with the Taylor series definition of e^x. The power and the logarithm of a unit quaternion then can be defined as

$$q^t = \cos(t\theta) + \hat{u}\sin(t\theta) \tag{II.10}$$

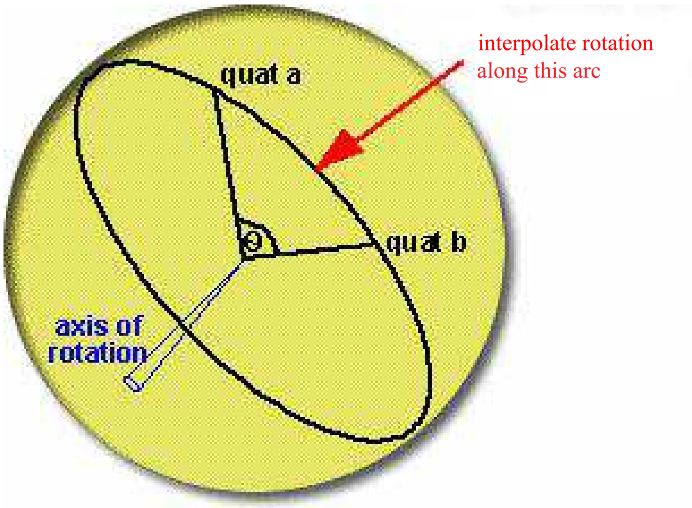


Figure II.7: Visualization of quaternions on a sphere.

$$\log(q) = log(\exp(\hat{u}\theta)) = \hat{u}\theta \tag{II.11}$$

We can visualize unit quaternions as a rotation in 4D space where the $\hat{u}$ forms the axis of rotation and the θ forms the angle of rotation. All the unit quaternions form a sphere of unit length in the 4D space as shown in Figure II.7. This sphere of unit quaternions forms a sub-group, S^3, of the quaternion group and the spherical metric of S^3 is the same as the angular metric of $SO(3)$. Given two unit quaternions q_1 and q_2, the unit quaternion $q_3 = q_2^{-1}q_1$ gives a rotation that transforms the orientation q_1 into a new orientation q_2. The value of $\log(q_2^{-1}q_1)$ is a vector $\hat{u}\theta$ which represents the rotation about $\hat{u}$ by an angle 2θ. The norm of $\hat{u}\theta$, which equals to θ (the angle of rotation), then can be understood as the geodesic distance between two orientations q_1 and q_2. Thus, as we will see in later chapters, $\left\|\log(q_2^{-1}q_1)\right\| = |\theta|$, where θ is the angle subtended by q_1 and q_2 along a great circle of the hypersphere.

Notice that a quaternion q and its negative $-q$, which is $(-s, -\mathbf{v})$, both represent the same rotation because $-q$ indicates a negative rotation around the negated axis. Therefore, the geodesic distance between two orientations q_1 and q_2 should be

$\min(\left\|\log(q_2^{-1}q_1)\right\|, \left\|\log(q_2^{-1}(-q_1))\right\|)$.

II.3.3 Interpolation

Interpolation is an important issue for most computer generated animations. Available data are usually discrete and it's desirable to estimate values in between sample data points. Interpolation determines a curve that passes through given control points, such as keyframed data by the animator. The interpolation method chosen depends on the properties one desires the resulting curve to have. There are many interpolation methods; linear interpolation and spline interpolation are discussed in this section.

Linear interpolation is the most popular and widely used reconstruction method. Linear interpolation in one dimension is simply connecting control points with straight lines. More specifically, let α be a number between 0 and 1, the linearly interpolated value $P(u)$ is defined as follows:

$$P(\alpha) = (1-\alpha) \cdot P_0 + \alpha \cdot P_1 \tag{II.12}$$

where P_0 and P_1 are the two control points.

An alternative method performs interpolation using splines. Spline interpolation consists of the approximation of a function by means of series of polynomials over adjacent intervals with high order continuity. Common used splines include Bezier splines, B-splines, and Catmull-Rom splines. B-splines are perhaps the most popular in computer graphics applications. One of the advantages of using splines to interpolate is higher order of continuity. However, compared to linear interpolation, splines are more difficult and expensive to implement.

Interpolation of Orientations

Many applications in computer animation interpolate orientations using Euler Angles. As mentioned earlier, we may encounter "gimbal lock" , the loss of one degree of rotational freedom, using Euler Angles. Quaternions are safe from gimbal lock and have been used for years to handle spacecraft [Shoemake 1985]. We can interpolate orientations represented by quaternions on the 4D sphere formed by unit quaternions. However, if interpolating by cutting across the sphere, the rotation would speed up in the middle. An interpolation of orientations as quaternions without speeding up corresponds to a great arc along the sphere as shown in Figure II.7. It is called *spherical linear interpolation* and given by

$$Slerp(q_1, q_2; u) = q_1 \left(q_1^{-1} q_2\right)^u \tag{II.13}$$

II.4 Psychophysics

Plausibility rather than accuracy is acceptable for computer graphics and animation because of the limitation of observer's visual perceptions. For example, it is increasingly difficult for an observer to distinguish many of the computer-generated or computer-altered images from photographs. To determine or predict the visual quality of computer generated animations or images, it is often necessary to measure the sensitivity of observers to physical parameters involved in the process.

Psychophysics is the scientific study of relationships between physical stimuli and perception. In a typical psychophysical experiment, a variable of physical simulation is applied to a subject and then the corresponding variable of his response is recorded. A psychometric function reports the underlying relation between the performance on psychophysical tasks and physical stimulus level. It is normally a plot of the percent of a particular response against stimulus level. Figure II.8 shows an example of a psychometric function [Leek 2001]. A detection threshold is a limit to perception, i.e., the smallest amount of stimulus required for the subject to detect. It can be measured on the psychometric function developed by psychophysical experiments.

There are two general strategies for choosing the sequence of stimuli to obtain a complete characterization of psychometric function and detection thresholds, methods of constant stimuli and adaptive methods. In methods of constant stimuli, the experimenter decides in advance which stimuli to use, and how many of each to present. The stimuli are normally presented in a random series. However, methods of constant stimuli are often

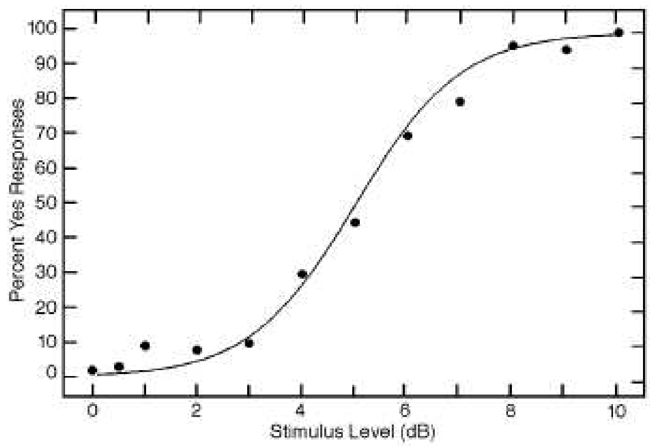


Figure II.8: An example of a psychometric function, measuring the number of times a particular stimulus was detected, depending on what the strength of the stimulus was. Image from [Leek 2001].

expensive in terms of experiment time. Adaptive methods of measurement have been developed with the goal of preserving accuracy and reliability, while maximizing efficiency and minimizing subject and experimenter time [Leek 2001]. It is based on the fact that some trials are not as informative about threshold as others in terms of the stimulus level the trial was placed. Essentially, in adaptive methods, the stimulus on each trial is determined by the stimuli and responses that occurred in the previous trial or sequence of trials. In sum, the method of constant stimuli has the advantage of reducing the influence of stimulus history, and the adaptive methods increase the efficiency of threshold estimation by concentrating on stimuli near the threshold.

II.4.1 Design of Adaptive Procedures

An experimenter must make several decisions when designing an adaptive procedure. First, one must decide when to change the stimulus level (the decision rule). The stimulus level may be changed after every trial, or when results match a predetermined pattern, or when performance deviates from its target by a critical amount. Second, one must decide to what level should the stimulus be changed for each trial (the step size). In other words, how large a "step" shall we take in the direction determined? We could use fixed steps or adjustable

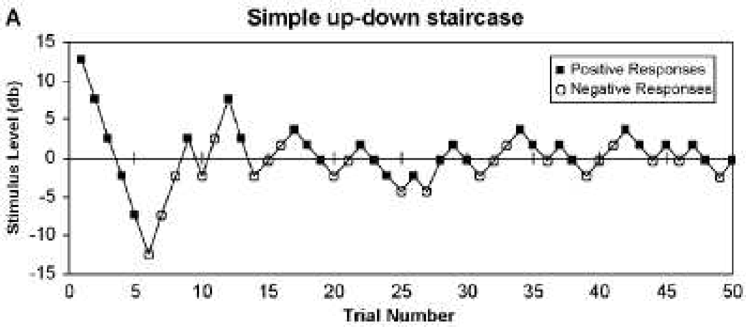


Figure II.9: An adaptive track following a simple up-down staircase procedure. Image from [Leek 2001].

steps. Third, one must specify when to end an experimental run (the stopping rule). The experimenter can decide to stop the run after a fixed number of trials, or fixed number of reversals, or other rules. After the design decisions are made, the actual experiment could be run and the psychometric function and the threshold could be measured.

The staircase procedure is one of the commonly used adaptive methods. For example, a simple up-down staircase reduces the stimulus level when the response is positive and increases the stimulus level when the response is negative as shown in Figure II.9 [Leek 2001]. Such an staircase procedure targets the 50% level for which the probability of a correct responce equals to the probability of an incorrect response. To target a higher performance level, the sequence for upward or downward movement maybe be two or more corresponding (positive or negative) responses. For instance, a two-down, one-up procedure targets at the 70.7% level on the psychometric function [Wetherill and Levitt 1965].

Regarding the methods of assessment, forced-choice methods are often used in studies of measuring psychometric functions. In forced-choice methods, the subject is presented a number of alternative choices in each trial in which the stimulus is presented. The subject is forced to choose one of the alternatives. For example, in a two-alternative forced choice (2AFC) procedure, the subject would have to choose one of the two alternatives for each trial.

CHAPTER III

RELATED WORK

Related work is discussed in this chapter. It can be characterized into three topics, work on inverse kinematics solvers, work on motion editing, more specifically motion signal processing techniques and motion generation frameworks, work on motion transitions methods, and work on empirical studies.

III.1 Inverse Kinematics

As mentioned previously, the inverse kinematics (IK) problem is to find the angles for each joint in an articulated model to achieve a goal position for an end effector. Inverse kinematics solvers can be roughly divided into two categories: numerical and analytic solvers. A numerical method is an iterative procedure to solve a numerical optimization problem. The numerical method is often a practical approach for a general-purpose inverse kinematics problem. An analytic method, on the other hand, is used when the problem is simple and because of its computational efficiency.

Inverse kinematics is an essential part of many motion editing techniques. Motion blending is often used to generate motion transitions, but blending may introduce artifacts into the resulting motion. One common scenario is that the character's feet move when they are supposed to remain planted, an artifact called foot-slide. To fix such problem, inverse kinematics is often used as a post-process.

Numerical methods have been used to solve an inverse kinematics problem of an articulated human figure. Rose et al. [1996] used this approach to handle spacetime constraints and inverse kinematics constraints. Their method for creating motion transitions uses a combination of these two constraints in order to generate seamless and dynamically plausible transitions. To enforce the inverse kinematics constraint, they developed an algorithm based on techniques presented in [Zhao and Badler 1994]. The algorithm kinematically

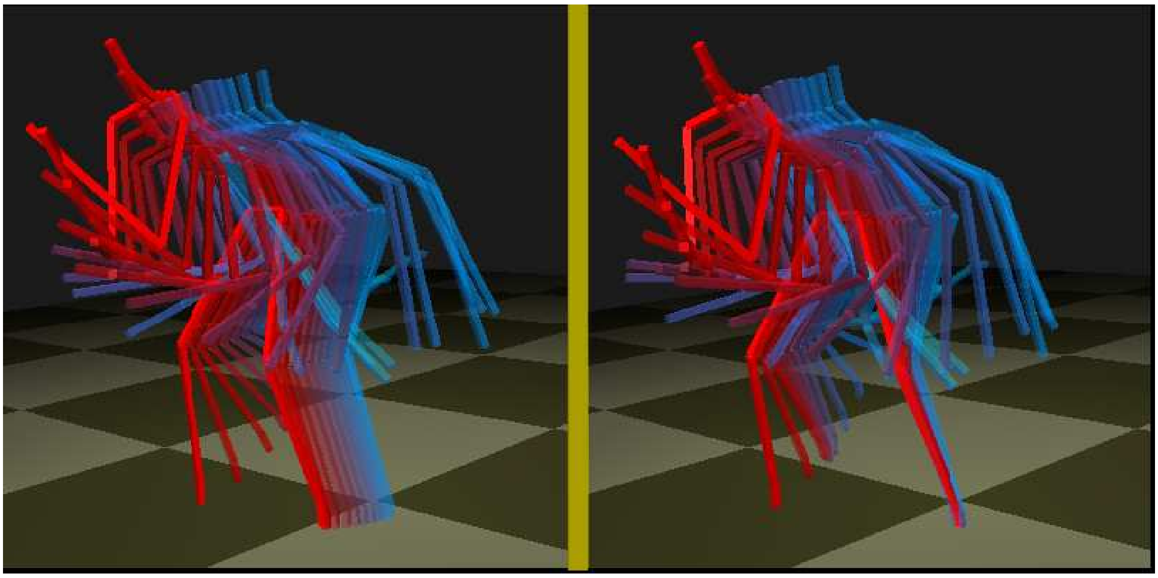


Figure III.1: Effect of inverse kinematics constraint on placement of feet. Image from [Rose et al. 1996].

controls the support limbs by optimizing for coefficients influencing a range of time. Figure III.1 shows the effect of inverse kinematics constraint on placement of feet in their work.

Lee and Shin [1999] presented an approach for interactive motion editing that combines a hierarchical curve fitting technique with inverse kinematics solver. They introduced the hierarchical motion representation for manipulating motion to satisfy constraints and for editing motion sequences. They introduced two inverse kinematics algorithms, a numerical approach for a general tree-structured figure and a faster specialized approach that combines the numerical techniques with an analytical method for a human-like figure with limb linkages.

Kovar et al. [2002b] presented an algorithm for removing foot-slide artifacts introduced by motion capture editing. They used an analytic IK algorithm for foot-slide constraints. They then added smooth adjustments to skeletal parameters when trying to satisfy constraints. They maintained continuity of the final motion by blending neighborhoods of constrained frames. Figure III.2 shows an example of their algorithm.

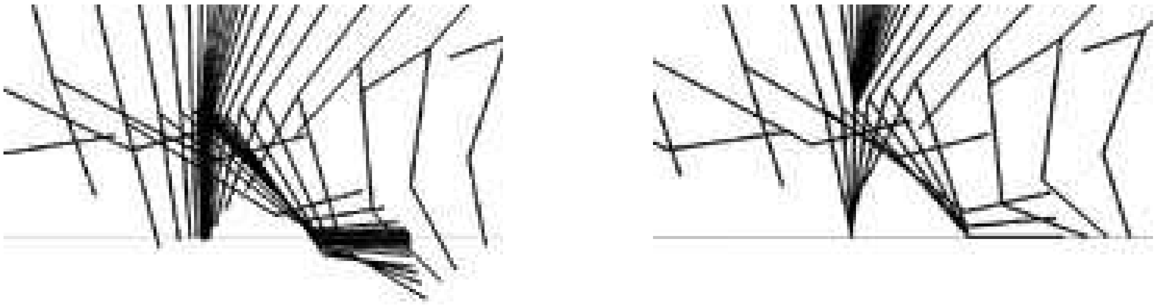


Figure III.2: An example of foot-slide and the result after applying inverse kinematics. The pictures show the location of the right foot over a portion of a walking motion. Image from [Kovar et al. 2002b].

Inverse kinematics has been used primarily to position limbs to maintain constraints in motion editing systems. In our work, linear blending was used to generate motion transitions. Inverse kinematics is then used as a post process to fix foot-slide introduced by linear blending. Many algorithms have been proposed in previous works to address these artifacts. We used the inverse kinematics solver provided by MotionBuilder 6.0 to constrain support limbs and correct foot-slide.

III.2 Motion Editing

Generalizing motion capture beyond simple playback requires the ability to edit the motion. For example, one could take a walking motion and stylize and alter it to create a new walking style. Or one could apply a sequence of captured motion to a character with different proportion. We will discuss research work in motion editing in this section.

III.2.1 Motion Signal Processing

The motion data of an articulated figure can be treated as signals that contain the values at each frame for each degree of freedom. Techniques from signal processing then can be adopted to provide ways to edit and modify animated motion.

Bruderlin and Williams [1995] discussed basic signal processing techniques applied to motion signal processing. They presented a method of multiresolution filtering and multitarget motion interpolation. They introduced several important editing techniques for motion interpolation, dynamic timewarping, waveshaping, and motion displacement mapping. Multiresolution filtering and interpolation used the knowledge that low frequencies contain general information of the signal and high frequencies contain detail and subtleties. Multiresolution filtering filtered bands of frequencies of the signal and multiresolution interpolation interpolates the signal at each frequency band. Timewarping plays a critical role in motion interpolation when speed of the motions to be blended varies. The timewarp procedure finds the optimal sample correspondence between the motion signals which implies a combination of stretching and compressing of the signals. We developed a timewarping method in our work based on the timewarping procedure discussed in this work.

Witkin and Popovic [1995] presented motion warping as a technique for editing motion. New motions can be created by joining captured motion clips and blending the overlap region. A timewarp curve is constructed to find the reasonable alignment of the corresponding points. Their motion warping technique fits well into the traditional keyframe animation. The timewarping technique discussed in this paper provided an alternative way for timewaping during motion blending. However, this technique requires manual intervention for selecting scale or shift factors.

Lee et al. [2002] introduced a method for applying filters to orientation data represented by unit quaternions. The orientation data is transformed to vector space and the filter mask is applied in the vector space. This method preserves coordinate-invariance, time-invariance, and symmetry properties of the orientation data. We are interested in exploring a distance metric of orientation data represented by linearized quaternions.

Signal processing techniques can be used to edit or modify animated motion data or higher level motion parameters. For example, blending of motion is essentially interpolation of joint angles and joint coordinates representing the movement. Signal processing techniques such as filtering and timewarping provide better control of the blending and can also be used to modify the motions to create better correspondences. In our work, linear blending was used to generate motion transitions and a timewarping algorithm was developed to find better correspondences (see Chapter V). Signal processing technique thus form the core of our methods.

III.2.2 Motion Synthesis

Motion capture research has concentrated on studying ways to editing and modifying existing motions to synthesize new motions. One of the earliest techniques for editing motion capture was Perlin's work [Perlin 1995]. They proposed a high level "textural" approach using modulated sine waves and stochastic noise to create responsively animated characters in real time. They aligned the natural cycles of walks and other rhythmic motions so that they can be blended together. In particular, they addressed the importance of transition durations by saying, "*It is surprising how much expressiveness one can achieve by tuning these transition times*". This thought is a key motivation of the present work.

Lamouret and van de Panne [1996] discussed motion synthesis by example. They presented the idea that new animations can be created by re-sequencing the example motion segments. Best-fit motion primitives are selected and tailored for the desired motion. The fitness of the sample motion for the desired motion is measured by the distance metric that considered both the continuity of the motion as well as its suitability for the terrain. They validated their technique on physically based animation. This work provided an insight into constructing motion generation frameworks by re-sequencing.

Wiley and Hahn [1997] focused on real-time interpolation synthesis of motion based on motion capture data. Interpolation synthesis is limited to small numbers of parameters and is specified by a value that occurs over the entire motion or at a specific point in time. They pointed out that the combination of motion capture and interpolation synthesis has the potential to animate the characters needed in virtual reality environments.

Rose et al. [1998] presented a technique called "verbs and adverbs" for creating believable human motions. Verbs describe motions and adverbs describe emotional expressive-

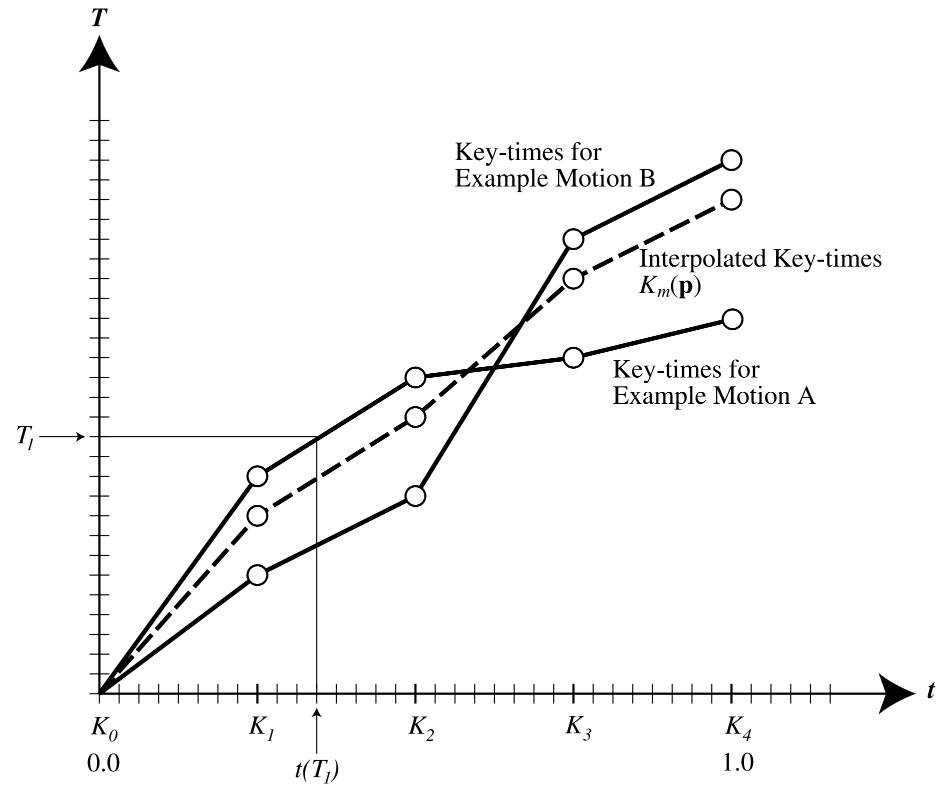


Figure III.3: Mapping between generic time and keytimes. Image from [Rose et al. 1998].

ness or control behaviors. Verbs can be combined with seamless transitions to construct verb graph. Adverbs provide interactive control over the virtual character's action and style. Radial basis functions are used to interpolate between motions. Timewarping is used to parameterize the motion to generic time as shown in Figure III.3. Motion examples are interpolated within the generic timeframe with an ease-in ease-out blending function. Transitions between two verbs are created by blending with a monotonically decreasing blending function. The duration of the transition is calculated by taking the average of the lengths of the two blending regions. Park et al. [2002] described a timewarping scheme similar to Rose et al. [1998] for aligning motion clips of different speeds. Both these techniques required manual construction of transition intervals. We are interested in developing timewarping methods similar to these methods but in an automatic way.

