

**MOTION CAPTURE BASED
MOTION ANALYSIS AND MOTION
SYNTHESIS FOR HUMAN-LIKE
CHARACTER ANIMATION**

ZHIDONG XIAO

July 2009

National Centre for Computer Animation

Bournemouth University

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

**MOTION CAPTURE BASED MOTION ANALYSIS
AND MOTION SYNTHESIS FOR HUMAN-LIKE
CHARACTER ANIMATION**

ZHIDONG XIAO

*A thesis submitted in partial fulfilment of the requirements of the Media
School of Bournemouth University for the degree of Doctor of Philosophy*

July 2009

National Centre for Computer Animation

Bournemouth University

© Copyright 2009 ZHIDONG XIAO

All Rights Reserved

ABSTRACT

Motion capture technology is recognised as a standard tool in the computer animation pipeline. It provides detailed movement for animators; however, it also introduces problems and brings concerns for creating realistic and convincing motion for character animation.

In this thesis, the post-processing techniques are investigated that result in realistic motion generation. A number of techniques are introduced that are able to improve the quality of generated motion from motion capture data, especially when integrating motion transitions from different motion clips. The presented motion data reconstruction technique is able to build convincing realistic transitions from existing motion database, and overcome the inconsistencies introduced by traditional motion blending techniques. It also provides a method for animators to re-use motion data more efficiently.

Along with the development of motion data transition reconstruction, the motion capture data mapping technique was investigated for skeletal movement estimation. The per-frame based method provides animators with a real-time and accurate solution for a key post-processing technique.

Although motion capture systems capture physically-based motion for character animation, no physical information is included in the motion capture data file. Using the knowledge of biomechanics and robotics, the relevant information for the captured performer are able to be abstracted and a mathematical-physical model are able to be constructed; such information is then applied for physics-based motion data correction whenever the motion data is edited.

PUBLICATIONS RESULTING FROM THESIS

During the period of research, several publications have been produced to report the technical developments achieved, which are listed as below:

- Xiao, Z., Nait-Charif, H. and Zhang, J.J., 2009, Incomplete motion data reconstruction for character animation. *In Proceeding of IADIS Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP 2009) Conference*, pp 89 – 98.
- Xiao, Z., Nait-Charif, H. and Zhang, J. J., 2009, Real time automatic skeleton and motion estimation for character animation. *Journal of Computer Animation and Virtual Worlds*, 20(5-6), pp 523 – 531. John Wiley & Sons, Ltd, ISSN: 1546-427X (online), ISSN: 1546-4261 (print), DOI: 10.1002/cav.277.
- Xiao, Z., Nait-Charif, H. and Zhang, J.J., 2008, Automatic estimation of skeleton motion from optical motion capture data. *Lecture Notes in Computer Science. Motion in Games 2008*, pp 144 - 153. Springer: Berlin-Heidelberg, ISBN: 978-3-540-89219-9, ISSN: 0302-9743, DOI: 10.1007/978-3-540-89220-5.

ACKNOWLEDGMENT

Foremost, I would like to thank my supervisors, Professor Jian J. Zhang and Dr. Hammadi Nait-Charif. Professor Zhang has guided me throughout this excited research project. I sincerely appreciate all the help and support that Professor Zhang has provided me with during my Ph.D studies. Dr. Nait-Charif has given me lots of valuable feedbacks on my research, and it has been extremely enjoyable to have discussions with him.

I would like to thank Professor John Vince, the former head of National Centre for Computer Animation, for his support and help all the time even after his retirement. My sincere appreciation goes to Mrs. Jan Lewis and Mrs. Joyce Power for their kindly daily help during my studies. I would also like to thank Dr. Stephen Bell, Gordon Cooper, Andrew Cousins and a number of Chinese colleagues from Computer Graphics and Visualisation Research Group, for their support and help during my research.

Finally, and most importantly, I would like to thank my family. I am very grateful to my wife, Ying, for her day-to-day care and encouragement, and to my lovely son, Pengyi, for bringing me the happiness outside of my studies. I would also like to thank my sister and brother-in-law, Zhiying and Mark. Their support accompanies with me all the time. Without the love and help from my parents, I would not have finished my studies.

DECLARATION

The research described in this thesis was carried out by the author. The work is original and the work of others has been acknowledged in the text and reference. No material contained in this thesis has been used in any other submission for an academic award.