Pullen and Bregler [2000] described a method for motion synthesis by manipulating motion capture data. They started from motion capture data and synthesize the data by breaking it into frequency bands. They synthesized important joint angles and translation data. They applied their technique on a wallaby hopping motion data and used the synthesized data to animate wallaby-like characters.

Brand and Hertzmann [2000] introduced a framework called a "style machine", a probabilistic model for synthesizing motion capture data in interpolation or extrapolation of styles and using cross-entropy optimization to learn motion patterns. They applied style machines to synthesize new style of the existing motion. New choreography was generated by applying style machines on modern dance. This probabilistic method is powerful but may eliminate subtleties of the motion during synthesis that give the motion a sense of richness. Our work focuses on creating motion based on example motions while preserving the the subtleties of the original motion.

Popovic et al. [2000] described an interactive method for manipulation of rigid multibody dynamic simulations. They implemented their system by first constructing a control module, secondly building a rigid body simulator and finally developing a user interface for motion editing. An animator can manipulate the position and velocity of the object directly through the interface so that the he can guide the system to the appropriate solution. The relevance of this work is that transition durations in this work can be computed based on the dynamics of the motions whereas transition durations in our work are computed based on the correspondences of the motions.

Gleicher [2001] compared a range of methods for constraint-based editing of motion capture data. He compared different approaches for motion editing at performance, range of constraints handled, order independence, and other aspects.

Galata et al. [2001] presented an approach of learning structured behavior models using variable length Markov models. The experimental data was obtained by tracking using a simple contour tracker of individuals performing exercise routines. Their goal was to develop models with recognition and generative capabilities. They used variable length

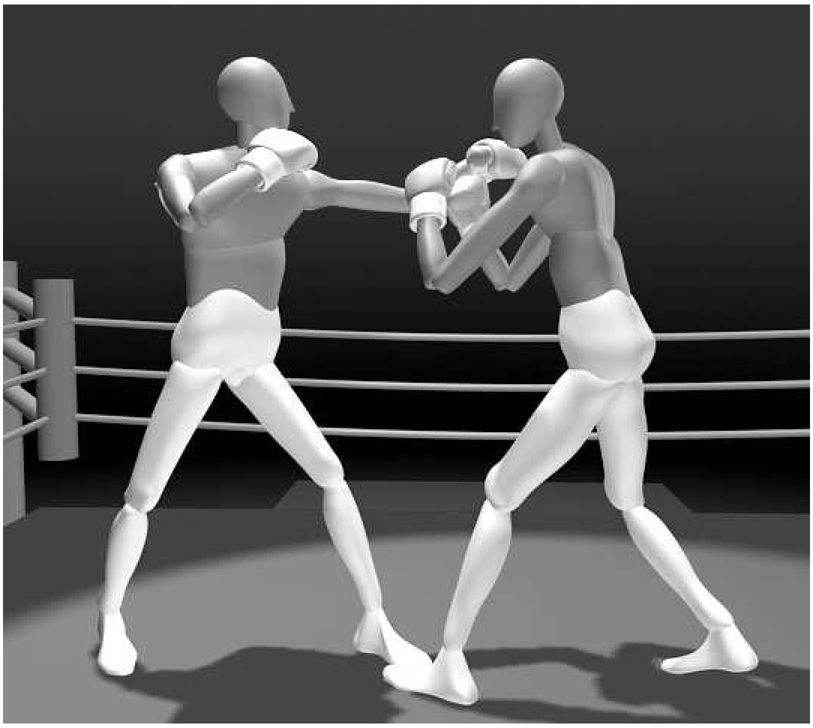


Figure III.4: Motion capture-driven simulations. Image from [Zordan and Hodgins 2002].

Markov models for learning complex behavioral dependencies and constraints at a lower level and temporal dependencies at a higher level. This work can be applied to motion editing techniques since it used Markov models to learn complex behaviors.

Zordan and Hodgins [2002] described a framework for producing dynamic character animation by combining dynamic simulation and human motion capture data. They aimed at interactive motions between characters, such as collisions and external force. They used a dynamic motion controller to control the acceleration of the character based on the physics of the character. They introduced a motion transition method that incorporates constraint information to generate continuous motion. Figure III.4 shows an example of a boxing motion generated using their approach. We would like to incorporate our methods of generating new motion into dynamical simulation systems that produce long streams of animation like the way they presented in this paper.

Dontcheva et al. [2003] introduced a system that allows the animator to mimic aspects of the desired motion and creates and edits the character animation in real-time. They used widgets to prove connections between the actor and the character. The animation was created layer by layer where each layer concentrates on editing different aspect of the animation.

Zordan et al. [2005] presented a technique for synthesizing motion capture sequences and incorporating physics-based response for transitions. They first find a transition-to motion capture sequence by comparing simulated data with sequences in motion capture library. Secondly, they used a joint controller to simulate a transition and interpolate between them to create the final motion. The controller uses an *inertia-scaled* PD-servo at each joint to computer torques. They also manipulated the delay time during the transition to make the character's action appear realistic.

Manifold Learning

Manifold learning of high dimensional motion data has gained attention recently in the graphics community. It has been shown that most dynamic human behaviors are intrinsically low dimensional, for example, legs and arms operate in a coordinated way. Bowden [2000] discussed modeling motion capture data by reducing dimensionality and constructing a Markov chain. Motion data were captured by retro-reflective IR markers. They used Principal Component Analysis (PCA) [Jolliffe 1986] on a training set such that a low number of key frames can reproduce the motion. They modeled temporal constraints as Markov chain and constructed discrete probability density functions indicating transition probabilities.

Safonova et al. [2004] described a method that utilized the low dimensional representation of motion. They used PCA to represent motion as six to ten dimensional basis vectors. Optimization is then used to find a motion that minimizes some objective function relating these vectors and satisfies the user-specified physics constraints.

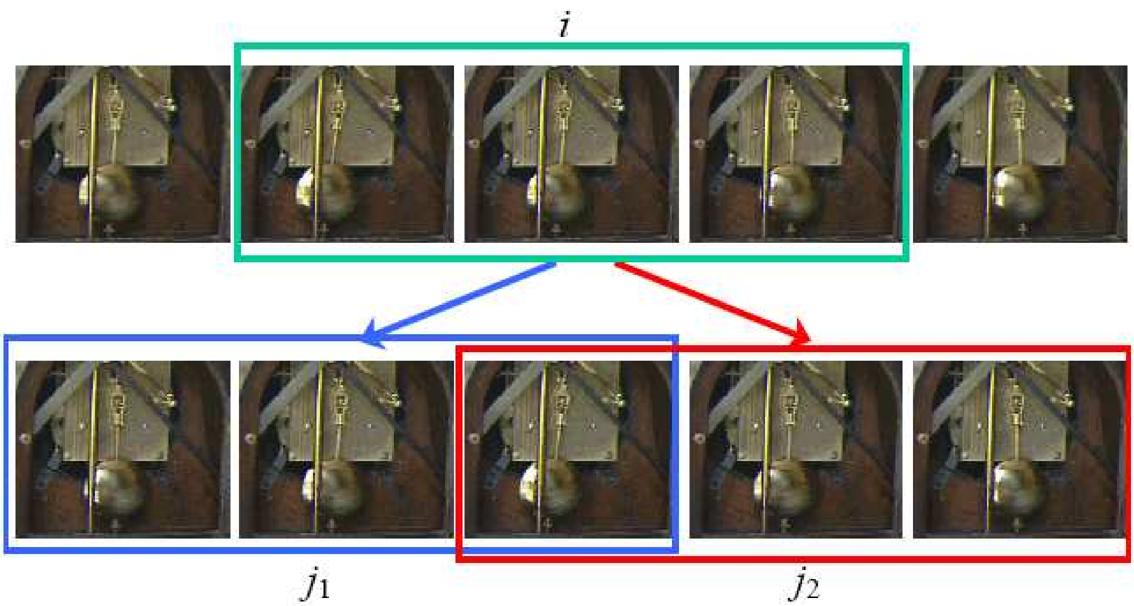
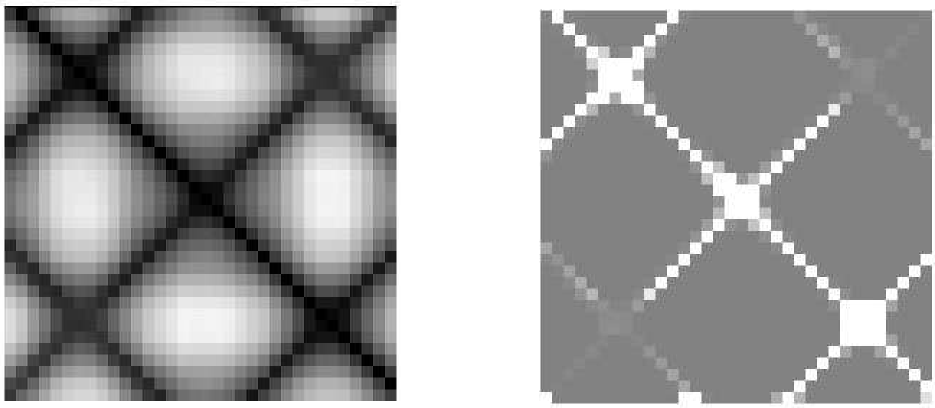


Figure III.5: Finding good transitions in a pendulum sequence. Frame i in the top row matches both frames j_1 and j_2 of the bottom row very closely. However, of these two possibilities, only frame j_2 comes from a sequence with the correct dynamics. Image from [Schödl et al. 2000].

Motion Graphs

Recently, many researchers have drawn inspiration from the work of Schödl et al. [2000] on video textures to retain the original motion sequences but play them back in non-repetitive streams. The video texture is a stream of video images which is synthesized from a repertoire of images by assembling and blending the source images. Finding the transition points in the video sequences and smoothing visual discontinuities at the transitions are two important factors in this framework. They compute the L_2 distance as the measure of similarity between all pairs of frames in the input sequence and the probability of transitioning between frames in the sequence. Figure III.5 shows an example of finding good transitions in a pendulum sequence. Figure III.6 shows the distance matrix and transition probabilities for the clock pendulum sequence.

Lee et al. [2002] modeled motion as a two layer structure. The lower layer is a first-order Markov process that generates motion sequences by creating transitions among



Distance Matrix　　Transition Probabilities

Figure III.6: Distance matrix and transition probabilities for the clock pendulum sequence. Image from [Schödl et al. 2000].

motion clips, based upon probabilities of transitioning. The higher layer is a cluster forest that captures the connection between motion frames by constructing cluster trees and the similarities in motion frames by clustering. They explored three types of interfaces, the choice-based interface, the sketch-based interface and the vision-based interface. They applied their system on four different examples: maze, terrain, playground and step stool. When creating transitions in the lower layer, they used a blend interval ranging from 1 to 2 seconds, depending on the example. To fix constraint violations such as foot-slide, they used the hierarchical motion fitting algorithm presented by Lee and Shin [1999]. The probability of transitioning is calculated based on the difference between motion frames. The distance function reflects the weighted differences of joint angles and joint velocities. The weights on joints are set to one for shoulders, elbows, hips, knees, pelvis and spine. Weights are set to zero for joints at the neck, ankle, toes and wrists. Our work evaluates the cost function for determining transition points proposed by this paper and proposes a new set of weights on joints.

Another motion editing framework was presented by Kovar et al. [2002a] for generating motions by synthesizing streams of captured motion and creating transitions automatically. Figure III.7 shows an example of motion created using their algorithm. After the motion graph is constructed, the motion that satisfies the user's need can be extracted from the graph. They apply their approach to cyclic locomotion. They generate transitions by first calculating the similarities metric for motions, secondly selecting transition points which are local minima with small error values, and finally creating transitions using SLERP.

The similarity metric as shown in Figure III.8 is calculated based on the distance between two point clouds generated by sampling the mesh of the animated character. The point clouds are generated over of window of n frames, typically 10 (corresponding to one-third of a second), and consist of the point clouds of the sample mesh at each frame in the window. The distance between two frames is computed as a weighted sum of squared dis-

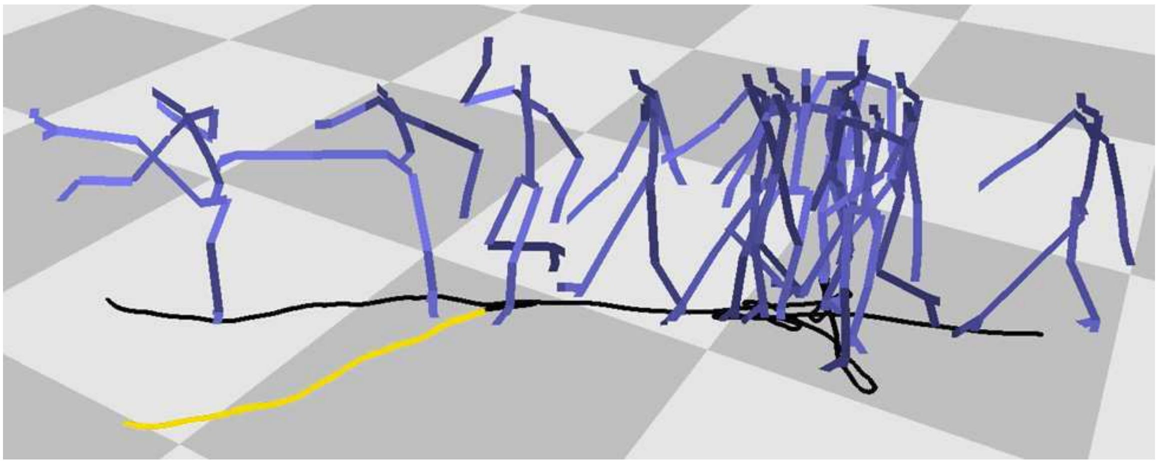


Figure III.7: Motion generated using the search algorithm presented in the paper. Image from [Kovar et al. 2002a].

tances between corresponding points in the point clouds. The main difference between the cost metrics used in our work and this work is that, the cost metric for transitioning used in our work is joint based, where the cost metric used in this work is model based. There are several complicated factors involved in evaluating such a model based metric. First, we need to determine how many points to sample in total and how many points to sample for each body part. Second, the metric potentially includes a weight on each individual sample point of the mesh at each frame. Kovar et al. do not report what their weighting scheme was. A tractable approach is to assign uniform weights to each body part, possibly tapering them depending on position within the window. Based on this cost metric, in their recent work they developed automated methods for identifying and extracting logically similar motions and using them to build a continuous parameterized space of motions by applying blending techniques [Kovar et al. 2004].

Similarly, Arikan and Forsyth [2002] presented a framework for motion synthesis of motion capture data meeting a wide variety of constraints. A motion graph is first constructed by collecting existing motion sequences and creating edges between motion frames and labeling cost for edges. New motion can be generated by a randomized search of the motion graph. Instead of creating transitions between the discontinuity of the motions, they

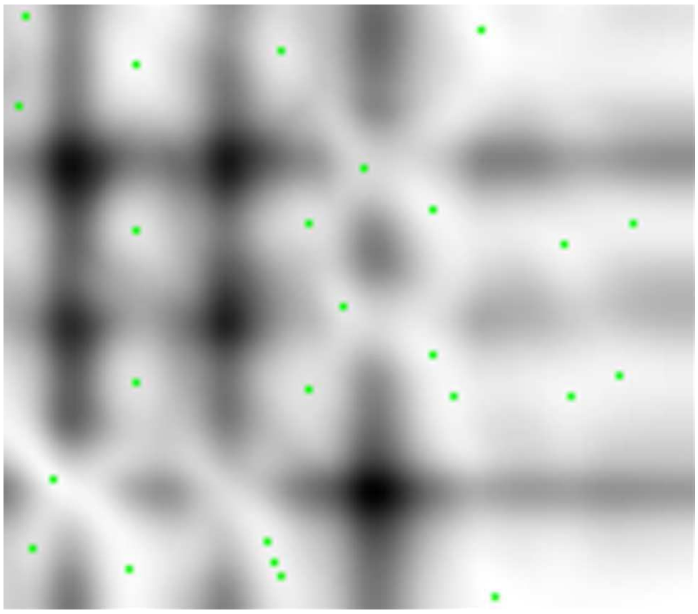


Figure III.8: An example error function for two motions. The entry at (i, j) contains the error for making a transition from frame i of the first motion to frame j of the second. White values correspond to lower errors and black values to higher errors. The colored dots represent local minima. Image from [Kovar et al. 2002a].

used localized smoothing and adding the result back to the signal. They chose the smoothing domain to be 0.1 second. Smoothing provided an alternative way for concatenating between motion segments, but it can cause artifacts like foot-slide on the ground.

Sidenbladh et al. [2002] discussed modeling human motion for synthesis. They modeled high dimensional motion data as an implicit empirical distribution. They structured the motion database as a binary tree and used a probabilistic search method to stochastically generate sample motions. The search algorithm exploited the coefficients of the low-dimensional representation of motion data.

Li et al. [2002] described a motion texture as a statistical model for synthesizing and editing motion capture data. They synthesized new motion based on motion textons and their distributions. Motion textons are represented by linear dynamic system to capture the dynamic nature of the motion. They synthesized the motion textons by sampling noise. They developed transition matrices to define the likelihood of transitioning between motion

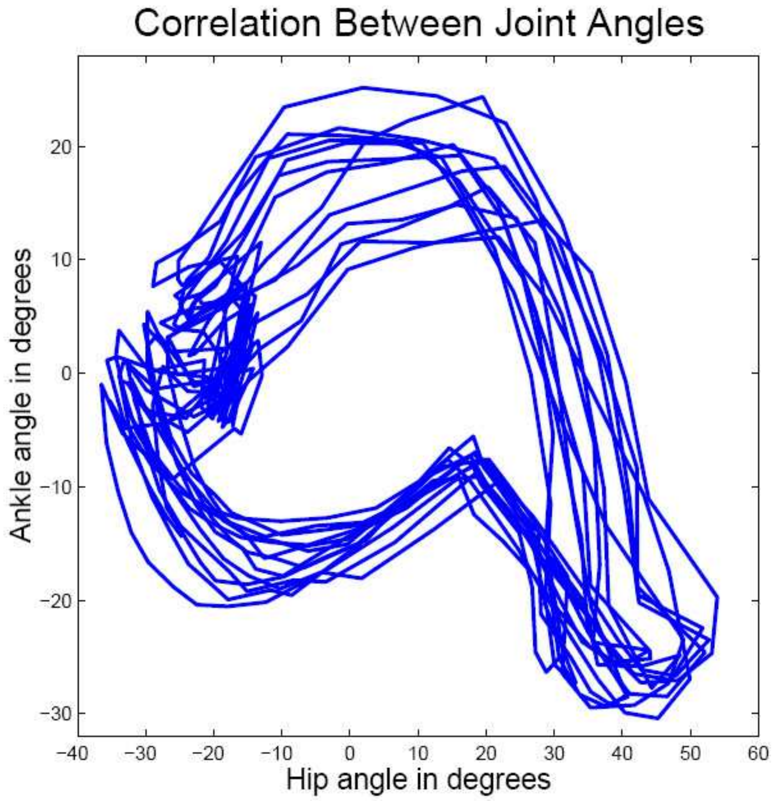


Figure III.9: Correlation between joint angles. Shown is the ankle angle versus the hip angle for human walking data. The fact that this plot has a definite form demonstrates that the angles are related to each other. Image from [Pullen and Bregler 2002].

textons.

Pullen and Bregler [2002] described a method for creating new motion by using both keyframes and motion capture data. Their framework allows the animator to keyframe the rough motion. They showed that joint motions are often correlated as shown in Figure III.9 so that a fraction of the DOFs can be used to sketch an animation. They used the information in motion capture data to add details to the desired motion. Note that in this work they found a duration of 0.2s to 0.8s for smoothing to work well, based on user selection and depending on the particular motions, whereas we focus on developing methods to automatically determine the optimal transition durations in our work.

Arikan and Forsyth [2003] presented an interactive motion synthesis framework. The motion database is annotated by describing actions before synthesis. A small part of the motion database is annotated by the user first, Support Vector Machine (SVM) classifiers are used to generalize the annotation for the whole database. The motion that meets the user's specification is assembled from the motion database. The motion synthesis algorithm for the system is based on dynamic programming optimizations.

Reitsma and Pollard [2004] described an algorithm for assessing motion graph for its utility in navigation tasks. One problem with motion graph data structure is that the quality of result may not be easy to predict from the input motion segments. They defined metrics for evaluating motion path quality and reaching ability of motions and presented an embedding algorithm for capturing all possible paths. This work showed that evaluation techniques can provide insight into character capabilities captured in a motion graph structure.

III.3 Motion Interpolation and Transition

Transitions are an essential component of motion editing systems. Most of the previous work emphasized selecting appropriate transition points while creating new motions rather than on the durations of the transition. While it is true that transitions are less of a problem if the motions are similar, visual artifacts can still appear if the duration is too short or too long. One of our goals in this research is to determine optimal transition duration for creating motion transitions.

Blending is a basic way to create motion transitions. If incorrectly applied, simple blending can produce undesirable results for cases where the motions are not properly aligned. In contrast, Rose et al. [1996] used dynamic simulation to generate transitions, using a combination of spacetime constraints and inverse kinematics constraints to create dynamically plausible transitions. Figure III.10 shows an example of transitions created using this technique. Motion data are spliced and mixed to create new motions. Spacetime constraints are formulated by minimizing joint torque over time. The time for transitioning usually ranges from 0.3 to 0.6 seconds. Spacetime transitions are more computationally

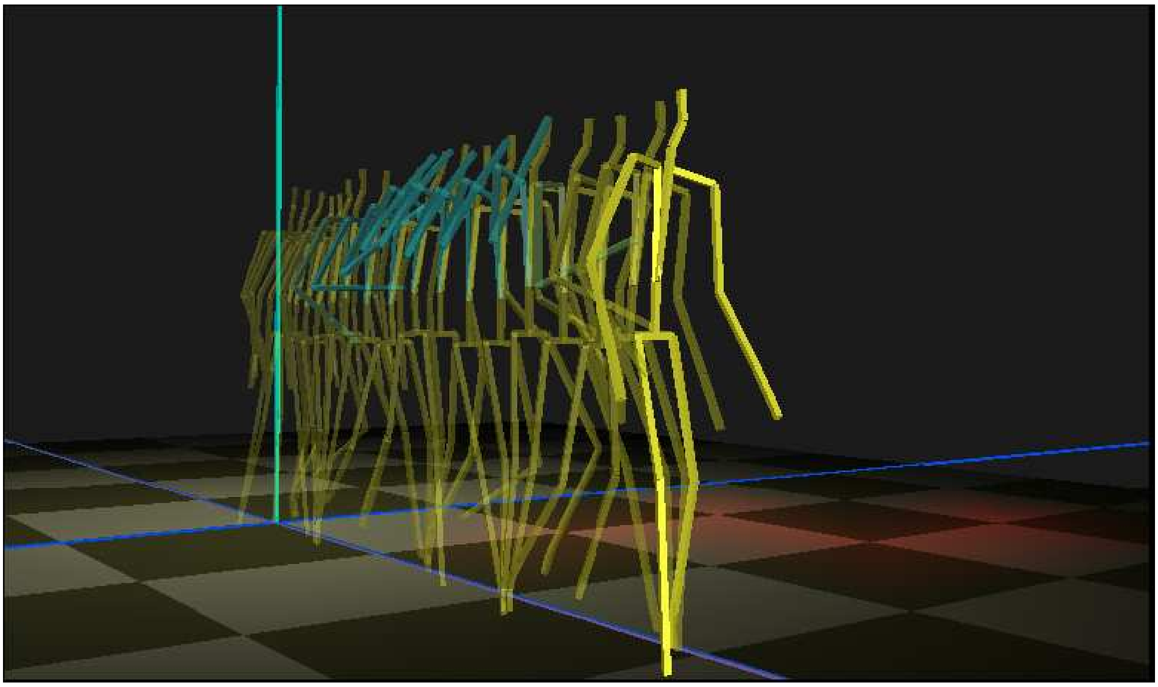


Figure III.10: Walk motion transitioning to salute motion and back to walk motion. Arm degrees of freedom affected by the transition are colored green. Image from [Rose et al. 1996]

expensive than joint angle interpolation techniques. This paper proposed a range of the transition duration, but left the exact specification of the duration to the operator.