Zhidong Xiao

LIST OF CONTENTS

Abstract	iv
Publications Resulting from Thesis	v
Acknowledgement	vi
Declaration	vii
List of Contents	viii
List of Figures	xi
List of Tables	xvi
Chapter 1 Introduction	1
1.1 Overview	3
1.2 Methodology.....	4
1.3 Contribution	6
1.4 Thesis layout	7
Chapter 2 Related Work	9
2.1 Motion editing for character animation	9
2.1.1 Keyframing	10
2.1.2 Inverse kinematics control	11
2.2 Physics-based motion generation	13
2.3 Motion transition	18
2.4 Motion Synthesis from motion capture	20
2.4.1 Human-like skeleton mapping from motion capture data	21
2.4.2 Motion capture based motion synthesis	22
2.5 Summary	24
Chapter 3 Estimation of Skeletal Movement	26
3.1 Introduction	26
3.2 Motion capture and skeleton fitting	29

3.2.1 Marker identification and joint determination	30
3.2.2 Joint angle calculation	33
3.3 Implementation and results	36
3.4 Discussion	43
3.5 Summary	44
Chapter4 Motion Data Analysis	46
4.1 Introduction	46
4.2 Motion data acquisition and analysis	48
4.2.1 Principal component analysis (PCA)	49
4.2.2 Motion construction from motion database	53
4.2.3 Cyclic motion synthesis	54
4.3 Transition from walking to running	60
4.4 Experiment results	62
4.5 Discussion	64
4.6 Summary	65
Chapter 5 Motion Transition by Incomplete Motion Data Reconstruction	67
5.1 Introduction	67
5.2 Problem description	69
5.3 Incomplete motion data reconstruction	72
5.3.1 Motion analysis from given motion sequence	73
5.3.1.1 Motion data interpretation	73
5.3.1.2 Motion path design and analysis	75
5.3.2 Motion reconstruction	75
5.3.3 Time alignment.....	76
5.4 Experiment results	78
5.4.1 Walking along a curve path.....	78
5.4.2 Sprint.....	82
5.5 Discussion	84
5.6 Summary	85
Chapter 6 Motion Data Correction	86
6.1 Introduction	86
6.2 Human character model	89
6.2.1 Physical properties of human character	91
6.2.2 Joint limitation of human character.....	95
6.3 Physics-based human character.....	97
6.3.1 Dynamic balance constraint.....	98
6.3.2 Ground contact force.....	100
6.4 Constrained optimization method.....	101

6.4.1 Cubic B-spline	101
6.4.2 Objective function.....	103
6.5 Experiment results.....	105
6.5.1 Foot Slipping correction	106
6.5.2 Stair walking	107
6.6 Discussion.....	108
Chapter 7 Conclusion and Future Work	109
7.1 Conclusion	109
7.2 Future work	111
7.2.1 Automatic motion data segmentation and searching	111
7.2.2 Motion data acquisition	112
7.2.3 Physically based motion reconstruction	112
7.2.4 Motion blending in marker space	112
7.2.5 Dynamic reaction with virtual environment.....	113
Appendix	114
Reference	118

LIST OF FIGURES

Figure 1.1	Structure of the thesis	8
Figure 2.1	Keyframing techniques for creating an animation trajectory by setting four key positions of a sphere in a 3D space	11
Figure 2.2	Inverse kinematic handle control for one leg of human character	12
Figure 3.1	The procedure of motion capture data mapping to a human skeleton. (a) Stick figure constructed from markers position; (b) The prototype of human skeleton; (c) Data mapping from markers position to human skeleton in joint space	29
Figure 3.2	Topology of human character and some marker positions shown with black dot	30
Figure 3.3	Human character joint position calculation at knee joint...	32
Figure 3.4	Hierarchical chain	34
Figure 3.5	Coordinated transformation of linked system. P is the current joint end position in i^{th} joint coordinate, P^* is the desired end joint.....	36
Figure 3.6	Anthropometric measurements from Winter (1990).....	37
Figure 3.7	(a) Human body segment as fraction of human height, + and x from two male performers, O from a female, are obtained after the mapping procedure by the MotionBuilder software. (b) Based on anthropometry measurement, relative errors (in percentage) of each body segments from three sets of data obtained from the presented method (blue) and MotionBulider output (yellow), from left to right, the columns correspond to the upper arm, forearm, lower leg and thigh respectively	38

Figure 3.8	Column (a) are the raw optical motion data based on each individual marker set captured from Reactor2, Column (b) are the constructed skeleton (green) sphere displays joint, blue bar is the rigid body segment, and yellow dot shows the centre of mass of each rigid body)	40
Figure 3.9	Column (a) are the raw optical motion data based on each individual marker set captured from Vicon system (black square shows the marker position). Column (b) are the constructed skeleton (green sphere displays joint, blue bar is the rigid body segment, and yellow dot shows the centre of mass of each rigid body)	41
Figure 3.10	(a) and (b) show human skeleton display with joint Axis (pink), which (a) is the skeleton constructed from Reactor2 system and (b) is from Vicon system; (c) is the skeleton output from the presented method (green) compares with the skeleton achieved from MotionBuilder (yellow)	42
Figure 3.11	Image sequences of a dog running and jumping with estimated skeleton	42
Figure 4.1	Segmentation and re-sampling of walking and running motion data in joint angle position within a cycle. Variation of angular velocity and acceleration has also shown in the walking cycle	49
Figure 4.2	PCA analysis of human motion: (a) Walking that includes 6 subjects, 5 speeds and 1 cycle; (b) Running that includes 6 subjects, 4 speeds and 1 cycle; (c) Walking that process 180 motions together from (a); (d) Running produce 120 samples together from (b); (e) Mean Square Error between a full-dimensional motion and the corresponding l -dimensional representation for walking and running behaviours	52
Figure 4.3	Walking speed corresponding to the coefficient of the first Principle Component from 6 person's motion data	55
Figure 4.4	Walking speed corresponding to the coefficient of the Second Principle Component from 6 person's motion data	55
Figure 4.5	Running speed corresponding to the coefficient of the first Principle Component from 6 person's motion data	56
Figure 4.6	Running speed corresponding to the coefficient of the Second Principle Component from 6 person's motion data	56
Figure 4.7	Walk cycle: two joints angle comparison from 6 different persons	57

Figure 4.8	Walking frequency function corresponding to normalised speed	58
Figure 4.9	Running Frequency function corresponding to normalized speed	59
Figure 4.10	The analysis of joint angle position, angular velocity and angular acceleration during the transition from walking to running	60
Figure 4.11	Speed variations from walking to running corresponding to the coefficient of the first Principal Component from 6 people's motion data.....	61
Figure 4.12	Speed variations from walking to running corresponding to the coefficient of the second Principal Component from 6 people's motion data	62
Figure 4.13	Principal component analysis of walking motion: (a) Walking motion construction from different motion database; (b) Construct same speed walking motion according to different resources	63
Figure 4.14	Two different size skeletons with identical movement in same speed: (a) Walking cycle; (b) Running cycle; (c) Walking cycle analysis in same walking speed from top left.....	64
Figure 5.1	Story board organizations for 3D animation product from Parent (2002).....	68
Figure 5.2	Pose constraints for employing motion blending	69
Figure 5.3	(Top) Running motion sequences from blending of standby and run motion clips; (Bottom) Motion analysis before and after motion blending from these two motions.	70
Figure 5.4	Translation analysis of root joint after motion blending includes position and velocity.....	71
Figure 5.5	Motion data interpretation from three areas. (a) Computer graphics skeleton display; (b) Motion data file format; (c) Motion data display	74
Figure 5.6	Motion path planned for IMDR. (a) Walking along a direct line; (b) Walking along a curve	75
Figure 5.7	Example of dynamic time warping applied to a walking motion and a running motion.....	77
Figure 5.8	Construct turn around motion, where $Q_1(t)$ is the first motion clip, $Q_2(t)$ is the required construct motion clip, $Q_3(t)$ is the second motion clip, r is radius of circle, θ is the rotation angle covering the missing gap, F_c is the centripetal force and v is velocity	78

Figure 5.9	Joint angles constructions by applying different number of principal components: (Top) 8 PCs; (Bottom) 13 PCs, and joint angle trajectories comparisons between given motion in low speed (blue), original captured motion in high speed (red) and constructed motion in high speed (green)	80
Figure 5.10	Image sequences of IMDR. (a) Is the original motion clip; (b) Search a proper motion for motion construction; (c) Define a proper path in spatial and temporal space; (d) Adjust and retarget input motion for motion reconstruction; (e) Top view of step (d); (f) End of motion data reconstruction	81
Figure 5.11	Sprint motion sequences constructed by employing IMDR method. (Top) Motion sequences of sprint from standby to run; (Bottom) Velocity analysis of Root joint.....	82
Figure 5.12	RootTZ joint comparison from original blended motion sequences and the constructed motion sequences by IMDR method.....	83
Figure 6.1	The structure of motion synthesis by using constrained optimization method	88
Figure 6.2	(a) CG Human character model with skeleton. (b) Skeleton of human character obtained after Mocap mapping.....	89
Figure 6.3	Human character animation based on motion capture data. (a) walking; (b) running; (c) feet penetrate ground; (d) airborne walking.....	90
Figure 6.4	Cylinder with uniform density in a reference frame	93
Figure 6.5	Efficient dynamic constraints of human character's motion (Dynamic balance constraint: ZMP calculation and coulomb's Contact Model)	99
Figure 6.6	Comparison of curve fitting by applying different number of control points of Cubic B-spline	102
Figure 6.7	Example of Maya API node structure.....	104
Figure 6.8	Eliminate foot slip during human character walking motion. (Top) Original walking motion, right foot slip during touch-down period; (Bottom) Comparison of original motion (skeleton with foot print) and corrected motion (skeleton without foot print).....	106
Figure 6.9	Comparison of human character walking upstairs from original motion and corrected motion.....	107