Mizuguchi and Calvert [2001] presented a framework for data driven transitions. The objective for designing this framework is to ease the communication between the animator and the programmer. They developed an Application Programming Interface (API) which allowed the programmer to access and apply the transition data. An interactive editor was also developed to allow the animator to define the transitions. They discussed the role of blend length for creating transitions by blending. Their impression is that 10 frames is a good starting point and 20 frames is generally good enough. Also, by changing the target frame of a transition, the correspondence of motions to be blended are changed as shown in Figure III.11. They also conducted an informal user evaluation for the framework and had encouraging feedbacks for the potential of pursuing such a system. We develop ways to determine optimal blend length and compare them with constant blend lengths proposed

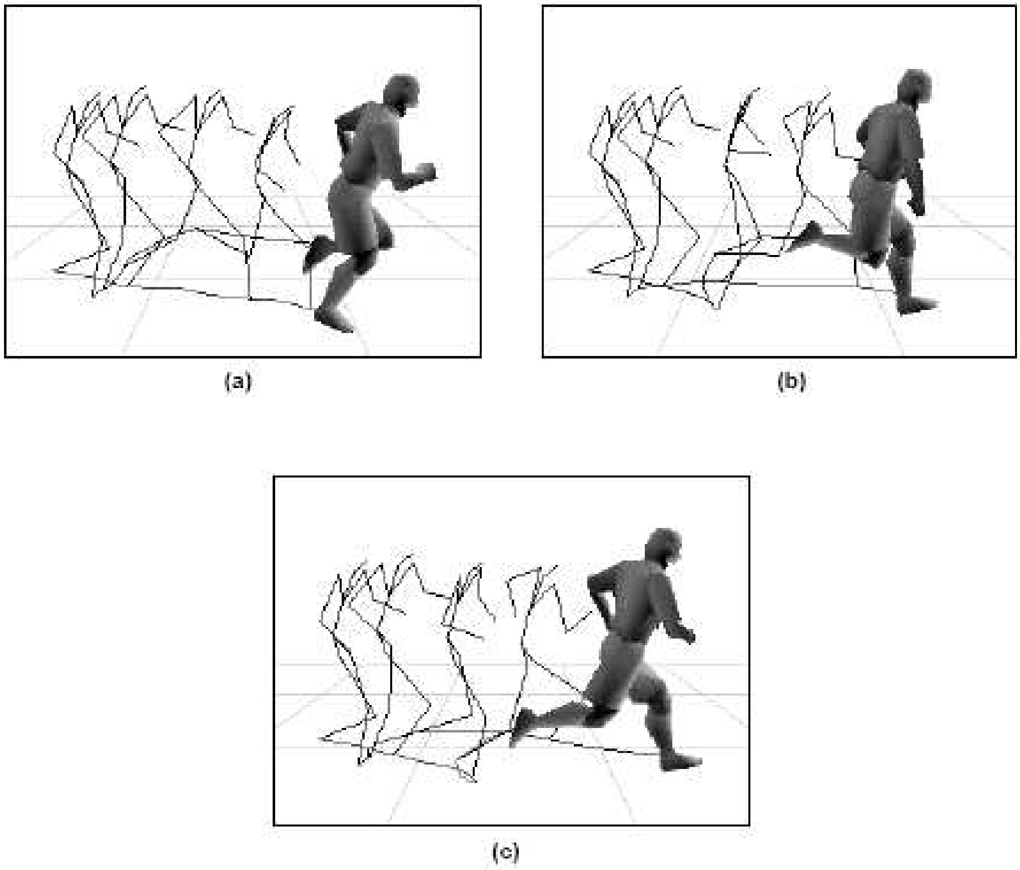


Figure III.11: Three transitions of jog to run with only target frame varied between them. (a) Good - There is a good correspondence causing the feet to strike the ground properly and the legs to move in a proper cycle, (b) Poor - The target frame has a different support leg causing a hop, (c) Moderate - The correspondence is improved over (b) but the right foot loops at one of the contact points. Image from [Mizuguchi et al. 2001]

by this work.

Kovar and Gleicher [2003] introduced registration curves for motion blending. Registration curves determine relationships of the input motions automatically. A registration curve is built by creating a timewarp curve, creating an alignment curve and identifying constraint matches. They discussed the application of the registration curve for creating transitions, motion interpolation and continuous motion control. The distance function for calculating the frame correspondence is based on the position of two point clouds, which are windows of 5 frames of neighborhoods. The timewarp curve is created by fitting a strictly increasing function to the frame correspondences. The algorithm for creating transitions takes as input two frames and the width of the transition.

There is little work in computer animation that directly evaluates the visual appeal and physical correctness of synthesized human motion. Natural human motion can be obtained by motion capture technique. However, motion editing, especially creating transitions between segments of captured motion, often introduce artifacts such as hops and foot-slide into the resulting motion. Several recent works address these issues.

Ren et al. [2005] developed an approach to quantifying the naturalness of human motion. They trained classifiers to distinguish between natural and unnatural movement based on human-labeled, ground-truth data. The training motions are composed of a repertoire of captured natural human motions including locomotions and other behaviors. The testing motions contains both natural and unnatural motions obtained from motion editing, keyframed motions, motions with intentionally added noise, or insufficiently cleaned motion capture data. They explored the performance of three motion learning techniques, naive Bayes, hidden Markov models (HMM), and switching linear dynamic systems (SLDS). They implemented a simple marginal histogram probability density estimator based on the naive Bayes models the baseline to compare to these three motion learning techniques. Receiver operating characteristic (ROC) curves were utilized to present and compare the result. They found that, for all of their learning methods except SLDS, the most difficult unnatural motions to detect were the bad motion graph transitions.

Interpolation of motion capture data has shown to be able to generate visually pleasing motion. Safonova and Hodgins [2005] analyzed the motions produced by interpolation for physical correctness in terms of a number of basic physical properties such as linear and angular momentum during flight phase, foot contact, static balance and friction with the ground during stance, and continuity between flight and stance phases. They suggest small modifications to the standard interpolation technique such as interpolating using the position of the center of mass instead of using root positions during a flight. These modifications in some circumstances will improve the visual quality of interpolated motions while guaranteeing their physical correctness. For example, while interpolating a forward jump with no turn and forward jump with 360 degree turn, they found that the interpolation of the root results in an unnatural motion during the flight phase but a natural looking motion if the center of mass is interpolated.

III.4 Empirical Evaluation

Empirical evaluation has recently gained popularity in the graphics community, although there is considerable work on point light experiments in the psychophysics literature, e.g., [Johansson 1973]. He was concerned about visual perception of biological motion. The motion of the living body was represented by a few bright spots describing the motions of the main joints. These elements are abstract mathematical points as carriers of the motion. The results of their studies showed that the proximal motion patterns presented carried all the essential information needed for immediate visual identification of motions such as human walking, running, dancing, etc. The model for visual vector analysis is applied in their study for motion and space perception. The result of the vector analysis showed that subtracting or adding common components to the element motions do not have a disturbing effect on the identification of the motion. The relevance of these experiments is that users can discriminate between subtle effects given a point-light representation of human motion and absent other visual cues.

Hodgins et al. [1998] conducted studies on the perception of human motion with polygonal model and stick figure model as shown in Figure III.12. They used A/B comparison tests. They conducted experiments on three types of motion variations, torso rotation, dynamic arm motion and additive noise. The motion used in the study were generated by making kinematic modifications to dynamically simulated motions. They used the responses from the subjects to compute the sensitivity measure. The study shows that the subjects were more sensitive to changes of the motion when rendering with polygonal model.

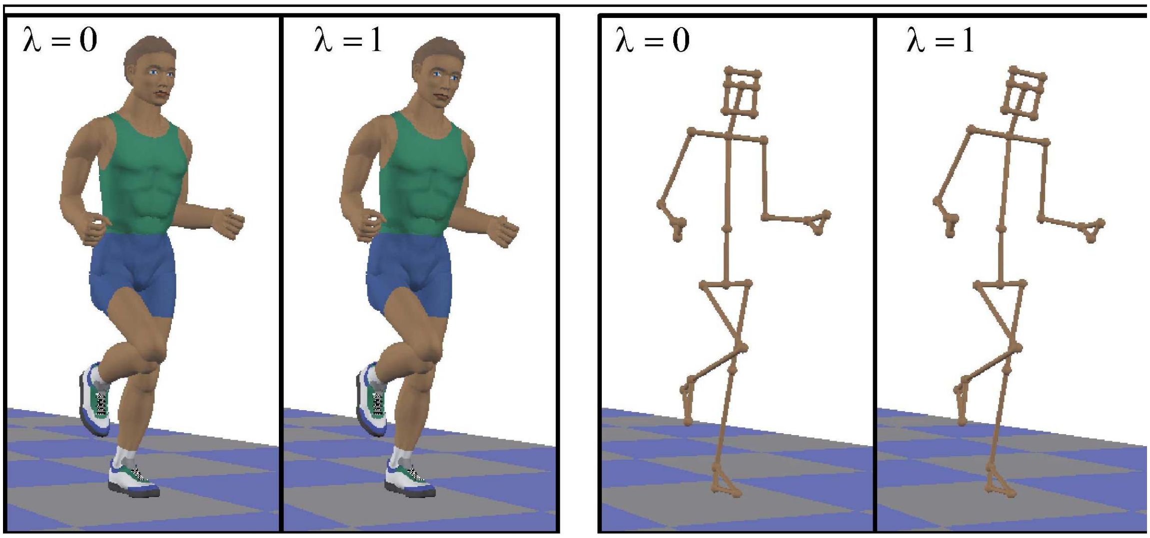


Figure III.12: Images of animated human runner. (a) Two running motions rendered using a polygonal model. (b) The same pair of motions are rendered with a stick figure model. Image from [Hodgins et al. 1998]

This work conducted user studies to explore the perception of human motion with different models, our work conducted user studies to compare different methods for creating motion transitions.

Bodenheimer et al. [1999] conducted studies to determine the "naturalness" of motion. They introduced natural-looking variability by constructing noise function into cyclic animations of human motion. The experiment is designed to vary the amount of noise in the simulation and ask the subject to decide what level of noise appeared most "natural". The experiment showed that for the male running simulation most users prefer some noise over no noise or higher amounts of noise.

Oesker et al. [2000] explored the effect of level of details in naturalistic character animations. They conducted experiments to test the influence of articulated body animations on the judgment of human observers of the level of overall skill exhibited by four simulated soccer teams. Four levels of details were added to a two teams playing soccer motion and the observers failed to notice the difference. This work performed a study methodologically similar to our own but trying to assess the effects of level of details in animation.

O'Sullivan et al. [2003] discussed the visual fidelity of animations by proposing a metric for measurement. They conducted a set of psychophysical experiments to establish acceptance thresholds for visual sensitivity. They used randomly interleaved staircase designs in the study. They examined the effects of angular distortions, momentum distortions and spatial-temporal distortion. They investigated two case studies, simulation levels of detail and constrained dynamics. This work is similar to our work in the aspect of conducting experiments to determine thresholds for visual sensitivity.

Reitsma and Pollard [2003] presented results of a study of user sensitivity to errors in ballistic motion. Errors in the motion were entered by twisting the translational velocity of the center of the mass. They proposed a perceptual metric based on the results of the user study. The results from the user study also showed that subjects are more prone to detect errors in horizontal velocity than in vertical velocity; subjects are more prone to detect added accelerations than decelerations. This work conducted user studies on perception of errors in ballistic motion and our work conducted user studies on perception of difference using different blend length for generating motion transitions.

The visual perception of length change in animation was investigated in [Harrison et al. 2004]. They conducted studies to examine to what extent the lengths of the links in an animated articulated-figure can be changed without the viewer being aware of the change. In terms of the design of the user study, they used a two alternative forced choice paradigm and a staircase method to dynamically vary the change in length following each observer's response. All the detection thresholds are presented in terms of relative changes of length. The results of the experiments provided guidelines for obscuring length changes during animation, for example, length changes should never be greater than 20%, even if the viewer is not attending to the figure, and changes are more difficult to perceive during fast motions.

The above experiments explored various aspects of visual perception of an animation. We are interested in exploring the effects of physical parameters on the visual perception of motion transitions. It is relatively easy to make high quality motion using motion capture, but generating visually pleasing transitions among those motions is still difficult and involves significant manual labor. An animator often needs to manually tune the parameters such as the transition interval. Many methods of generating motion transitions have been proposed. Each method has its own strengths and weaknesses. Evaluation of these methods regarding the subjective qualities such as naturalness is essential. In the following chapters, we will present several methods for generating motion transitions and empirical studies evaluating them.

CHAPTER IV

THE COST FUNCTION: OPTIMIZING THE WEIGHTS

High quality motion can be obtained by motion capture. However, once a sequence of motion is captured, it is hard to modify and reuse. Much recent research has focused on creating new motion by re-ordering the original motion [Lee et al. 2002; Kovar et al. 2002a; Arikan and Forsyth 2002]. These authors developed re-sequencing systems for libraries of motion capture data. Motion transition plays an important role in such systems and in video games. Determining an optimal transition point between motion clips and generating a seamless transition are two critical components to the visual appearance of the resulting motion. Transition points are the points at which the motion will change from one segment of captured motion to another segment, either within the same motion or another motion. Because these transition points represent discontinuities in the motion stream, selection of good transition points can be crucial to the quality of the resulting motion.

Each of the works cited above uses a different distance function to calculate the cost of transitioning from one frame to another. These distance functions are all parameterized through user-selected weights. In this phase of the work, we evaluate the cost function described by Lee et al.[2002] for determining good transition points. Lee et al. proposed an original set of weights for their metric. We compute a set of optimal weights for the cost function using a constrained least-squares technique.

The weights are then evaluated in two ways: first, through a cross-validation study and second, through a medium-scale user study. The cross-validation shows that the optimized weights are robust and work for a wide variety of behaviors. The user study demonstrates that the optimized weights select more appealing transition points than the original weights.

IV.1 The Cost Function

In this section, the specifics of the cost function are reviewed. Given two sequences of motion, indexed by their frame number, we construct a probability matrix. The (i, j)th element of the probability matrix is the probability of transitioning from frame i to frame j. The probabilities are constructed from a measure of similarity between the frames using an exponential function and given by

$$P_{ij} \propto exp(-D_{j,j-1}/\sigma) \tag{IV.1}$$

where $D_{i,j-1}$ represents the cost for transitioning between frame i and frame $j-1$, and σ controls the mapping between the cost measure and the probability of transition.

The cost for transitioning between frame i to frame j is given by

$$D_{ij} = d(p_i, p_j) + d(v_i, v_j) \tag{IV.2}$$

where $d(v_i, v_j)$ is the weighted distance of joint velocities, and $d(p_i, p_j)$ is the weighted difference of joint orientations. This term is given by

$$\begin{aligned} d(p_i, p_j) &= \left\|p_{i,0} - p_{j,0}\right\|^2 + \sum_{k=1}^{m} w_k \left\|\log\left(q_{j,k}^{-1} q_{i,k}\right)\right\|^2 && \text{(IV.3)} \\ &= \left\|p_{i,0} - p_{j,0}\right\|^2 + \sum_{k=1}^{m} w_k |\theta_{(i,j),k}|^2 && \text{(IV.4)} \end{aligned}$$

where $p_{i,0}, p_{j,0} \in \mathbf{R}^3$ are the global translational positions of the figure at frames i and j, respectively (zero denotes the root joint); m is the number of joints in the figure; and $q_{i,k}, q_{j,k}$ are the orientations of joint k at frames i and j, respectively, expressed as quaternions. The log-norm term represents the geodesic norm in quaternion space, which equals to $\theta_{ij,k}$. $\theta_{ij,k}$ is the angle of rotation that transforms the orientation $q_{i,k}$ to the orientation $q_{j,k}$ (see explanation in Chapter 2). Each term is weighted by w_k.

For this work, our skeleton consisted of 16 joints and the complete figure had 54 degrees of freedom. Each joint was a three degree of freedom joint, and there were degrees of freedom for global position and orientation. All motion capture data was sampled at 30 frames per second.

IV.2 Optimizing the Weights

The cost function contains parameters to modify the transition cost. The parameters take the form of weights. The cost function weights both the geodesic norm between joint orientations and the joint velocities, and contains another parameter, ν, trading off the velocity and position distances. Lee et al. report setting the weights to one for the shoulders, elbows, hips, knees, pelvis, and spine; others are set to zero. No value for ν is given.

We would like to use motion capture to determine optimal values for the weights. We will contrast motions using optimized weights versus the weights Lee et al. report. We will refer to the sets of weights used by Lee et al. as the original weights. An example of the cost function for transitions from one motion to another for the original weights is shown in Figure IV.1 with $\nu = 1$. The figure is normalized so that an intensity of zero corresponds to the minimal value of the cost function for that motion and an intensity of 255 corresponds to the maximum cost. The minimum and maximum before normalization are 0.0993 and 20.0677. The cost function is reasonably uniformly distributed over its ranges, so the linear normalization gives an accurate picture of the function's variation. In the figure, the vertical axis represents one motion and the horizontal axis represents the other motion. A pixel i, j in the figure corresponds to the cost of transitioning from frame i in the first motion to frame j in the second motion. Darker values correspond to lower costs for transition.

To optimize the weights, we took a set of 16 different segments of captured motion, each several seconds long. These segments consisted of a variety of motions including walking in different styles, running in different styles, jogging, dancing, and gesturing. For these segments we manually selected 16 good transitions and 26 bad transitions. A good transition was one in which the visual discontinuity of the transition was minimal; a bad

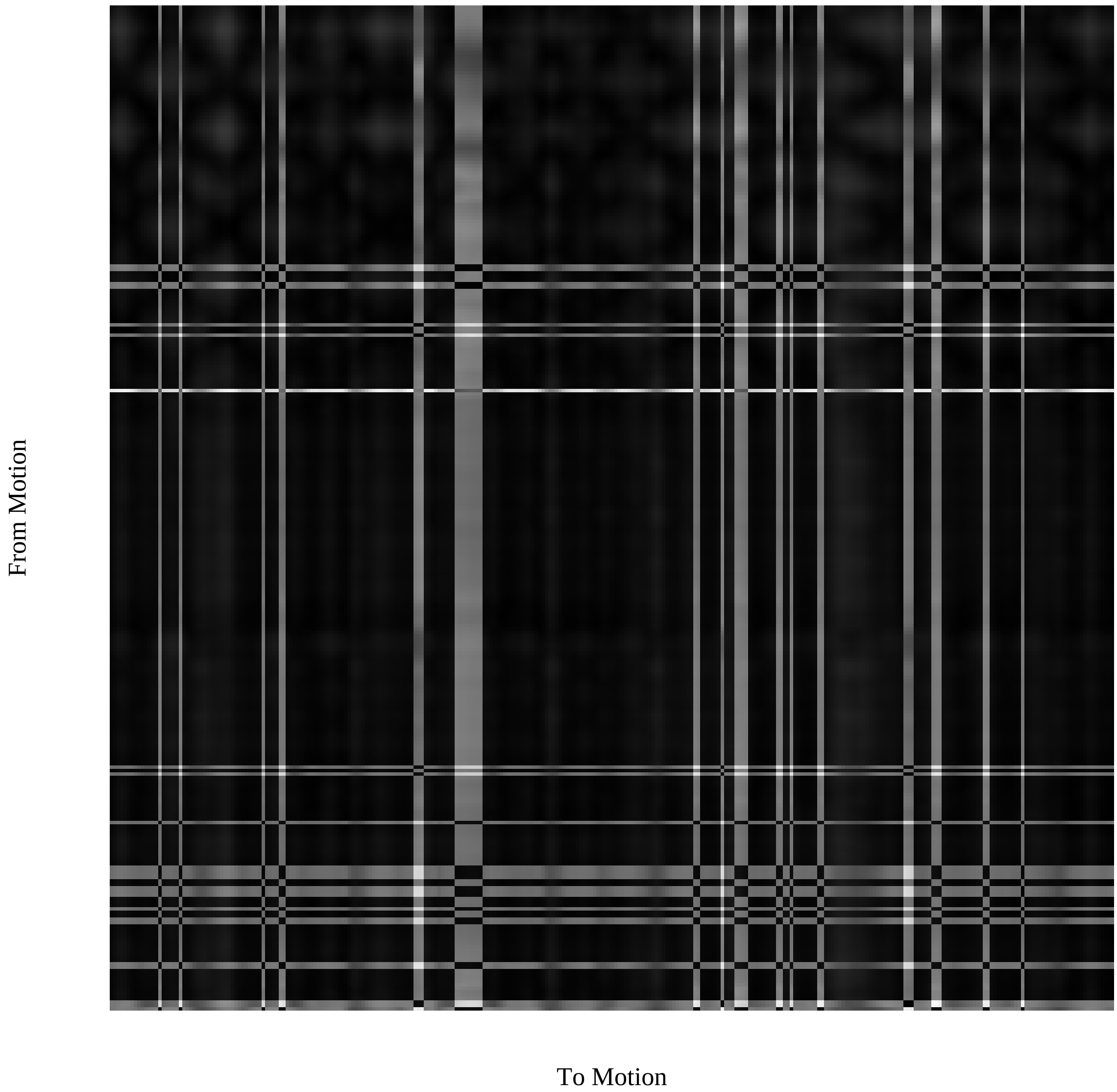


Figure IV.1: The cost matrix for two clips of dance motion with original weights. Each motion is 10 seconds long. Darker values correspond to lower costs for transition.

transition was one in which the visual discontinuity was disconcerting. The transitions were selected by a single person with animation experience and critically examined by two other experienced viewers for approval. Our optimization will depend on how well these transitions were selected, but in our experience it is not difficult to manually select good and bad transitions.

We then solved for the optimal values of the weights using a constrained least-squares minimization, we form a matrix A of example motions and joint distances as follows: the (i, j)th entries of A are the velocity and position difference terms for the jth joint in Equation IV.2 in motion i. We then solve

$$\min_{w} \|Aw - b\|_2^2 \tag{IV.5}$$

where w is a vector of weights, A is the matrix of the position and velocity distances of Equation IV.2 described above, and b is a vector of ones and zeros—an entry is one if it corresponds to a bad transition, and zero if it corresponds to a good transition. The optimization was constrained such that the weights were non-negative and symmetric, i.e., the weight for the left shoulder must be identical to the right shoulder. The symmetry constraint makes intuitive sense but will generally not be the result of the optimization without this constraint. The optimization problem was solved using an active set method similar to that described in [Gill et al. 1981].

Lee et al. actually use a slightly different cost function than Equation IV.4. They have a weight ν on the velocity term that trades off position and velocity. The weight ν enters the terms in the A matrix above non-linearly, thus requiring solving a more complicated constrained non-linear optimization problem. However, for motions in our database, our experience is that the velocity term makes little effective difference in the cost. Figure IV.2 shows this insensitivity to ν for transitioning from one dance motion to another dance motion, with $\nu = 0.1$ and $\nu = 10$. In fact, the global minimum for this motion was unchanged

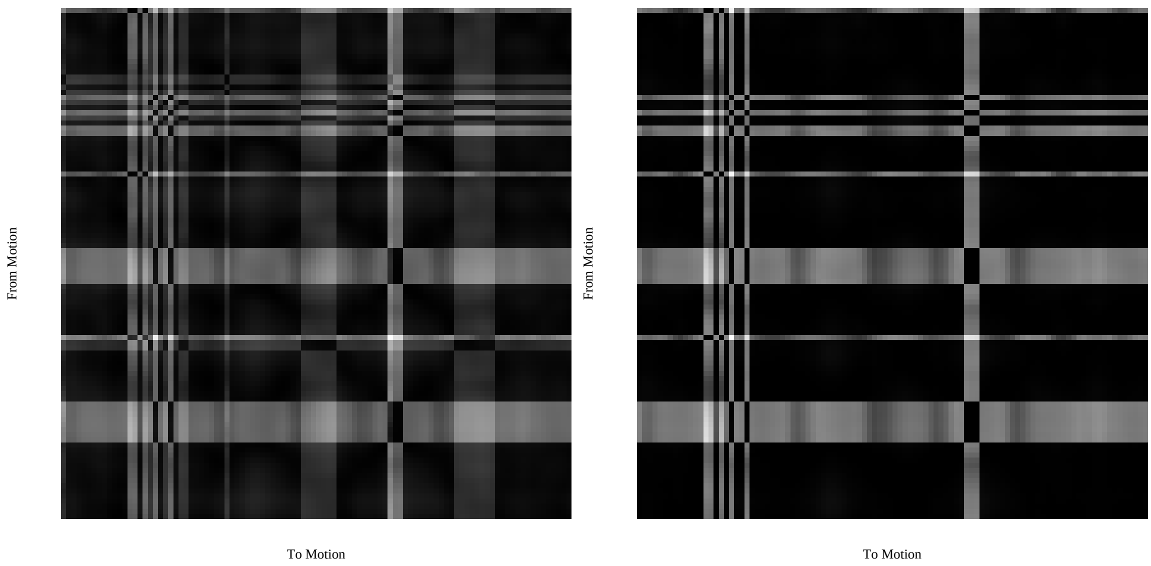


Figure IV.2: The cost matrix for two clips of dancing motion using the original weights. In (a) $\nu = 0.1$ and in (b) $\nu = 10$.

as ν was varied from 0 to 100. Thus, ν remained one in the optimization process.

The normalized weights (largest scaled to one) from the optimization process are shown in Table IV.1. The cost matrix for the motions using the optimal weights are shown in Figure IV.3. This figure is for the same motions that have the cost matrix shown in Figure IV.1. There is substantial difference between the original weights and the new weights. In general, the cost with new weights has become more restrictive. Numerically, the optimization zeroed weights associated with joints found to be unimportant. Most of the weights were found to be unimportant: only the hips, knees, shoulders and elbows were important. Note that a least-squares minimization is a quadratic optimization, so that the weights that don't help are driven to zero. The result is consistent with our expectation that those joints are the most important ones, but surprising nonetheless since they imply the rest are unimportant in selecting a reasonable transition.

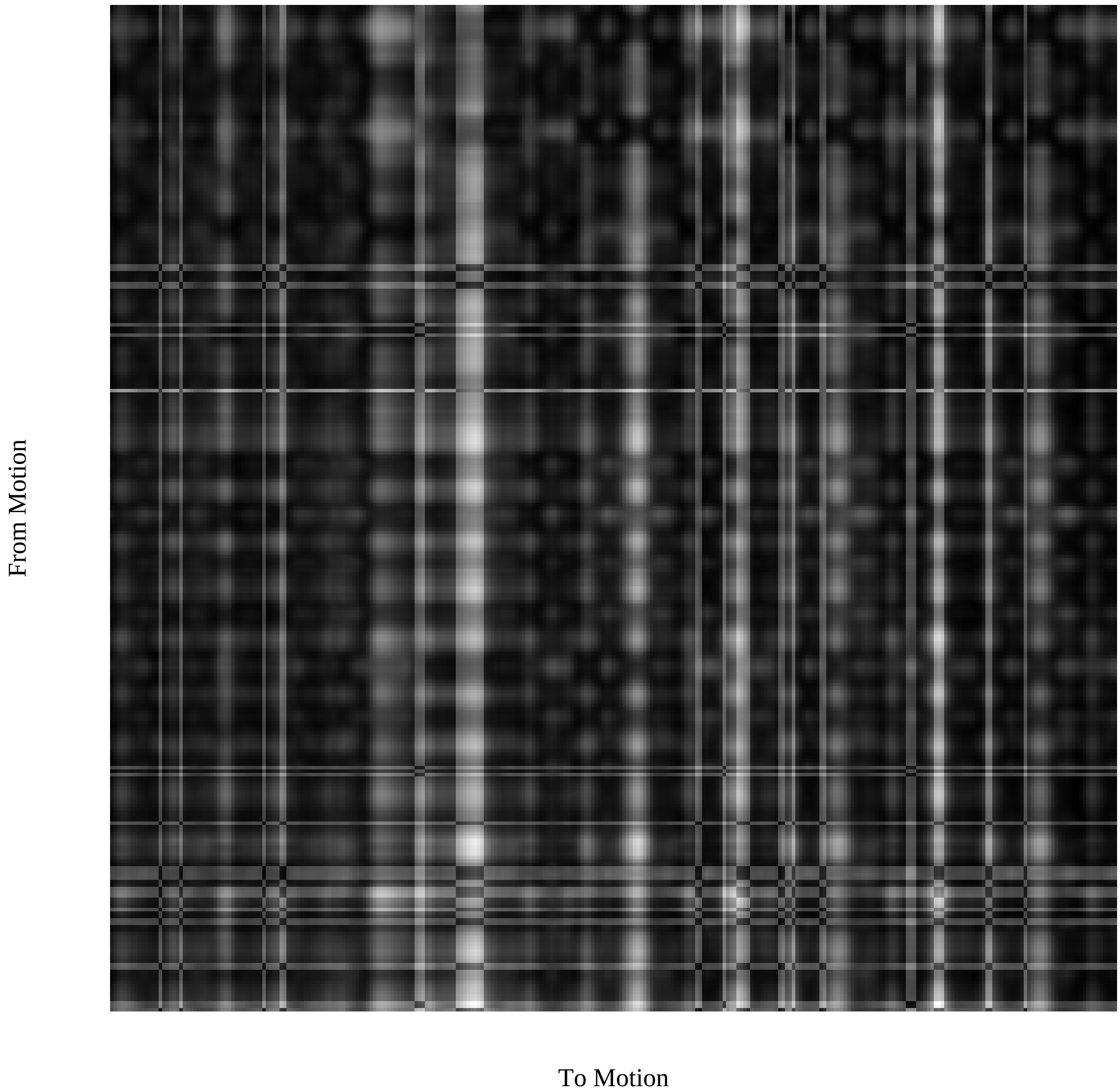


Figure IV.3: The cost matrix for two clips of dance motion with optimized weights. Each motion is 10 seconds long. Darker values correspond to lower costs for transition.

Table IV.1: Joints with non-zero weights and their associated weights when solved as described in the text. The optimization zeroed the weights for the remaining joints.

Joint	Original Weight	Optimized Weight
Right and Left Hip	1.0000	1.0000
Right and Left Knee	1.0000	0.0901
Right and Left Shoulder	1.0000	0.7884
Right and Left Elbow	1.0000	0.0247

IV.3 Cross-Validation

To estimate the generalization rate of the optimized weights, we employed a full leave-one-out cross-validation study [Duda et al. 2001]. In this technique, the weights are optimized with one set of training data deleted, and the resulting weights are then used to compute the optimal value of a transition for the deleted data set. Recall that our training set contained a rich variety of motions. The results of this study were quite encouraging. The average absolute deviation between the full optimization and that of the leave-one-out optimization was less than one frame in the animation sequences. The median absolute deviation was zero frames. Additionally, we performed a similar test by again deleting one set of training data, re-computing the optimal weights, and then computing the optimal transition for a completely different motion than the weights were trained on, a dancing motion from a different performer containing different dynamics. For this study, the resulting weights computed the same optimal transition in 41 of 42 cases. For the case where the optimal transition was not computed identically to the other cases, it was one frame different. Based on these empirical approaches, we believe that the optimal weights we computed are both robust and generalize to pick reasonable transitions for a wide variety of motions. However, whether the optimal weights are necessarily better than the original weights cannot be verified by this technique. Instead, we must conduct a user study to determine the result.

IV.4 Experimental Evaluation

IV.4.1 User Study of the weights

A user study was conducted to evaluate the weighting determined by the optimization. A motion capture sequence of dancing was created by a performer different from the one used to capture the motions used in Section IV.2 for optimizing the weights. This practice was employed to eliminate the possibility of any performer-dependent effects on the weighting. Several frames of an animated sequence used in the study are shown in Figure IV.4.

The participant group consisted of 26 adults with normal color vision who had no prior

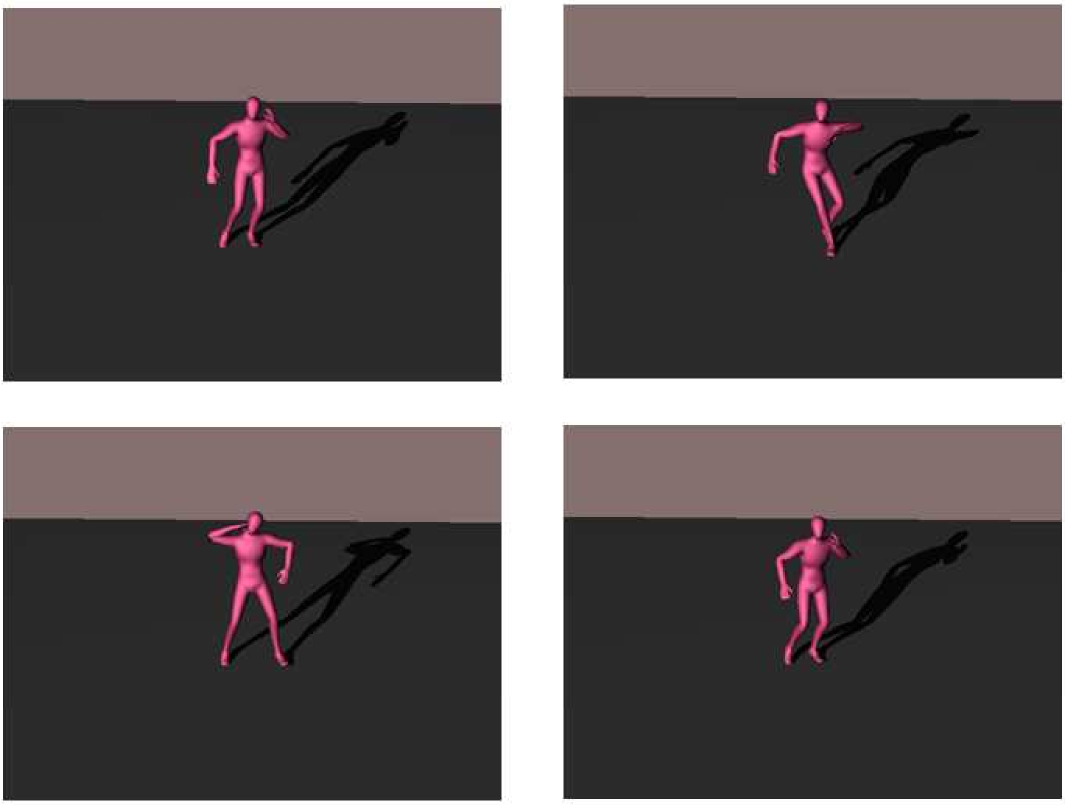


Figure IV.4: The animated character used in the user study.

experience working with animation outside of computer games and the like. Participants in the study were told they would be viewing an animation of motion sequences and shown an example animation of a walking sequence using the model that would be used in the experiment. They were then told the motions they would be viewing would have a discontinuity in the motion, and shown an example of an egregious discontinuity. Participants were told they would be asked to rate how noticeable and natural the discontinuities were, both individually and in comparison to another motion. Participants were shown two motions. Each motion was a six-second clip; the transition from the first motion sequence to the second occurred at $t = 3$s in the clip. The motions consisted of the globally optimum Lee cost transition with the weights used in Lee at al.[2002], and the globally optimum Lee cost transition using weights determined in Section IV.2. These two motions were different, i.e., the lowest cost transition occurred at different points in the motion for each cost function. In particular, for the dancing motion used in the user study, the optimal transition for the original cost occurred from frame 91 to frame 281, whereas the optimal cost transition occurred from frame 232 to frame 280.

The sets of motions participants were asked to compare were the original cost versus the optimal cost. To best evaluate the transition, no interpolation or smoothing between the sequences was done. Since a complete animation system, such as that present in a video game, will employ some sort of motion transition mechanism, this decision may seem odd. However, we believe that this decision is necessary for the following reasons. Employing a motion transition mechanism involves making many engineering decisions about how the motions are to be blended. For example, the time over which the transition will occur must be specified. The method of blending must be determined, e.g., linear interpolation, ease-in ease-out, or employing a very sophisticated mechanism such as in Rose et al.[1996]. Additionally, some sort of inverse kinematics routine is usually required, because blending introduces the problem of foot-skate or foot-slide [Kovar et al. 2002b]. There are a number of inverse kinematics routines available,[Lee and Shin 1999; Kovar et al. 2002b; Rose et al.

1996] and each of them also makes engineering decisions that affect the quality of the resulting motion.

Additionally, blending works two ways. It can mask the selection of a bad transition (for example, it can smooth a discontinuity), or it can make a bad transition extremely obvious (for example, if the transition causes one part of the body to intersect with another). Which effect predominates depends on the quality of the transition selected. Our work tries to evaluate the quality of transitions in the absence of the other confounding factors such as the decisions made in creating smooth transitions. Once we have empirically established that a metric produces reasonable or good transitions, then we can begin to evaluate the engineering factors associated with producing visually compelling and optimal transitions. We note, however, for each transition the global position and orientation of the character was matched at the point of transition.

We controlled for order effects in the presentation by randomly dividing the participants into two equal-sized groups. The first group was presented the original and optimized weighted motion sequences first, the second was presented the optimally weighted motion sequences first. After viewing each sequence, participants completed a post-sequence questionnaire consisting of questions asking them to compare and rate their impressions of the motions using a five-point Likert scale. Likert scale responses rather than forced choice responses were chosen to exploit the statistical power the former offers in distinguishing subtle differences. Users were asked which sequence seemed to have better quality, was more natural, and compare the sequences based on realism of the motions and the noticeability of the transitions.

Four questions were asked of the subjects in our study. The first was to rate the realism of motion generated using the original weights; the second question was to rate the realism of motion generated using the optimized weights; the third question was to rate the noticeability of the transition generated using the original weights; and the fourth was to rate the noticeability of the transition generated using the optimized weights. In analyzing the results of this study the null hypothesis regarding the realism and noticeability of the results is that the two methods are identical. The null hypothesis regarding the order condition is that the order of presentation is unimportant. Results of all the analysis were considered significant if $p < 0.05$.

If order effects were not tested as part of the design of this experiment, then statistical analysis of the results would be a classical example of a study suitable for analysis using the Wilcoxon signed ranks test, since the responses are ordinal. The inclusion of presentation order as a condition in the experiments complicates the analysis, since there is no 2×2 non-parametric tests suitable for application [Hayter 1986]. We could analyze the results straightforwardly using a 2-way ANOVA. When we did this, we found that order was not significant ($F = 0.87$, $MS = 0.94$, $p = 0.35$), i.e., that we cannot reject the null hypothesis for presentation order; that there were no interaction effects; and that subjects found the optimally weighted metric significantly more realistic that the original weights ($F = 6.39$, $MS = 6.94$, $p = 0.01$) but not significantly more noticeable ($F = 1.24$, $MS = 1.23$, $p = 0.27$). However, assuming that the Likert responses are samples of underlying continuous interval data is questionable. The problem here, of course, is that we don't know if the distance between "somewhat realistic" and "realistic" is the same as the distance between "not realistic" and "somewhat realistic" (for example). We could convert the Likert scale responses to continuous variables by analyzing responses based on frequency, which would be continuous. The drawback of this approach is the loss of statistical power we would encounter in the conversion.

We therefore analyzed our data using the following approach. The 2×2 design of the experiment means that we have four sets of data, which we will denote as indicated in Table IV.2. For the realism and noticeability questions, we applied a Wilcoxon to the data sets O1M1 and O2M1, and then to O2M1 and O2M2. Results from this analysis show that the order effects are not significant, so that we cannot reject the null hypothesis for order. Concluding that we then grouped the O1M1 and O2M1 data together to form a Method 1 data,

Table IV.2: Symbols denoting the data cells from the original weights versus optimized weights user study.

	Original Weights First Order 1	Optimized Weights First Order 2
Original Method	O1M1	O2M1
Optimized Method	O1M2	O2M2

Table IV.3: Summary of results for direct comparisons of optimized versus original functions. Preferences were rated on a scale of zero to four where zero corresponded to looking much worse (or very unnatural) and four corresponded to looking much better (or very natural). For example, participants were asked if they thought that the optimized motion "looked better" than the original motion.

Comparison	Mean	Std. Dev.
Optimized Weight vs. Original Weight		
Looks Better?	2.73	1.00
More Natural?	2.69	0.84

and the O1M2 and O2M2 data together to form a Method 2 data. Analyzing this data using a Wilcoxon signed rank test shows that the optimal transition for the weighted cost was significantly more realistic than the original cost ($p = 0.01$). Regarding the noticeability of the transitions, there was no statistically significant preference for either of the transition methods ($p = 0.34$) of the transition methods one over another. These results are shown in Table IV.3. This latter result may be because the relative transitions were not obvious to the users, or it may be that both transitions were noticeable, and that users simply preferred the noticeable transition using the optimized weights to the original weights. Since we are ultimately interested in transitions using blending, we did not pursue this question further. Finally, as an aside, we used the above "powering up" approach of combining orders into methods because a pairwise test of all combinations of order against method did not individually yield significant results for all comparisons.

IV.5 Discussion

This work produced weights for the cost function described by Lee et al. that led to the determination of superior transitions. We cross-validated the weights to assess the generalizeability and robustness of the optimization procedure. We compared the optimized weights with the original weights by running a user study. The user study showed that optimized weights produce perceptually better transition points.

However, there are several limitations and possible sources of bias in our results. When optimizing the weights, our motion data did not contain highly dynamic motions such as would be typical from a gymnastic floor exercise or other sources. The weights may not be a good predictor of good transitions for such motions. It was also limited to 16 different sequences of motion for the optimization. We would like to repeat our experiments using a larger library with more dynamic motion. Rendering style can affect the quality of the perceived motion as shown by Hodgins et al.[1998]. Our weight optimization used only one performer, although our motions were tested in the cross-validation and the user study on motion generated by a different performer. Finally, our motion data did not contain a large repertoire of "backward" motions, which may have resulted in the position-velocity weight ν having marginal impact for the Lee cost. Our data suggest that the velocity component of the cost function is not significant for a wide variety of motions. Removing these limitations or better understanding their necessity is an on-going project.

CHAPTER V

TRANSITION METHOD

Another important component of creating new motion is the ability to generate seamless transition. This research develops methods for determining a visually appealing blend length for a motion transition, i.e., a segue between two sequences of character animation. For reasons of efficiency and speed, linear interpolation is often used as the transition method. The blend length of a transition using this technique is critical to the visual appearance of the motion. Two methods for determining an optimal blend length for such transitions are presented. These methods work for different types of motion.

We chose to build our transition methods on top of linear blending because linear blending is the most common and widely used method for generating transitions between motion segments. However, linear blending violates the laws of physics because it distorts the real motions, being the weighted sum of two or more motions. Linear blending is nonetheless a popular method because it is simple and often generates visually pleasing results.

For two motions, spherical linear interpolation is used to blend between the quaternions of each joint using a linear weight function. A sigmoidal weight function produces similar results with only subtle differences. The facing direction and the position of the figure on the floor plane are aligned during the blending. We assume in this work that a start frame in a *from* motion and an end frame in a *to* motion are specified. The start and end frames indicate the beginning and end of the transition, respectively. In next chapter, we will discuss other ways of specifying motion transitions and blending motions.

Linear blending may introduce artifacts such as foot-slide. To fix such problems, inverse kinematics or other techniques [Kovar et al. 2002b] are often used as a post-process. These methods may be automatic. We used the inverse kinematic solver provided by MotionBuilder 4.02 to constrain support limbs and correct foot-slide. Other than correcting foot-slide, it rarely affects the visual appearance of the motion.

The methods for generating transitions are empirically evaluated by conducting user studies. The studies for the transition methods show (1) that visually pleasing transitions could be generated using our optimal blend lengths without further tuning of the blend parameters; and (2), that the users prefer these methods over a generic fixed-length blend.

V.1 Methods for Computing Blend Length

We develop two methods to compute blend length based on two hypotheses on the nature of blending.

V.1.1 Method I: Using the Geodesic Distance

One hypothesis for motion blending is that a transition will be smooth if two windows of the motions to be blended have strong correspondences, which implies that these two pieces of motion have consistent velocities. We compute the best blend length for blending between two arbitrary frames by calculating the cost for blending where the blend length normally ranges from 0.03 to 2 seconds (1 to 60 frames), and pick the blend length with minimum cost.