Figure A (a) The procedure of iterative Newton-Euler dynamic formulation, outward and inward process; (b) A single rigid link rotate with velocity ω and acceleration $\dot{\omega}$ due to the moment N acting on it and linear acceleration \dot{v} due to force F ; (c) A single rigid body in force balance due to different sources acting on it 115

LIST OF TABLES

Table 5-1	Obtained weights correspond to the existing data.....	79
Table 5-2	Comparison of IMDR and traditional blending method	84
Table 6-1	Physical properties of human character from a male performer	95
Table 6-2	Configuration of Joints	96

CHAPTER 1

INTRODUCTION

Three-dimensional character animation, especially articulated human-like character animation, is widely used in computer games, the film industry, sports training, biomechanical analysis and various areas of ergonomics. Although research in computer graphics has progressed tremendously over recent years, to create realistic and convincing human-like character animation remains a challenging task in computer animation.

In computer graphics, a human-like character is represented by a system of hierarchical links connected through revolute joints where a root joint contains extra three-dimensional translation information. This creates more than forty degrees of freedom in joint space which must be addressed to achieve realistic movement. The human figure is a complex structure and has been a challenge to animators to create life-like movement. Although keyframing is widely used, it is tedious and time consuming to create believable human motion.

Physical simulation is another approach for creating realistic human character animation. A physical-mathematical model is designed and integrated into a computer-generated human character. The joint trajectories of an articulated skeleton are then computed according to the animator's directions. However, the main limitation of using dynamic simulation is

that human motion is difficult to be expressed comprehensively by a set of nonlinear equations. Furthermore, dynamic simulation relies heavily on the stability of computation; any small perturbations in initial calculation time can affect the results in subsequent computational steps. Animators are often unable to control the movement of an animated character during the simulation process. Consequently, the resulting character animation only represents the achieved motion that is physically plausible, and it may not look realistic, and can appear rather robotic like.

In contrast to dynamic simulation, a spacetime optimisation method has been a promising method for creating realistic human character movement since 1988 (Witkin and Kass 1988). The spacetime optimisation method employs an objective function as a metric of performance. And a set of constraints are defined according to the skeletal movement in the time domain. An optimisation method is able to find the best result among the set of solutions. But this technique is very sensitive to the initial starting position in the computation and it may not converge if the initial guess is far away from the solution. Similar to dynamic simulation, it is difficult to create convincing character movement that involves multiple degrees of freedom.

In recent years, one state-of-the-art technology, motion capture, has been recognised to be a standard tool for creating high-fidelity animations in computer games and film special effects. Motion capture (MoCap) is a technique and a process that digitally records the movements of a live “performer”, such as a human or an animal in a specified 3D volume. The recorded motion data are then used to animate a computer-generated character. MoCap provides a powerful tool for animators to create plausible and realistic motion sequences for an articulated character animation in less time than other techniques. Currently, it is one of the most effective means to achieve realistic actions from a “performer” for computer-generated character animation.

1.1 Overview

A MoCap system is able to record realistic and detailed human movement from a performer in a constrained operating space. Once the MoCap data are processed they can be used to control and drive a computer-generated character. The character's motion reflects the performer's actions during a motion capture session. However, there are problems when MoCap techniques are used to animate human-like characters in the animation production pipeline:

1. An optical MoCap system can play a valuable role in the animation pipeline. However, it only records a set of three-dimensional Cartesian points that describe the trajectories of markers attached to the live performer. The recorded three-dimensional motion data cannot be used for manipulating a character directly because the output format of motion data does not explicitly construct a hierarchical skeleton. Unfortunately, it is not straightforward to construct a skeleton that is driven by joint rotational data rather than position data. In order to simplify the format of motion capture data and make it editable and controllable, the conversion of the markers' Cartesian position data into a hierarchical skeleton movement in joint space is an important post-processing procedure. Such a procedure makes further motion data editing more convenient and shortens the production pipeline of computer animation.