The per-frame cost for transitioning from frame i to frame j with blend length b is computed by averaging the difference of all pairs of corresponding frames within the blend window and is given by

$$D_{f_i f_j} = \sum_{t=1}^{b} d_{f_i f_j t}/b. \tag{V.1}$$

In this equation, $d_{f_i f_j t}$ is the difference between two corresponding frames given by

$$d_{f_i f_j t} = \sum_{k=1}^{m} w_k \left\| \log \left(q_{j-b+t,k}^{-1} q_{i+t-1,k} \right) \right\|^2 \tag{V.2}$$

where m is the number of joints in the figure, and $q_{i,k}, q_{j,k}$ are the orientations of joint k at frames i and j, respectively, expressed as quaternions. The log-norm term represents the geodesic norm in quaternion space, and each term is weighted by w_k. Note the difference

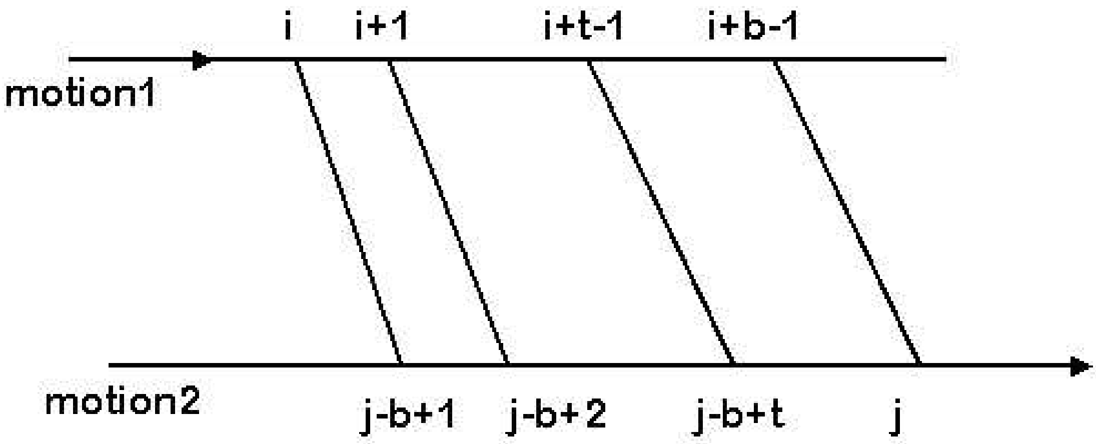


Figure V.1: The cost for blend length b is the average of the difference of corresponding frames. The transition is from frame i to frame j.

between two corresponding frames is basically calculated by the cost function described in the last chapter. The weights were those determined in the last chapter. Global degrees of freedom of the beginning of the second motion were matched to the end of the first motion and interpolated during the transition.

Figure V.1 illustrates how the cost as a function of the blend length b is calculated. The cost is the sum of the difference of corresponding frames. Once the costs for a blend length from 0.03 to 2 seconds are computed, the minimum cost can be computed; the optimal blend length is given by this minimum. An example of the cost for a walking to walking transition versus different blend lengths is shown in Figure V.2. The optimal blend length is 0.5s.

V.1.2 Method II: Using Joint Velocities

Another approach for predicting optimal blend length is inspired by the idea that the rate of change for any joint in the pose should not change radically for a smooth transition. We calculate the joint difference between the start frame and the end frame for each major joint, i.e., shoulders, elbows, hips and knees. We then compute the optimal blend length based on the velocity of the joint that has the maximum difference between the start and

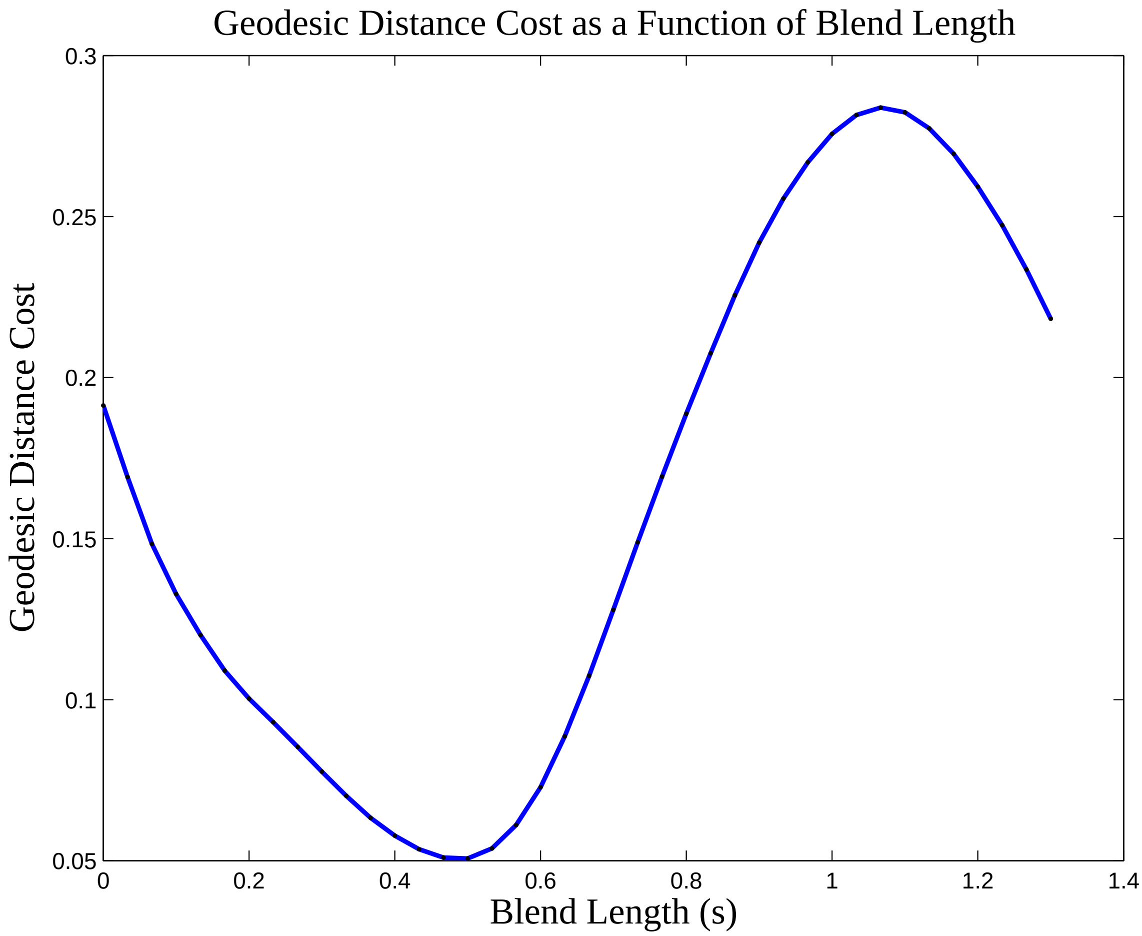


Figure V.2: An example of cost as a function of blend length. The optimal blend length is 0.5s for this example. The transition is from a walking motion to another walking motion.

Table V.1: An example of the joint differences between two frames of motions. The maximum difference is on the right shoulder.

Right Hip	0.0352
Right Knee	0.0481
Right Shoulder	**0.3029**
Right Elbow	0.0167
Left Hip	0.0044
Left Knee	0.0288
Left Shoulder	0.1192
Left Elbow	0.0178

end frames.

The difference between frame i and frame j for joint k is

$$d^k_{f_i f_j} = \left\| \log \left(q^{-1}_{j,k} q_{i,k} \right) \right\|. \tag{V.3}$$

The optimal blend length is then the joint difference of the joint with maximum difference divided by the velocity of that joint

$$d^p_{f_i f_j} / \|v\| \tag{V.4}$$

where p is the joint with the maximum joint difference, and v is the average of the joint velocity for the beginning frame and the end frame of joint p, respectively. Table V.1 shows an example of the joint differences between two frames of motions. The maximum difference is on the right shoulder.

V.1.3 *Ad Hoc* Comparison

Our experience is that of these two methods, the geodesic distance is more suitable for cyclic locomotion such as walking and running as shown in Figure V.3 for which correspondences in the blend are critical. One possible reason is that cyclic motion has a fixed pattern and people are sensitive to movements that are out of phase. The geodesic distance method aims at the phase requirement of these motions and finds the best correspondence of frames of motion for blending.

On the other hand, the velocity method is more suitable for physical activities such as boxing and free-style dancing, etc. Figure V.4 shows example of motions in this category. For motions like these, people do not have strict perceptual predictions for the next move. However, a longer blend length does not necessarily mean a better transition. For example, a rather quick punch by a boxer might become a slow punch after a long blending. Therefore, finding the optimal blend length that produces smoothness and still preserves the quality of the target motion becomes important. The velocity method meets these requirements by smoothing the movement of every joint and does not unnecessarily stretch the resulting motion.

V.1.4 Alternative Methods

We studied alternative techniques for computing a good blend length, more complex and computationally expensive than the previous two. As noted in Section III, timewarping has been used for generating transitions. Thus, we modified the geodesic distance method to compute a blend length where the *from* and *to* motions can be timewarped. Given a transition from frame i to frame j, a timewarped blend length is calculated by computing a cost matrix of blend lengths in the *from* motion versus blend lengths in the *to* motion. Each entry in this cost matrix C_{ij} is given by

$$C_{ij}(b_{from}, b_{to}) = \sum_{t=1}^{b_{to}} \sum_{k=1}^{m} w_k \left\| \log \left(q^{-1}_{j-b_{to}+t,k} q_{i+\frac{b_{from}}{b_{to}}t-1,k} \right) \right\|^2 \tag{V.5}$$

if $b_{from} \le b_{to}$, and

$$C_{ij}(b_{from}, b_{to}) = \sum_{t=1}^{b_{from}} \sum_{k=1}^{m} w_k \left\| \log \left(q^{-1}_{j-b_{to}+\frac{b_{to}}{b_{from}}t,k} q_{i+t-1,k} \right) \right\|^2 \tag{V.6}$$

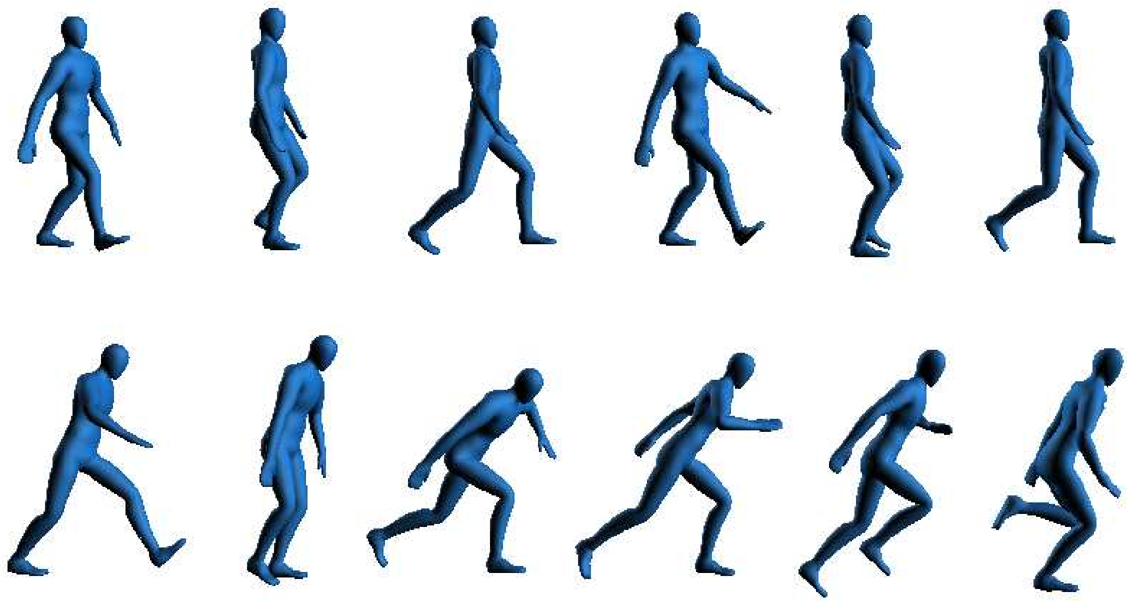


Figure V.3: Examples of walking motion and running motion in the category of cyclic locomotion. The geodesic distance method is suitable for these types of motion.

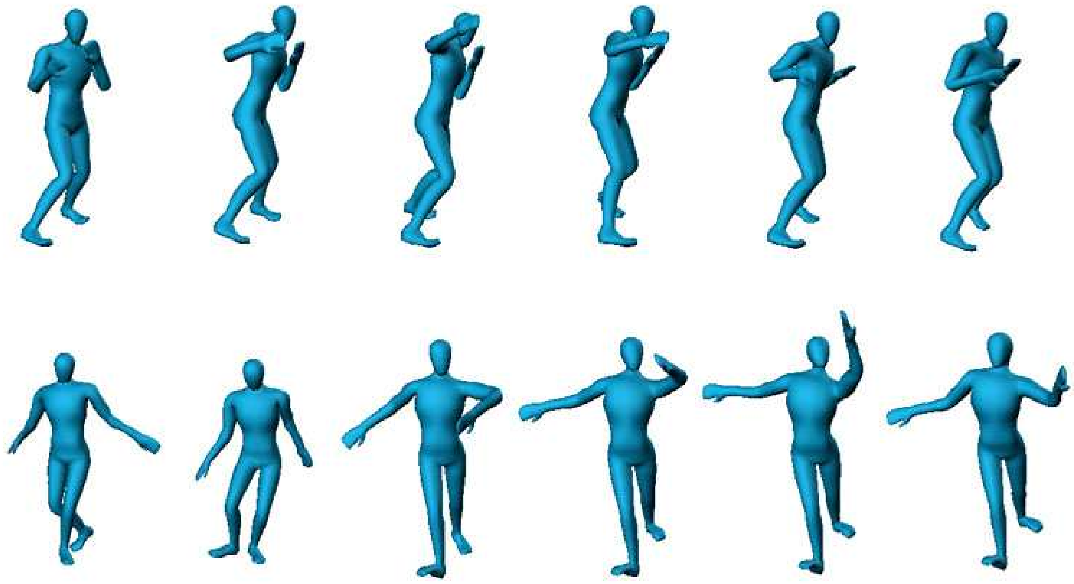


Figure V.4: Examples of boxing motion and free-style dancing motion. The velocity method is suitable for these types of motion.

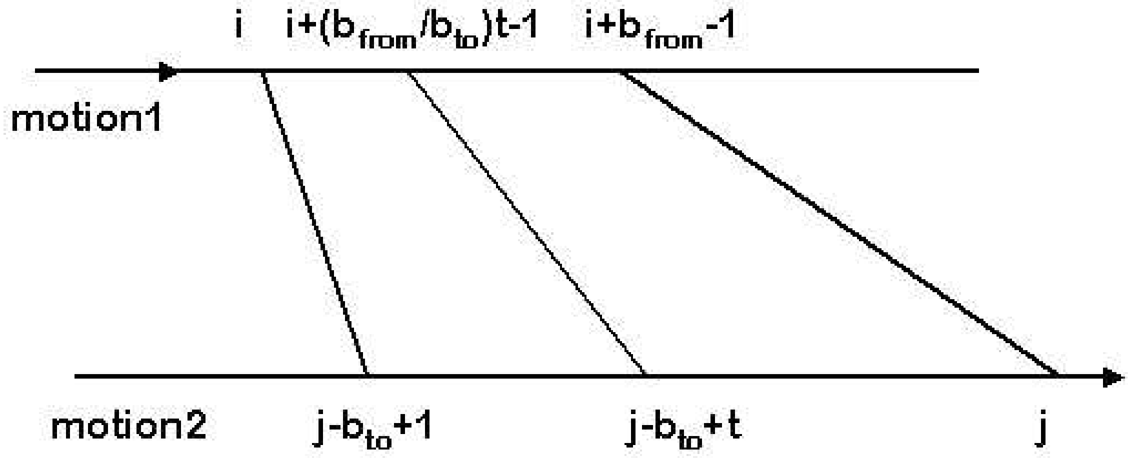


Figure V.5: An illustration showing the method for computing the cost of a blend length with timewarping. The blend length for the *from* motion is b_{from} and the blend length for the *to* motion is b_{to}. In this example, $b_{from} < b_{to}$.

if $b_{from} > b_{to}$, where b_{from} and b_{to} are the blend lengths in the *from* and *to* motions, respectively, and the other terms are defined as in Equation V.2. Figure V.5 illustrates how this cost is computed. The minimal cost from the cost matrix C_{ij} then gives the appropriate motion intervals with which to perform timewarped blending. When a computed frame time is not an integer, joint values of the pose are obtained by spherical linear interpolation between the two adjacent frames. Note that computing the optimal blend length using the geodesic distance method is $O(b)$ whereas the timewarped blend method is $O(b^2)$.

The second alternative we explored is the idea of using a non-uniform blend schedule on the degrees of freedom to produce a transition. We could, for example, transition a shoulder degree of freedom over 10 frames and a hip degree of freedom over 20. There are two drawbacks to this method, both related to the physical properties of the motion. First, as shown in Figure V.6, the physical coherence of the individual joints indicates that the optimal blend length as computed by the geodesic distance occurs at the same value for most important degrees of freedom. This figure shows the geodesic cost for individual joints of the motion shown in Figure V.1. The joints that do not have minima at 0.5s are the left shoulder, right shoulder, and right elbow, although the cost for the left shoulder is nearly

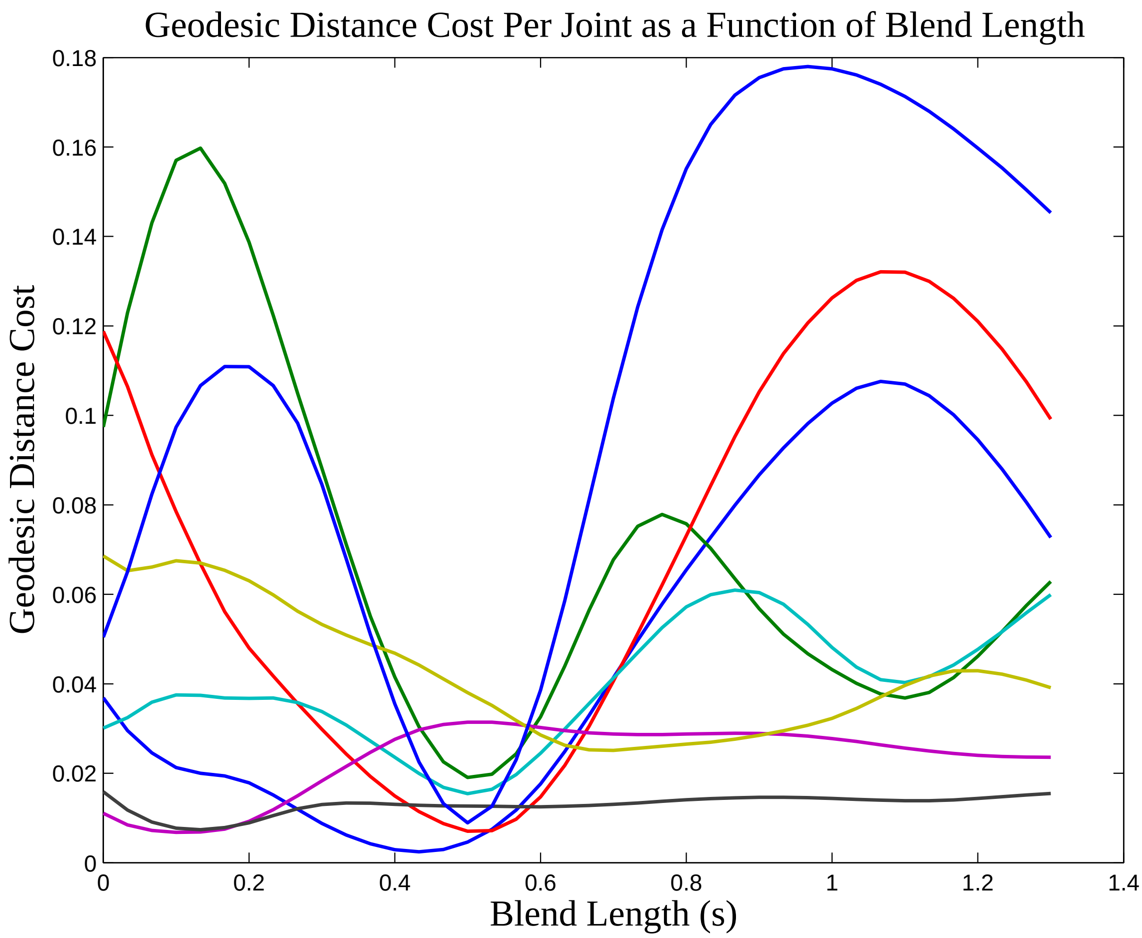


Figure V.6: The geodesic distance cost for each of the joints of the motion transition shown in Figure V.1 (walking to walking). The joints that do *not* have minima at 0.5s are the left shoulder (black), right shoulder (yellow) and right elbow (magenta).

constant. The second, more important drawback is that different blending schemes destroy the physical coherence of the degrees of freedom. When trying to adapt the methods to different blend schedules, visual artifacts were apparent.

We additionally tried to modify the velocity method according to the methods implemented by [Grassia 2000]. We found no improvement from the basic method described above.

V.2 Experimental Evaluation of the Transition Methods

There are a number of interesting psychophysical evaluations that could be conducted to the methods described previously. In this study, the methods described above were compared to a typical transition scheme that employed a fixed blend length. The goal of these experiments was to determine whether users preferred the methods' results and how strongly over a wide repertoire of motions.

V.2.1 Procedure

The fixed blend length we chose was 0.33s (10 frames). This value was chosen because it is the value suggested by Mizuguchi et al. [2001], used by [Kovar et al. 2002a], and in the range specified by both [Rose et al. 1996] and [Pullen and Bregler 2002]. Our experience from conducting pilot studies prior to these experiments leads us to believe that the results described here will hold for any fixed blend length.

All experiments were run in a single session consisting of four distinct studies. The participants were volunteers from our institution with no prior animation experience beyond exposure to video games and film. Thirty-five people volunteered: 20 male and 15 female, aged 22 to 40 years. All participants had normal or corrected-to-normal vision. Additionally, participants were naïve as to the purpose of these experiments.

Motion transitions were created from a variety of motion capture data and shown in the same rendering style. Groups of motions were shown from the same camera position. Consistent with the point light experiments mentioned in Section III, we chose to omit rendering a ground plane. While the ground plane can provide important visual cues for some perceptual studies, e.g., [Reitsma and Pollard 2003], we judged it unnecessary for our purposes.

Study One: Geodesic distance method versus fixed blend-length

In this experiment, we studied whether participants judged that motions containing a transition generated by the geodesic distance method appeared more natural than motions containing a transition using a fixed blend-length of 10 frames. We selected eight different motion transitions consisting of such motions as standing and idling to walking or running (of different speeds), walking to running (of different speeds), and various turning motions. As discussed previously, these motion types are those we believed most suitable for the geodesic distance method. None of the optimal blend lengths were close to 10 frames. Optimal blend lengths for the motions tested ranged from 5 frames to 35 frames. Motion pairs were generated, one containing the optimal blend length and one containing the 10 frame blend length. The order of these was randomized.

Participants were presented eight motion pairs and asked to determine whether the first or second motion of a pair was more natural. They were again given five seconds between each motion pair to make their determination.

Study Two: Velocity method versus fixed blend-length

This experiment was conducted to determine whether participants judged motions containing a transition generated by the velocity method to appear more natural than motions containing a transition generated by a fixed blend-length. The experimental procedures and preparation of stimuli were identical to Study One above, except that the eight motions chosen for study consisted of boxing, dancing, and tai-chi motions, motions of a type we believed most suitable for the velocity method. Participants were again presented with eight motion pairs.

Study Three: Geodesic distance method versus timewarping method

In this experiment we studied whether participants judged that motions containing a transition generated by the geodesic distance method appeared more natural than motions containing a transition generated using the timewarping strategy discussed in Section V.1.4. The experimental procedures and stimuli were identical to those of Study One. The timewarped transitions averaged a warp of 10 frames; for example, one motion blended 18 frames in the *from* motion to 28 frames in the *to* motion.

Study Four: Geodesic distance method versus velocity method

This experiment was composed of two parts. The first part is to determine whether participants judged motions in the cyclic locomotion category such as walking and running containing a transition generated by the geodesic distance method to appear more natural than motions containing a transition generated by velocity method. The second part is to determine whether participants judged motions in the physical activity category such as boxing and dancing containing a transition generated by the velocity method to appear more natural than motions containing a transition generated by geodesic distance method. The experimental procedures and preparation of stimuli were identical to Study One above, except that participants were presented with three motion pairs while each pair contains transitions using geodesic distance method and velocity method presented in random order for the each of the two parts.

V.2.2 Results and Analysis for Studies of Transition Methods

Comparison of Methods

Table V.2.2 shows the percentage of study participants preferring various methods over the others as tested in Studies One, Two, Three, and Four. In particular, we see that 96.4% of the participants thought that the geodesic distance looked more natural when compared to a 10-frame blend for the motions studied, 65.7% of the participants favored the velocity method over the 10-frame blend when asked which produced more natural motion, 55.7% of participants favored the geodesic distance method over the timewarping method, 75% of participants favored the geodesic distance method over the velocity method for cyclic locomotions, and 88.9% of participants favored the velocity method over the geodesic distance method for physical activities.

Also shown in Table V.2.2 is the χ^2 test statistic applied to these studies. The χ^2 test is used to test the significance of a preference. In other words, we wonder whether the observed percentage deviates from the value expected by chance or sampling error alone.

The expected frequencies are equal–that is, half the sample would be expected to be in each of the two categories by "chance." An alpha level of .01 was used for all statistical tests (the critical value of χ^2 for this alpha is 6.64). There is one degree of freedom and a sample size of 280 (35×8) for studies One, Two, and Three. The sample size is 36 (12×3) for each part of study Four.

The observed percentages of users preferring the geodesic distance method and velocity methods over 10-frame blending is statistically significant. However, it is not clear that users can successfully distinguish between the geodesic method and our timewarping method. This result is supported by the comments of many of the participants who noted that the motions seemed very similar. The same results are found when the data is analyzed on a per motion basis. Also, for Studies One and Two, there were no individual motions for which users preferred the 10-frame blend.

Study Four shows that users have preference of methods for generating transitions one over another. More specifically, users prefer geodesic distance method over velocity method for cyclic locomtions such as walking, running, and jogging, and velocity method over geodesic distance method for physical activities such as boxing and free style dancing. This observation is consistent with our hypothesis that geodesic distance method targets at the correspondence of motions thus it is suitable for motions with fixed pattern, whereas for motions like boxing, movements of the player are often not predictable, and people are rather sensitive to changes in velocity.

V.3 Discussion

We developed two methods for determining the best blend lengths for generating a transition between two motions using linear blending. Visually appealing transitions are critical in the re-use of large motion data-sets, and the transition duration is one of the most important factors in creating a compelling transition. Human motions are highly varied, and developing a universal method for generating compelling transitions may not be possible.

Table V.2: The percentage of users that preferred various methods against other methods in Studies One, Two, Three, and Four. The first column gives the percentage favoring the first method listed over the second, and the second column gives the χ^2 test statistic for the experiment.

Method	% favoring	$\chi^2(1, N = 280)$
Geo. dist. over 10-frame	96.4	241.4, $p < .01$
Velocity over 10-frame	65.7	27.7, $p < .01$
Geo. dist. over Timewarping	55.7	3.66

Method	% favoring	$\chi^2(1, N = 36)$
Geo. dist. over Velocity on cyclic locomotion	75	9, $p < .01$
Velocity over Geo. dist. on physical activities	88.9	21.8, $p < .01$

Thus, investigating methods that work for categories of motions seems reasonable.

The first method, which we call the geodesic distance method, determines the best blend lengths for motions that have a cyclic nature, such as running and walking. The second method, which we call the velocity method, determines the best blend lengths for motions that are non-repetitive, activities such as free-form dancing and boxing. These methods automatically generate a blend length for linear blending given two motions and the frames in those motions to transition between. This information is readily available from such systems as [Lee et al. 2002; Kovar et al. 2002a; Arikan and Forsyth 2002]. There is no need for further modification of blend parameters by a user or animator.

In our experience, our methods work on a wide variety of motions and transition points. However, we also performed a quantitative evaluation of these methods through a user study. Users were shown transitions between motions appropriate to the particular methods. These motions consisted of running at different speeds, walking at different speeds, standing, idling, boxing, dancing, and tai-chi. When compared against a fixed blend length, users preferred both the geodesic distance and velocity methods for calculating blend lengths. The geodesic distance method was strongly preferred. In our user study, there were no motion transitions for which users consistently preferred the fixed blend length.

A surprising result of our study was that users showed no preference for our timewarping method over the geodesic distance method. Timewarping was found to be helpful for generating transitions by a number of researchers, e.g., [Bruderlin and Williams 1995; Rose et al. 1998; Kovar and Gleicher 2003]. We conjecture that there are two major reasons for this contradictory finding. First, the method of timewarping used by [Bruderlin and Williams 1995; Kovar and Gleicher 2003] is more sophisticated and powerful than our technique, involving dynamic programming. We avoided employing dynamic programming because its computational cost precludes its use in a system where performance demands are interactive and high, e.g., a video game. However, there are quite likely advantages to the more expensive approach. Another reason may be that timewarping has been found to be useful when the motions transitioned between have very different timings. While we included such motions in our user study, we may not have included motions with significant enough time variations to make timewarping necessary. More investigation of this area is ongoing.

The methods described in this research could be easily integrated into the systems described by [Lee et al. 2002; Kovar et al. 2002a; Arikan and Forsyth 2002; Arikan et al. 2003]. These systems determine transition points as part of their function. The only additional information needed would be the category of the motions, so that either the geodesic distance method or the velocity method could be chosen. The same holds true for integration into a video game. In particular, the computational cost of these methods is minimal and well within the performance bounds set for animation by most rendering engines.

An important issue for any automatic technique for re-using motion data is its applicability to motions for which it has not been tested. While our motion capture library is reasonably extensive, it does not contain highly specific motions that would be needed in, for example, a video game dealing with hockey. We may find new categories of motion for which we require different methods. Moreover, the perception of visual artifacts depends upon the task [Oesker et al. 2000] and upon the rendering style [Hodgins et al. 1998]. Insofar as task differs from the motion itself we assume that these effects are not significant, but have not tested this assumption.

Also, linear blending often exhibits artifacts when foot-slide occurs. In this work, a support limb is constrained to prevent foot-slide using a particular inverse kinematics formulation. There are other solutions to this problem [Kovar et al. 2002b; Lee and Shin 1999]. We believe that the artifacts present in the motion are dominated by the relative velocities of the two motions and differences in pose, and thus *not* by the mechanism of support limb constraint.

CHAPTER VI

THE JUST NOTICEABLE DIFFERENCE

This work has so far focused on improving the cost metric for picking transition points and developing methods to compute good blend lengths using linear blending. There is an important issue involved in determining blend length for transitions, that is, how noticeable the blend length of a transition was, i.e., how sensitive users are to changes in motion transition duration. If users are largely insensitive to changes in blend length, then the methods used to determine those lengths may be unimportant.

In this chapter, the results on the noticeability of transitions between two segments of motion are reported, for two transition specifications, start-end transitions and center-aligned transitions, respectively. These two transition specification are first discussed, and the methods for conducting a study for determining sensitivity of users are presented, finally the results are reported and analyzed.

VI.1 Transition Specification

There is no generally accepted standard for generating or specifying a transition in graphics community. [Rose et al. 1996; Rose et al. 1998; Mizuguchi et al. 2001; Kovar et al. 2002a] specify transitions using a start and end frame and blend from there, as shown in Figure VI.1. An alternative way is to specify a transition from frame i to frame j, what is meant is that frame i and frame j are the 50% in the blend ("center-aligned"). [Arikan and Forsyth 2002; Arikan et al. 2003; Kovar and Gleicher 2003] use this transition specification. Figure VI.2 illustrate the center-aligned transition scheme.

There are advantages and disadvantages to each of these two methods. Start and end frames have the advantage that they are intuitive and easy to specify. They also work well if the transition points are at the end or beginning of motion segments. They can change the alignment of the motions as the duration of transition is changed. Center-aligned

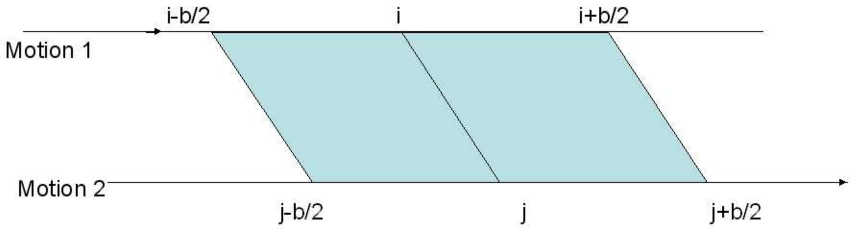


Figure VI.1: An illustration of center-aligned transition.

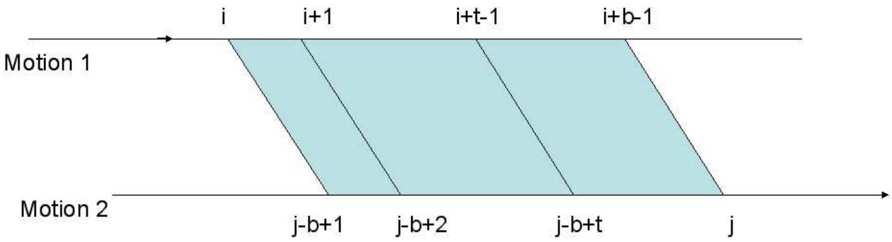


Figure VI.2: An illustration of start-end transition.

transitions have fixed alignment, which is both an advantage and a disadvantage. If the center-aligned poses are quite similar, then a center-aligned transition is more robust to variations in the blend length. On the other hand, if the poses are mismatched, then no amount of blending will make the transition look good. Thus the center-aligned transitions put a burden on the cost metric for picking transition points. Center-aligned transitions also have the disadvantage that depending on the blend length there is a region at the beginning and end of each motion segment for which a true blended transition can not be made.

VI.2 Method

We ran a set of experiments to determine the sensitivity of subjects to changes in blend lengths of motion transitions, i.e., to estimate the psychometric function of the differential threshold (or “just noticeable difference”). The “just noticeable difference” is the amount that something must be changed for the difference to be noticeable. It is observed by Ernst Weber, a 19th century psychologist, that the size of the differential threshold appeared to be related to the original stimulus magnitude [Hita et al. 1984]. In this experiment, the original stimulus, i.e., the blend length for the baseline transition, varies for the start-end transitions according to motions used in the study with relatively small variation, and are constant for the center-aligned transition.

We asked users to make a series of two-alternative forced choice responses. Participants were asked to watch two sets of animations. Each set contains a pair of motions. One set is the reference pair, consisted of two identical motions containing a baseline transition, a length k transition. K varies for motions and transition schemes used in the study as mentioned earlier. The motions in the other set are different, consisting of a length k transition and a transition with other lengths. Participants were asked to watch two motion sets and to determine which motion pair consisted of different motions. This somewhat cumbersome design is necessary to make a two-alternative forced choice test with an objective answer, meaning that there are two choices and a correct answer as to which set of motions is different. The presentation of the transitions was randomized, both pairwise and among pairs to eliminate the order effect. Subjects were allowed to guess if they were not sure about the answer.

VI.2.1 Constant Stimulus vs. Adaptive Method

First we conducted a pilot study using a constant stimulus to determine both the shape of the underlying psychometric function and the granularity with which a fine-grained study must be conducted. The pilot study uses “method of constant stimuli” [Leek 2001], where the threshold is extracted from a fully sampled function. However, in terms of experiment time, the method of constant stimuli is very expensive. The pilot study takes approximately four hours per person and is thus impractical for testing many subjects. Many trials are placed on locations that are not informative. Thus the pilot study conducted by two people only serves as the basis for the adaptive experiment, where most trials focus on the interesting region around the threshold with finer sampling. We chose a staircase adaptive method that employs an up-down procedure and minimizes the number of the trials.

In an adaptive study, the motion in the two sets of motions that is not of length k will be increased or decreased according the subject's past responses. We used a variant of the asymmetric staircase described by [Wetherill and Levitt 1965]. In our staircase the difference is continually decreased provided the subject has correctly differentiated the two previous tests. Upon an incorrect response, the difference is continually increased until the subject again correctly differentiate two consecutive sets of motions (See Figure VI.4 for an example). Our staircase converges to a discrimination of 69%, determined by Monte Carlo simulations. The staircase of [Wetherill and Levitt 1965] converges to 71% correct discrimination. As an aside, a symmetric staircase was used in the graphics community in [O'Sullivan et al. 2003].

VI.3 Experiments and Results

In the experiment for both center-aligned transitions and start-end transitions, the following variables were recorded: the sample point (transition duration) visited, the motion showed to the participant, the correct answer of the trial, and the answer of the participant. Motions were shown with randomly picked order of presentation in real time, i.e., the order of the reference pair and comparison pair and whether the different motion appears first or second in the comparison pair are both randomized in real time. The stopping condition for the adaptive procedure occurs when any sample point is visited ten times. The user interface for the adaptive study is shown in Figure VI.3. Figure VI.4 shows the adaptive track of one participant in the study. We can see that the number of trials is significantly reduced, and most of the trials were conducted around the threshold, which is the blue line in the figure. Such an adaptive study normally takes only around half an hour.

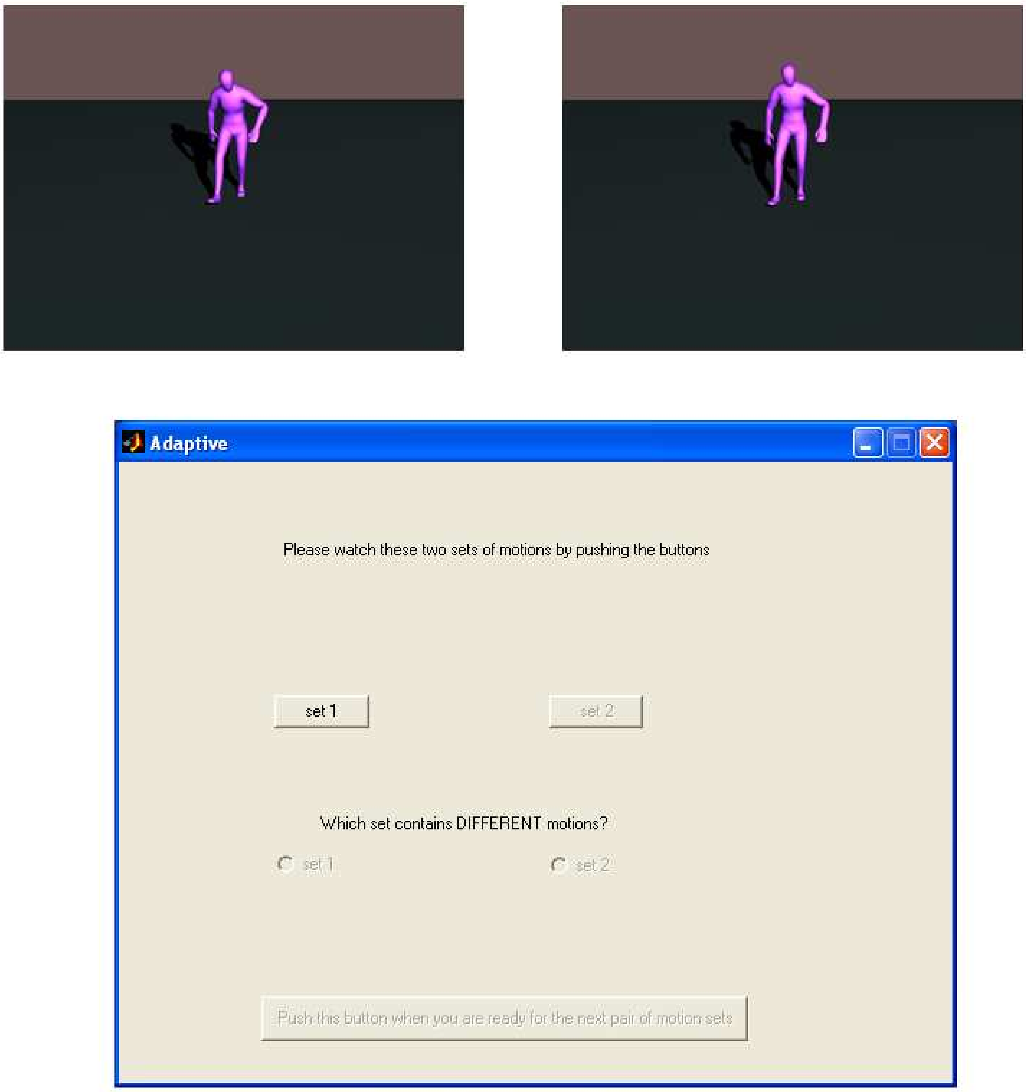


Figure VI.3: The user interface of the adaptive procedure.

VI.3.1 Center-aligned Transitions

For the center-aligned transitions between two different motions, transitions using blend length of 10 frames (k = 10) were generated, and also transitions of lengths 2 (k-8) to 22 (k+12) at two frame intervals were generated. Two different motions were used. Twelve adults participated in the study. Figure VI.5 shows the result of the study. The result we conclude from this study is that people can differentiate between center-aligned transition lengths that differ by -7 or 10 frames for center-aligned transitions. There was no statistically significant difference in the performance of the test across motions.

VI.3.2 Start-end Transitions

Similar studies were conducted for the start/end transitions between two different motions. Transitions using the optimal transition length k computed using the geodesic distance method were generated, and transitions of lengths k-15, k-10, k-5, k, k+5, k+10, and k+15 were generated for the pilot study. Three different motions are used for the adaptive study, k = 9, 15, and 16 respectively. More transitions of lengths around the thresholds determined by the pilot studies were generated for the adaptive studies. Again, twelve adults participated in the adaptive study. Figure VI.6 shows the results of the adaptive study for the start-end transitions. We found that people can normally distinguish between transitions length that differ by 2 or -3 frames for start-end transitions. There was no statistically significant difference in the performance of the test across motions. People are more sensitive to start-end transitions since some of the natural alignment of the motion is wrapped into the transition.

VI.4 Discussion

The blend length or duration of the transition is a critical component in the visual fidelity of a spliced animation system. The purpose of the just noticeable difference study is to know how well one can discriminate transitions differing in blend length. It saves the time for the animator of any unnecessary and overdone change of the blend length and provides

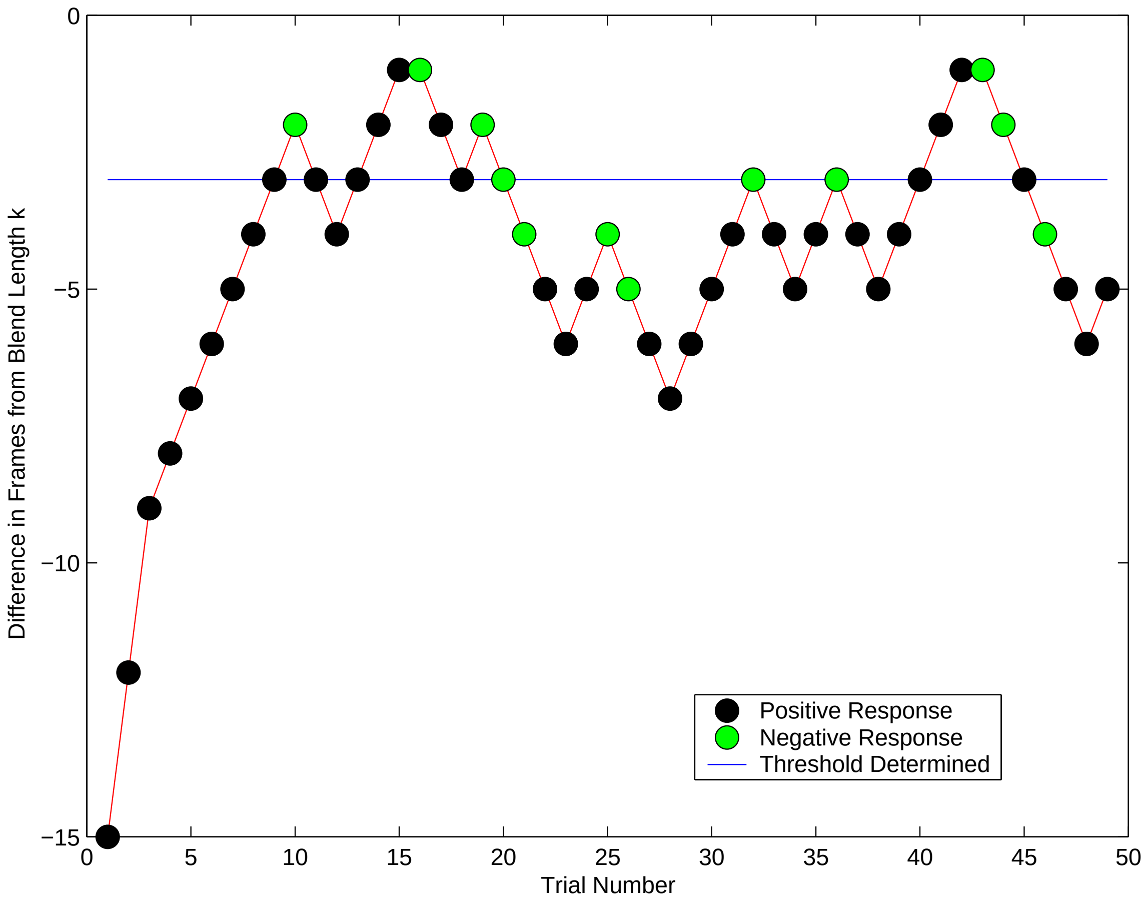


Figure VI.4: The adaptive track following the two-down, one up staircase procedure, for a test subject in our study.

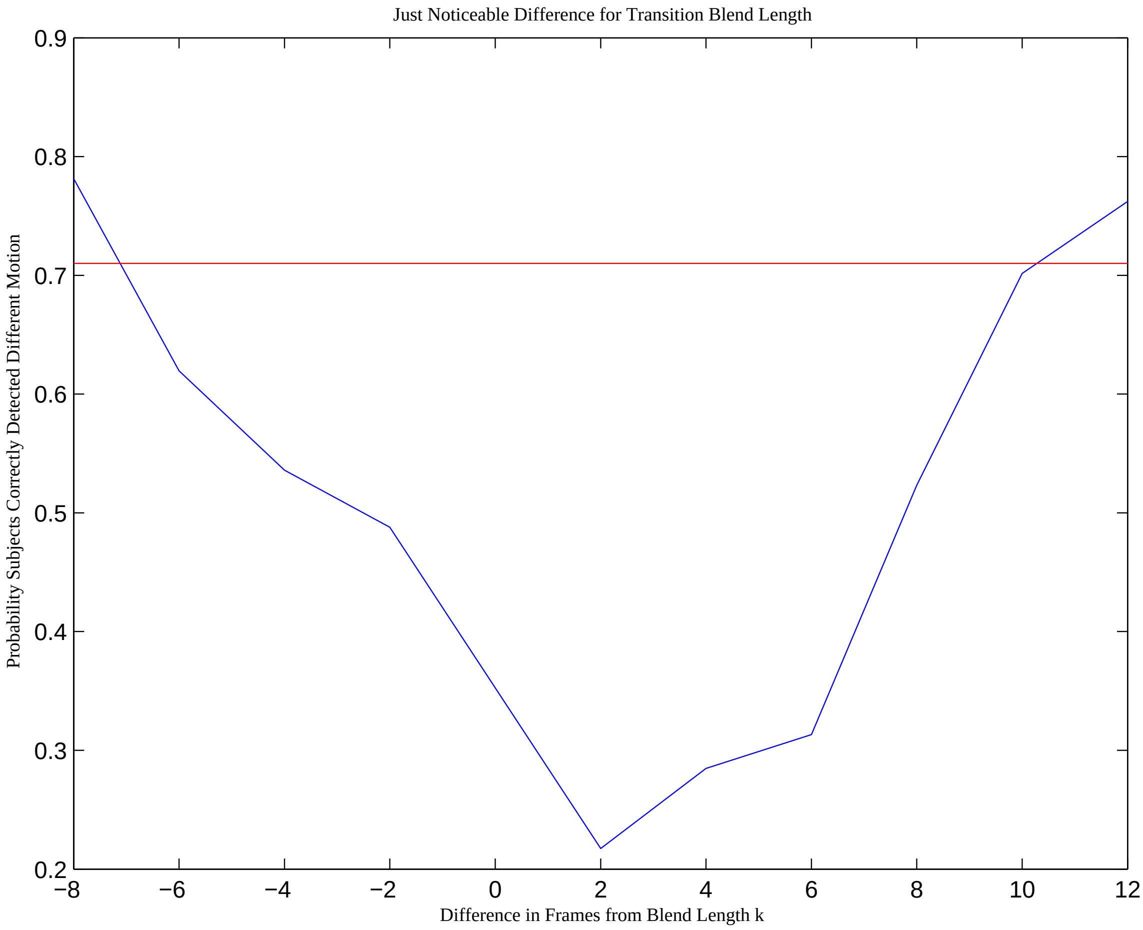


Figure VI.5: Results of the just noticeable difference adaptive study for center-aligned transitions. The *x*-axis shows the difference in blend length from 10 frame. The *y*-axis shows the probability of successfully detecting the motion pair containing the different motion.

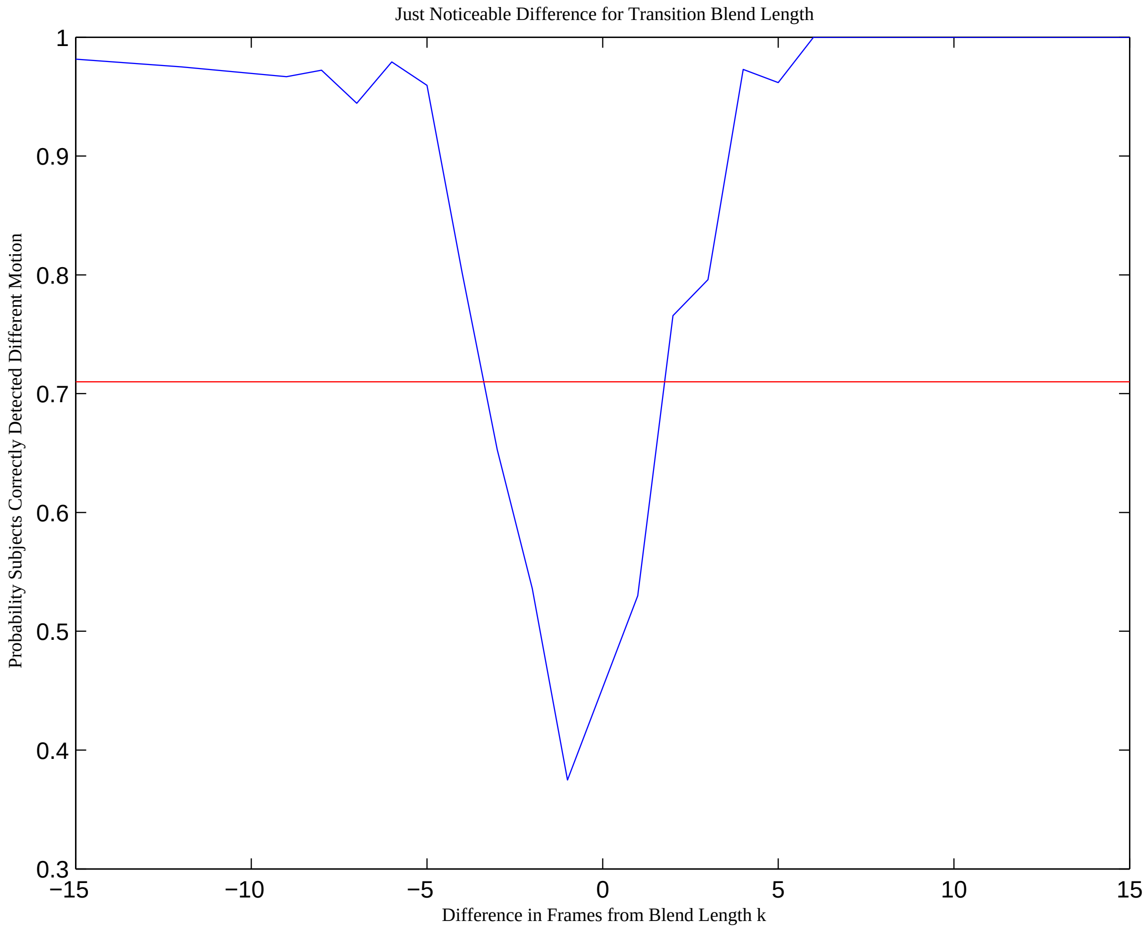


Figure VI.6: Results of the just noticeable difference adaptive study for start-end transitions. The x-axis shows the difference in blend length from the optimal value k computed using the geodesic distance method. The y-axis shows the probability of successfully detecting the motion pair containing the different motion.

a guideline for developing transition methods and comparing results by changing blend length. But note that the just noticeable difference experiment does not inform of us of the difference in visual appeal, which in general is a difficult quantity to measure without bias.

The results of the studies showed that it is important which specification you use to generate motion transitions. People are more sensitive to the changes in blend length of the start-end transitions than the center-aligned transitions. This finding makes sense because of the inherent phase component or alignment in the start-end transitions. Remember that the start frame of transition in the *from* motion and end frame for transition in the *to* motion are specified when creating a start-end transition. Therefore, the correspondence of frames of two motion segments for blending is shifted if the blend length is made shorter or longer. For cyclic locomotions such as walking and running, each frame in the motion normally corresponds to a certain phase of a cycle, namely, stance phase and swing phase. Consequently, the phase alignment of two motion segments for blending are shifted while changing the blend length. So far, the transition points for the start-end transitions were randomly picked, and the optimal blend length computed by the geodesic distance method was used as the baseline.

Center-aligned transitions are less sensitive to variations in the blend length. However, center-aligned transitions rely at present too heavily upon transition metrics. Many motion generation systems identify similar frames using different transition metrics and create transitions between them. However, there are no guarantees that optimal transition points selected by a method leads to visually appealing transitions regardless of the blend length. The quality of the result motion is directly related to the degree of the phase matching of the whole blending period. If there is a strong correspondence of the major joints such as legs and arms all through the blending, the resulting motion is normally smooth and without hops. Therefore, changing the transition points by changing the alignment, if it can be done in a computationally efficient way, represents a second-pass process that can improve the visual appeal of a transition. The optimal blend length determined by geodesic distance method for start-end transitions maximize the phase matching. For center-aligned transitions, if the poses picked by a distance metric are out of alignment, a longer blend length sometimes helps but there is often still artifacts in the resulting motion.

CHAPTER VII

SUMMARY AND FUTURE WORK

In this thesis we focused on three problems involved in character animation generation using motion capture. First, we improved a cost metric for pose comparison by optimizing weights used in its computation. Second, we developed methods to compute a good duration for blending when creating a motion transition. Third, we determined the sensitivity of observers to a blend duration by conducting psychophysical studies. These methods and studies provide insights for motion editing and for interactive video games.

New motion can be created by piecing together sequences of motion and generating transitions between them. The visual quality of these transitions is critical to the resulting motion. Several cost metrics have been proposed for pose comparison and determining good transition points. We demonstrated that a cost metric could be improved for determining good transition points by optimizing the parameters involved in the metric using examples of good and bad motion transitions. Prior to this work, such parameters were tuned in an ad hoc manner by animators and game designers. Moreover, we employed a leave-one-out cross validation of the weights to show that the optimized weights are robust and work for a wide variety of behaviors.

Once these transition points are selected, blending is often used to generate smooth transitions among segments of motions. The duration of such a transition is an important factor in the visual appearance of the synthesized motion. Many motion editing systems simply set fixed duration for motion transitions. However, using a fixed duration does not ensure the plausibility of the synthesized motion. We showed that by computing an optimal duration for blending, visually pleasing transitions could be generated without further tuning.

We developed two methods for determining an optimal blend length for motion transitions, the geodesic distance method and the velocity method. These two methods are suited to different types of motion. The geodesic distance method works well for cyclic locomotion such as walking and running; the velocity method works well for other physical activities such as boxing and dancing. These methods give guidance to designers of animation systems who wish to incorporate varying blend lengths into their systems.

All of our results were tested empirically. This testing allows us to have confidence in the robustness and generalizability of our results. We first conducted a user study to evaluate the weights obtained by optimization. The user study demonstrated that the optimized weights select more appealing transition points than the original weights. Second, we empirically evaluated the methods developed to determine the optimal blend length. We found that users prefer the transition method we developed over a generic fixed-length blend. Moreover, we found that users prefer the geodesic distance method over the velocity method for cyclic locomotion; users prefer the velocity method over the geodesic distance method for physical activities. This finding is consistent with our understanding of different transition requirements for different types of motions. Phase correspondence is important for transitions among cyclic locomotion; changes in velocity are important for transitions among physical activities.

Additionally, we conducted studies to determine how noticeable the blend length of a transition was, in other words, the "just noticeable difference" to the changes in blend length. We explored two transition specifications and found that it is important which specification we chose; people are more sensitive to changes in blend length of start-end transitions than the center-aligned transitions because of the inherent phase alignment in the start-end transitions. We concluded that people can differentiate between transition lengths that differ by -7 (i.e., shorter by 7 frames from a reference duration) or 10 frames (i.e., longer by 10 frames from a reference duration) for center-aligned transitions; people can distinguish between transition length that differ by -3 or 2 frames for start-end transitions.

The increasing richness of human characters in three-dimensional computer games should make transitions in motion data an increasingly important problem. We believe that these results give significant guidance to those concerned with creating virtual humans with a rich repertoire of behaviors, and may help in the re-use of large motion capture data-sets.

VII.1 Future Work

This work, while answering several important questions about motion transitions, gives rise to other questions, potentially as interesting. One important issue is the extent to which the methods we developed extend to motions and categories of motions different from those we have tested, especially some highly dynamic motions such as acrobatic gymnastics. We would like to test the methods developed in this work in a larger library with a greater variety of behaviors. For instance, we would like to include some “backward” motions (for example, backward running like a baseball outfielder backing up to catch a ball), which may result in the position-velocity weight having marginal impact for the cost function. Moreover, motions with significant time variations are necessary to investigate the effect of timewarping methods. It is difficult to find examples of such motions in current libraries, but examples of such motion would be dance or floor exercise where the tempo varies. We may also find new categories of motion for which we require different methods for determining duration of transitions. This work has ignored one of the main drawbacks of linear blending, that a linear blend may cause the geometry of the character to intersect with itself. For our motions, such intersections have usually not been a problem. It appears to be a difficult problem, determining if this occurs and what to do about it, but it is an important problem to consider for future work. Determining a fast way of detecting this situation would extend its utility.

Another approach to generating motion transitions uses dynamic simulation to generate motions from one segment to another [Zordan and Hodgins 2002; Rose et al. 1996]. It is particularly useful for behaviors such as multiple player sport motions, for example, football games, for which we need to create physics based transitions between motion capture segments to simulate the physical contact between players. We would like to test and incorporate our methods into such dynamical simulation systems that produce streams of animation. These types of systems typically impose torque constraints that will affect the duration of transitions, but it is likely that leeway exists in picking the transition.

Another interesting avenue to explore would be the evaluation of the naturalness of synthesized human motion. Much of the research in motion capture has focused on techniques for adapting existing data to new situations. They propose hypothesis on creating natural-looking motion, but a measure of *naturalness* does not exist to assess the quality of the resulting motion. The evaluation of naturalness of human motion is a difficult problem because humans are familiar with these motions and very sensitive to anything unusual. Also note that the just noticeable difference study we did measured how sensitive people are to changes in transitions, but it doesn't tell us about the difference in visual appeal. To assess the visual appeal, one possible method is to label example motions based on their naturalness manually, and train classifiers to distinguish between natural and unnatural motions based on these examples.

VII.2 New Directions

In this section we discuss interesting novel directions of research that represent an outgrowth of the current work, but for which only preliminary results are available. Human motion is usually highly correlated, as shown in Figure III.9. One approach to manipulating human motion data is to characterize it using a model with a lower dimensionality than the intrinsic dimensionality of the data (the *configuration space* of the body). The lower-dimensional space describes a lower-dimensional manifold of the configuration space. It also provides a geometric structure of the motion data that defines a controllable surface by which new motion data can be generated.

Principle component analysis (PCA) and multidimensional scaling (MDS) are classical techniques for dimensionality reduction. PCA and MDS discover the structure of data lying on or near a linear subspace of the high-dimensional input space and thus reduce the dimension of the input data. Tenenbaum et al. introduced Isometric feature mapping (Isomap) [Tenenbaum et al. 2000] as a nonlinear dimensionality reduction technique. Isomap provides a way to discover the essential nonlinear structure that are invisible to PCA. Isomap combines the major algorithmic features of PCA and MDS to learn a broad class of nonlinear manifolds. The Isomap algorithm has three steps: (1) Construct the neighborhood graph. (2) Compute the shortest paths. (3) Construct d-dimensional embedding.

The Isomap algorithm (Isomap algorithm from http://isomap.stanford.edu) takes as input the distance between all pairs of data points in the high-dimensional input space. We also need a distance metric that measures the dissimilarity between motion frames. We used the distance metric proposed in Chapter 2, which is based on joint orientations. We ran Isomap with 7 neighbors/point. Figure VII.2 shows the residual variance of a 120 frame walking motion shown in Figure VII.1. We pre-select the number of dimensions in which to embed the data, from one to 10 dimensions. We usually select the "knee" of the curve as the dimensionality of the embedding space. In this example, the "knee" occurs at three dimensions and the three dimensional Isomap embedding of the motion is shown in Figure VII.3. The intrinsic cyclic property of the motion is captured and shown as three circles by the Isomap algorithm. We also explored other types of motion, for example, dancing. Figure VII.5 shows the residual variance of a 700 frame hiphop dancing motion shown in Figure VII.4 for dimension one to 10. For this motion, it is not as clear what the dimension of the embedding should be. We selected five dimensions for training purpose (see next section). To have an idea of what the embedding space looks like, the three dimensional Isomap embedding of the motion is shown in Figure VII.6. It is difficult to discern any structure in this figure. It is probably because of the lack of correlation between joints of the motion. A small number of degree of freedom is not sufficient to describe such an animation behavior.

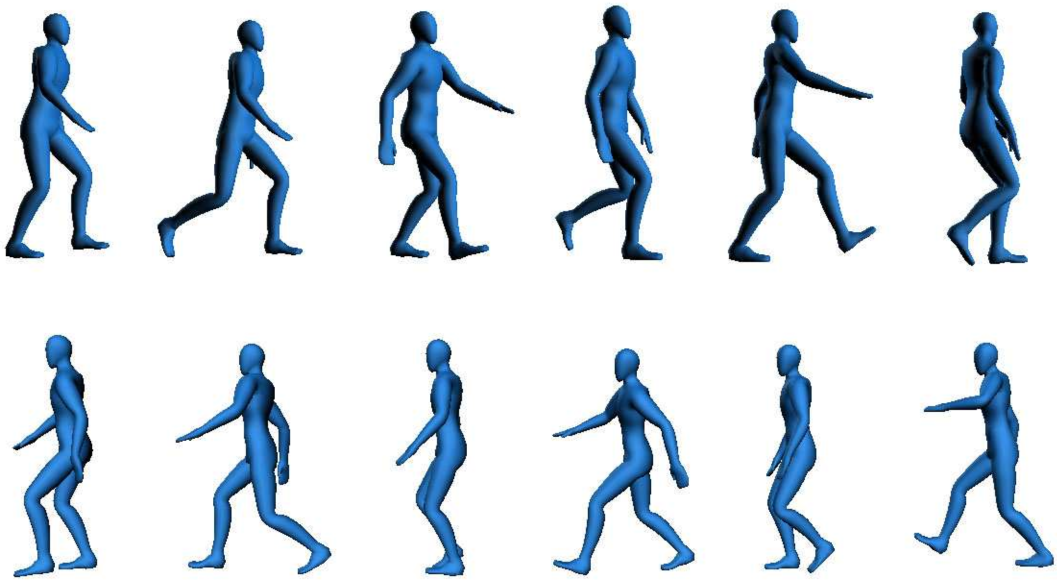


Figure VII.1: The walking motion used for dimensionality reduction.

VII.2.1 Neural Networks

After obtaining the low dimensional manifold of motions, we could potentially estimate new poses and thus, transitions, in this space. The important issue here is how to re-construct the motion data in the space of the original data given the poses in the low dimensional manifold. Re-construction of animation data must also deal with the global position and orientation of the body in space, as well as contact-constraint information such as avoiding foot-slide.

We explored one of the popular machine learning techniques, neural networks, to estimate motion data in the space of the original data. Neural networks provided a general, practical method for learning real-valued, discrete-valued, and vector-valued functions from examples [Mitchell 1997]. It has been successfully applied to problems such as speech recognition, face recognition, and learning robot control strategies. We tried the BACKPROPAGATION algorithm in our work. The BACKPROPAGATION algorithm is

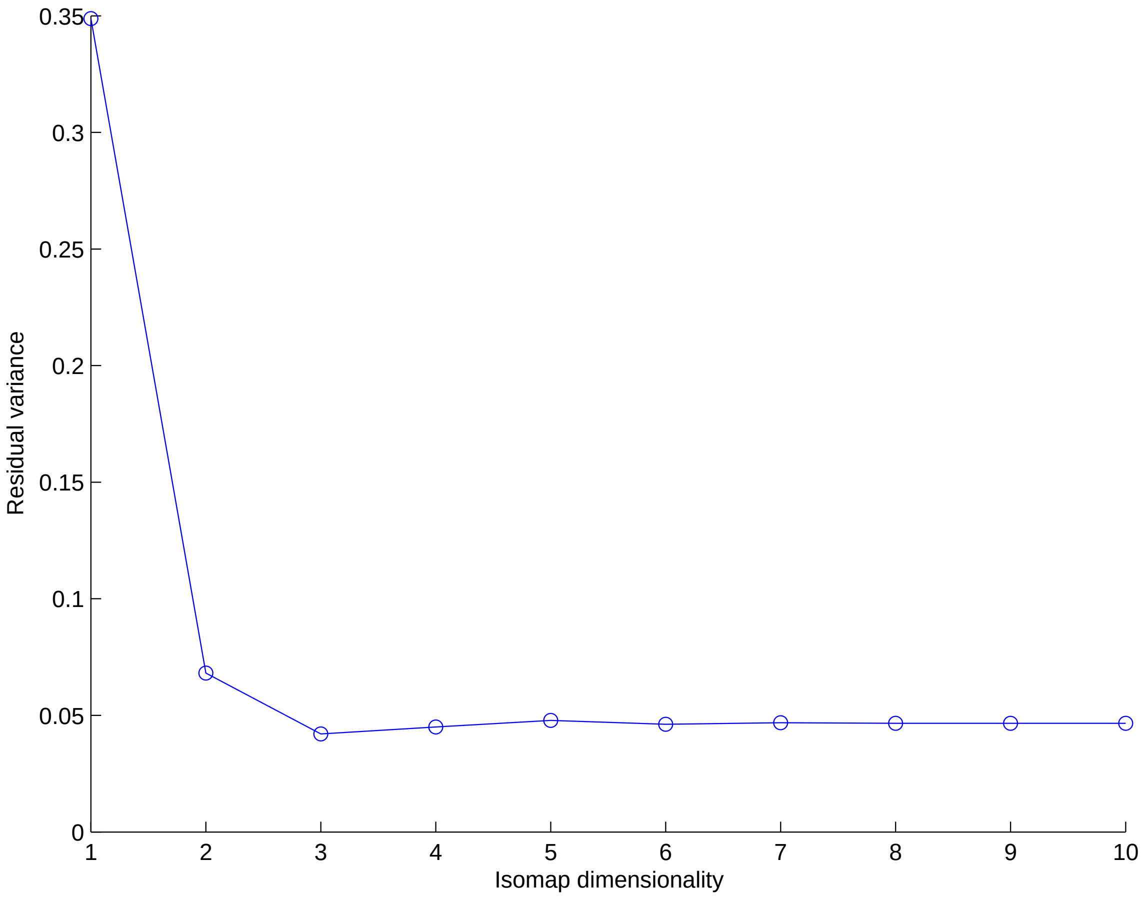


Figure VII.2: The residual variance between a full-dimensional walking motion and the corresponding Isomap dimensionality.

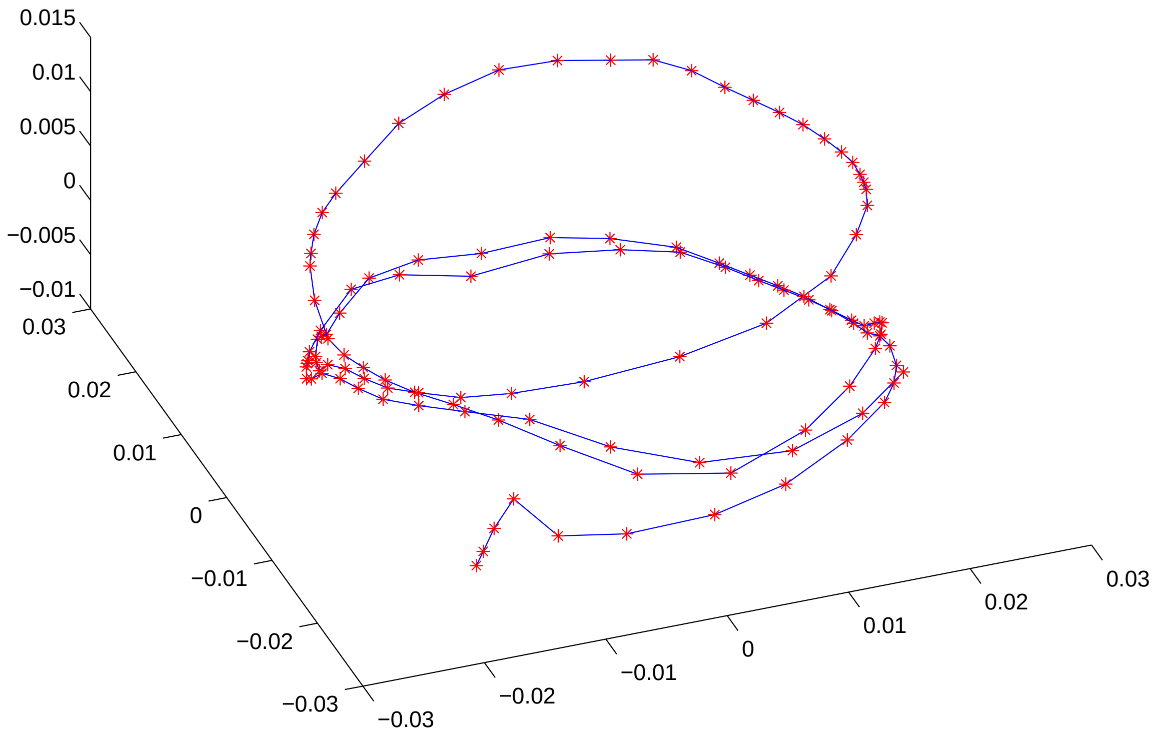


Figure VII.3: Three-dimensional Isomap embedding of a walking motion.

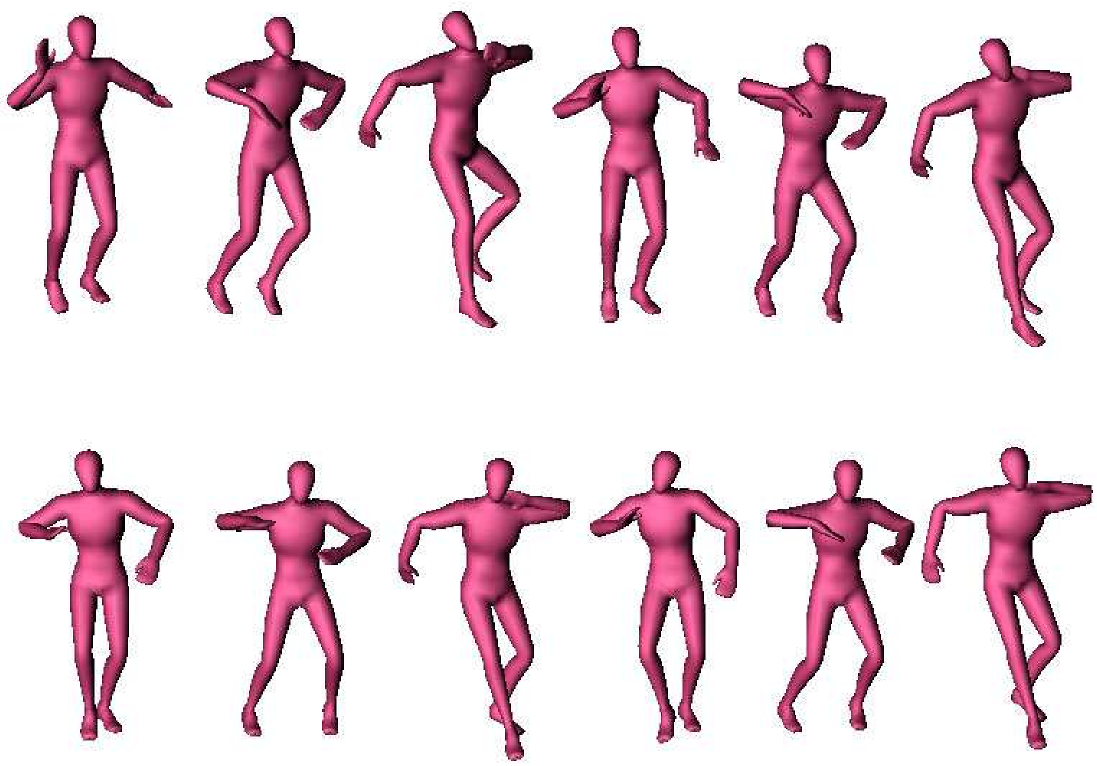


Figure VII.4: The dancing motion used for dimensionality reduction.

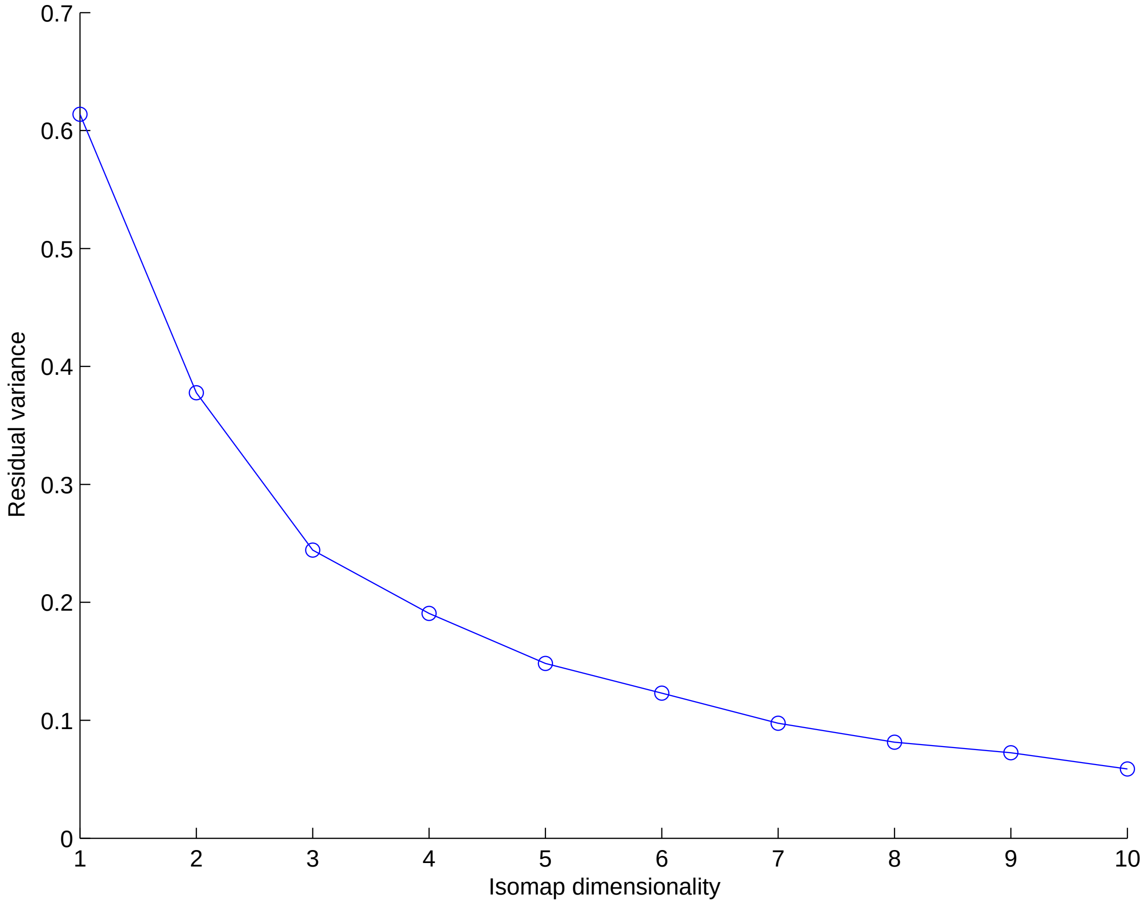


Figure VII.5: The residual variance between a full-dimensional hiphop dancing motion and the corresponding Isomap dimensionality.

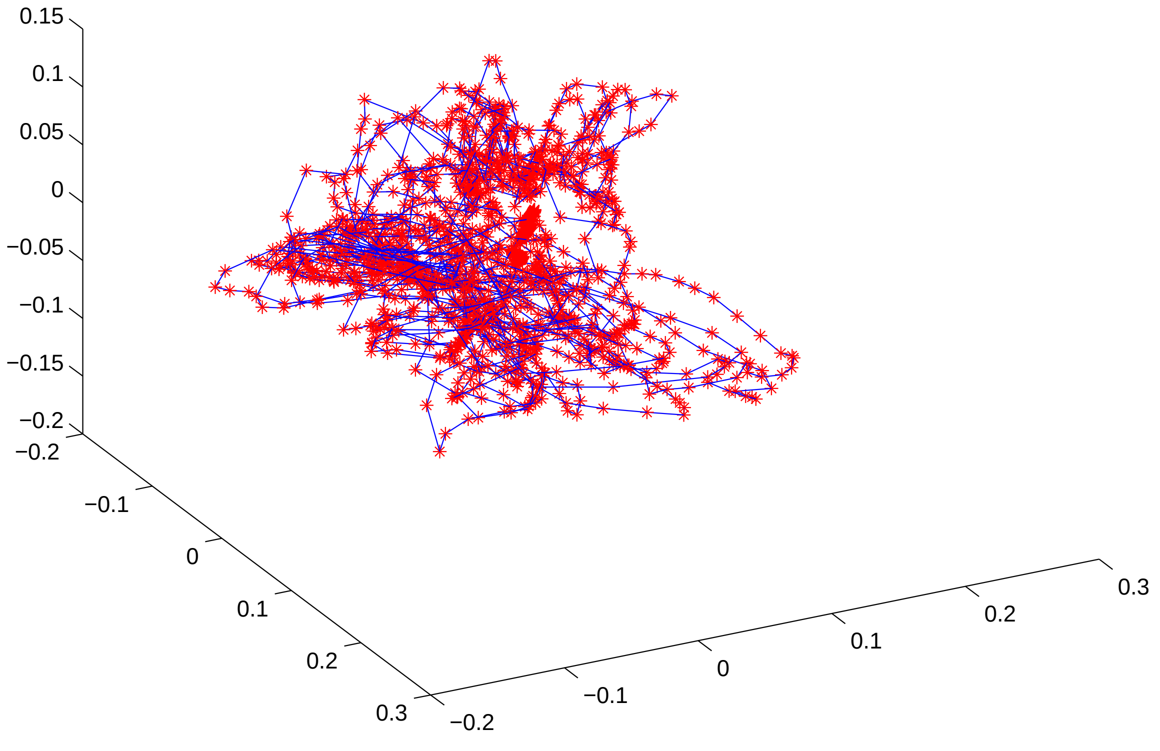


Figure VII.6: Three-dimensional Isomap embedding of a hiphop dancing motion.

the most commonly used Neural Networks learning technique. It employs gradient descent to attempt to minimize the squared error between the network output values and the target values for these outputs. It is appropriate for problems where interpreting the learned target function is not important and long training times are acceptable.

The termination condition for the BACKPROPAGATION algorithm needs to be specified by the user, for example, how many epochs are needed or what the error threshold of the training should be. However, since the BACKPROPAGATION algorithm has been shown to suffer from overfitting, an early stopping technique has been used to improve generalization. In this technique the available data is divided into three subsets, the training set, the validation set, and the testing set. The validation set is used to monitor the training process. When the network begins to overfit the data, the error on the validation set will typically begin to rise. The training is stopped if the validation error increases for a specified number of iterations. The testing set is used for comparing models, not for training purpose.

VII.2.2 Motion Synthesis

The parameterized low dimensional space obtained by neural networks allows us to estimate natural in-between frames. The input of the network is a low-dimensional Isomap embedding of the motion. The target is then the full dimensional motion except the global translation and orientation for training. The learned network is a parameterized space of motion. We can specify a arbitrary curve in low dimension, and generate the corresponding sequence of motion.

For the walking motion we used, the input is the three dimensional embedding of Isomap. For the hiphop dancing motion, the input is the five dimensional embedding of Isomap. We applied this method on motion re-sequencing whereas frames of motion were randomly picked and linear interpolated in the low dimensional Isomap embedding space, and then projected onto full dimensional motion space using the trained network. The current results need further improvement as of tuning the dimensionality reduction and training the network. This is the subject of on-going work.

REFERENCES

[Arikan and Forsyth 2002]ARIKAN, O., AND FORSYTH, D. A. 2002. Synthesizing constrained motions from examples. *ACM Transactions on Graphics 21*, 3 (July), 483–490. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[Arikan et al. 2003]ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2003. Motion synthesis from annotations. *ACM Transactions on Graphics 22*, 3 (July), 402–408.

[Bodenheimer et al. 1999]BODENHEIMER, B., SHLEYFMAN, A. V., AND HODGINS, J. K. 1999. The effects of noise on the perception of animated human running. In *Computer Animation and Simulation '99*, Springer-Verglag, Wien, N. Magnenat-Thalmann and D. Thalmann, Eds., 53–63. Eurographics Animation Workshop.

[Bowden 2000]BOWDEN, R. 2000. Learning statistical models of human motion. In *IEEE Workshop on Human Modelling, Analysis, and Synthesis*. CVPR 2000.

[Brand and Hertzmann 2000]BRAND, M., AND HERTZMANN, A. 2000. Style machines. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 183–192. ISBN 1-58113-208-5.

[Bruderlin and Williams 1995]BRUDERLIN, A., AND WILLIAMS, L. 1995. Motion signal processing. In *Computer Graphics*, 97–104. Proceedings of SIGGRAPH 95. Held in Los Angeles, CA.

[Dontcheva et al. 2003]DONTCHEVA, M., YNGVE, G., AND POPOVIĆ, Z. 2003. Layered acting for character animation. *ACM Transactions on Graphics 22*, 3 (July), 409–416.

[Duda et al. 2001]DUDA, R. O., HART, P. E., AND STORK, D. G. 2001. *Pattern Classification*, second ed. John Wiley and Sons.

[Galata et al. 2001]GALATA, A., JOHNSON, N., AND HOGG, D. 2001. Learning variable length markov models of behaviour. *Computer Vision and Image Understanding Journal 81*, 3 (Mar.), 398–413.

[Gill et al. 1981]GILL, P. E., MURRAY, W., AND WRIGHT, M. H. 1981. *Practical Optimization*. Academic Press.

[Gleicher 2001]GLEICHER, M. 2001. Comparing constraint-based motion editing methods. *Graphical Models 63*, 2 (March), 107–134. ISSN 1524-0703.

[Grassia 2000]GRASSIA, F. S. 2000. *Believable Automatically Synthesized Motion by Knowledge-Enhanced Motion Transformation*. Ph.d. dissertation, Carnegie Mellon University. CMU-CS-00-163.

[Harrison et al. 2004]HARRISON, J., RENSINK, R. A., AND VAN DE PANNE, M. 2004. Obscuring length changes during animated motion. *ACM Transactions on Graphics 23*, 3 (August), 569–573. Proceedings of ACM SIGGRAPH 2004.

[Hayter 1986]HAYTER, A. J. 1986. *Probability and statistics for engineers and scientists*. PWS Pub.Co., Boston.

[Hita et al. 1984]HITA, E., ROMERO, J., DEL BARCO, L. J., AND MARINEZ, R. 1984. Study of weber's law. two experiments in physiological optics. *European Journal of Physics 5* (July), 135–138.

[Hodgins et al. 1998]HODGINS, J. K., O'BRIEN, J. F., AND TUMBLIN, J. F. 1998. Judgments of human motion with different geometric models. *IEEE Transactions on Visualization and Computer Graphics 4*, 4.

[Johansson 1973]JOHANSSON, G. 1973. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics 14*, 201–211.

[Jolliffe 1986]JOLLIFFE, I. 1986. *Principle Component Analysis*. Springer-Verlag, New York.

[Kovar and Gleicher 2003]KOVAR, L., AND GLEICHER, M. 2003. Flexible automatic motion blending with registration curves. In *Symposium on Computer Animation 2003*, D. Breen and M. Lin, Eds., ACM SIGGRAPH/Eurographics, 214–224.

[Kovar et al. 2002a]KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics 21*, 3 (July), 473–482. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[Kovar et al. 2002b]KOVAR, L., SCHREINER, J., AND GLEICHER, M. 2002. Footskate cleanup for motion capture editing. In *ACM SIGGRAPH Symposium on Computer Animation*, 97–104.

[Kovar et al. 2004]KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2004. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics 23*, 3 (August), 557–566. Proceedings of ACM SIGGRAPH 2004.

[Lamouret and van de Panne 1996]LAMOURET, A., AND VAN DE PANNE, M. 1996. Motion synthesis by example. In *EGCAS '96: Seventh International Workshop on Computer Animation and Simulation*, Eurographics. ISBN 3-211-82885-0.

[Lasseter 1998]LASSETER, J. 1998. Principles of traditional animation applied to 3d computer animation. 263–272.

[Lee and Shin 1999]LEE, J., AND SHIN, S. Y. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, 39–48.

[Lee and Shin 2002]LEE, J., AND SHIN, S. Y. 2002. General construction of time-domain filters for orientation data. *IEEE Transactions on Visualization and Computer Graphics 8*, 2 (April - June), 119–128. ISSN 1077-2626.

[Lee et al. 2002]LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics 21*, 3 (July), 491–500. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[Leek 2001]LEEK, M. R. 2001. Adaptive procedures in psychophysical research. *Perception & Psychophysics 63(8)*, 1279–1292.

[Li et al. 2002]LI, Y., WANG, T., AND SHUM, H.-Y. 2002. Motion texture: A two-level statistical model for character motion synthesis. *ACM Transactions on Graphics 21*, 3 (July), 465–472. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

[Mitchell 1997]MITCHELL, T. M. 1997. *Machine Learning*. McGraw-Hill.

[Mizuguchi et al. 2001]MIZUGUCHI, M., BUCHANAN, J., AND CALVERT, T. 2001. Data driven motion transitions for interactive games. *Eurographics 2001 Short Presentations*.

[Murray et al. 1994]MURRAY, R., LI, Z., AND SASTRY, S. S. 1994. *Robotic Manipulation*. CRC Press.

[Oesker et al. 2000]OESKER, M., HECHT, H., AND JUNG, B. 2000. Psychological evidence for unconscious processing of detail in real-time animation of multiple characters. *The Journal of Visualization and Computer Animation 11*, 2 (June), 105–112.

[O'Sullivan et al. 2003]O'SULLIVAN, C., DINGLIANA, J., GIANG, T., AND KAISER, M. K. 2003. Evaluating the visual fidelity of physically based animations. *ACM Transactions on Graphics 22*, 3 (July), 527–536.

[Park et al. 2002]PARK, S. I., SHIN, H. J., AND SHIN, S. Y. 2002. On-line locomotion generation based on motion blending. In *ACM SIGGRAPH Symposium on Computer Animation*, 105–112.

[Perlin 1995]PERLIN, K. 1995. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (Mar.), 5–15.

[Popović et al. 2000]POPOVIĆ, J., SEITZ, S. M., ERDMANN, M., POPOVIĆ, Z., AND WITKIN, A. 2000. Interactive manipulation of rigid body simulations. In *Proceedings of SIGGRAPH 2000*, ACM SIGGRAPH, 209–218. Held in New Orleans, Louisiana.

[Pullen and Bregler 2000]PULLEN, K., AND BREGLER, C. 2000. Animating by multi-level sampling. In *Computer Animation 2000*, IEEE CS Press, 36–42. ISBN 0-7695-0683-6.

[Pullen and Bregler 2002]PULLEN, K., AND BREGLER, C. 2002. Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics 21*, 3 (July), 501–508.

[Reitsma and Pollard 2003]REITSMA, P. S. A., AND POLLARD, N. S. 2003. Perceptual metrics for character animation: Sensitivity to errors in ballistic motion. *ACM Transactions on Graphics* (July). Proceedings of SIGGRAPH 2003, to appear.

[Reitsma and Pollard 2004]REITSMA, P. S. A., AND POLLARD, N. S. 2004. Evaluating motion graphs for character navigation. *Symposium on Computer Animation* (August), 89–98.

[Ren et al. 2005]REN, L., PATRICK, A., EFROS, A., HODGINS, J. K., AND REHG, J. 2005. A data-driven approach to quantifying natural human motion. *ACM Transactions on Graphics* (July). Proceedings of SIGGRAPH 2005.

[Rose et al. 1996]ROSE, C. F., GUENTER, B., BODENHEIMER, B., AND COHEN, M. F. 1996. Efficient generation of motion transitions using spacetime constraints. In *Proceedings of SIGGRAPH 96*, ACM SIGGRAPH / Addison Wesley, New Orleans, Louisiana, Computer Graphics Proceedings, Annual Conference Series, 147–154. ISBN 0-201-94800-1.

[Rose et al. 1998]ROSE, C., COHEN, M., AND BODENHEIMER, B. 1998. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications 18*, 5, 32–40.

[Safonova and Hodgins 2005]SAFONOVA, A., AND HODGINS, J. K. 2005. Analyzing the physical correctness of interpolated human motion. In *Symposium on Computer Animation 2005*, ACM SIGGRAPH/Eurographics.

[Safonova et al. 2004]SAFONOVA, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics 23*, 3 (August), 512–519. Proceedings of ACM SIGGRAPH 2004.

[Schödl et al. 2000]SCHÖDL, A., SZELISKI, R., SALESIN, D. H., AND ESSA, I. 2000. Video textures. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 489–498. ISBN 1-58113-208-5.

[Shoemake 1985]SHOEMAKE, K. 1985. Animating rotation with quaternion curves. *ACM Transactions on Graphics 19*, 3 (July), 245–254.

[Sidenbladh et al. 2002]SIDENBLADH, H., BLACK, M. J., AND SIGAL, L. 2002. Implicit probabilistic models of human motion for synthesis and tracking. In *Computer Vistion — ECCV 2002 (1)*, Springer-Verlag, Copenhagen, Denmark, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., Lecture Notes in Computer Science, 784–800. 7th European Conference on Computer Vision.

[Tenenbaum et al. 2000]TENENBAUM, J., SILVA, V., AND LANGFORD, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science 290(5500)*, 2319–2323.

[Wetherill and Levitt 1965]WETHERILL, C. B., AND LEVITT, H. 1965. Sequential estimation of points on a psychometric function. *British Journal of Mathematical and Statistical Psychology 18*, 1–10.

[Wiley and Hahn 1997]WILEY, D. J., AND HAHN, J. K. 1997. Interpolation synthesis for articulated figure motion. In *Proceedings of the Virtual Reality Annual International Symposium*, IEEE Computer Society Press, 157–160.

[Witkin and Popović 1995]WITKIN, A. P., AND POPOVIĆ, Z. 1995. Motion warping. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 105–108.

[Zhao and Badler 1994]ZHAO, J., AND BADLER, N. I. 1994. Inverse kinematics positioning using non-linear programming for highly articulated figures. *ACM Trans. Gr. 13*, 4 (Oct.), 313–336.

[Zordan and Hodgins 2002]ZORDAN, V. B., AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH Symposium on Computer Animation*, 89–96.

[Zordan et al. 2005]ZORDAN, V., MAJKOWSKA, A., CHIN, B., AND FAST, M. 2005. Dynamic response for motion capture animation. In *ACM Transactions on Graphics*. Proceedings of ACM SIGGRAPH 2005.