2. Most MoCap data are related to the motion associated with person's every-day activities, and a typical MoCap session can capture a massive amount of motion data from different sources. The fact is that human behaviours are very unique, and everyone can distinguish various movements very easily. To abstract such behaviours and build corresponding patterns that are able to describe each unique movement, it will reduce the burden of capturing and processing similar motion data for different characters.

3. During a MoCap session, although the majority of movements are easily captured, there are still some that are very dynamic or dangerous and cannot be captured due to the spatial limitation of motion capture studio or other reasons. To overcome such limitations, it is necessary to synthesize a continuous, convincing, and realistic human character animation from several short motion clips. Generally, the technique of ‘motion blending’ is used to construct a continuous sequence of realistic motion from two or more clips of motion capture data. It is a very popular technique as it allows an animator to create complex animation from sequences of simple tasks. It provides a way of using available motion data for creating a certain range of animation scenarios. Such a blending technique also limits the animator to generate realistic movement since they depend on available motion data. According to a certain motion analysis, to generate such unavailable motion data from existing motion database for motion blending will help animators to create more convincing character animation.

4. Even though a MoCap system is able to supply high-fidelity motion for character animation, the movement after data post-processing might not satisfy an animator’s direct needs. Certain artefacts might appear after the motion data is adapted to a specific computer-generated environment, i.e. foot slips or ground penetration. Simple kinematic methods can eliminate such ‘errors’, but the properties of the character movement will appear unconvincing, e.g. the centre of mass of a character is out of the supporting area. To guarantee the generated motion is realistic, a physics-based method is required for correcting such artefacts.

1.2 Methodology

According to the above concerns, the techniques in the post-processing procedure of MoCap technology are focused on and studied.

The method that is proposed in this thesis for creating realistic and convincing human movement for human-like character animation is based upon MoCap techniques along with biomechanics, physical information and statistical analysis. Four techniques are investigated as below:

1. Skeletal movement mapping from optical motion capture data

MoCap records realistic and detailed human movement from a specific three-dimensional space. However, the data is in the form of discrete spatial Cartesian points spread in the temporal domain. Mapping motion capture data to an articulated character is a key procedure for animation in the production pipeline. The prototype and anthropometry of the human body is introduced as a standard template for motion data mapping. After the mapping process, the achieved human motion is guaranteed to be physically plausible and realistic for character animation. The character movement can be edited, retargeted and blended for further scene organisation of animation production.

2. Statistically-based Principal Component Analysis of human movement

The principal component analysis method is used for building a motion database from different motion prototypes. According to the result, a generic model can be constructed that can be used for creating different size character animations from the database. This technology helps animators to re-use existing captured motion data efficiently and effectively.

3. Incomplete motion data reconstruction

Due to spatial limitation or other reasons, motion capture systems are unable to capture a long, continuous motion sequence. An alternative is to create such sequences from several short motion clips or to create an intermediate transition between two motion clips. A statistical method, along with an animator's directions can be used to reconstruct the final animation. By using statistical analysis, the prototype of different motions is able to be analysed and abstracted from a motion database. The

statistical-based method will also extend an animator's ability to synthesize motion that is beyond existing motion database.

4. Motion data correction

During a motion capture session, markers are attached to a performer may change their position. As a result, the achieved articulated character movement will appear unrealistic and introduce artefacts such as 'foot slip'. Due to the fact that MoCap data only contains the kinematic information of human movement, joint trajectories in the time domain, it is unable to apply physics-based constraints for motion synthesis. Based upon biomechanical and physical information, the mass distribution of a human character is able to be achieved and a constrained optimization method is applied to correct such artefacts according to different computer-generated environments.

1.3 Contribution

By using MoCap technology, the motion synthesis method provides a general mechanism for creating realistic and physically-based valid motions that meet an animator's demands. In this thesis, three contributions are listed as follows:

- Automatic optical motion capture data mapping for articulated human character animation by applying less marker information from Reactor2 MoCap system. It simplifies the post-processing procedure by using MoCap in the animation production pipeline. Such a mapping approach can also be applied for achieving skeletal movement from other optical MoCap system;
- Incomplete motion data reconstruction technique is able to generate a wide range of similar motion from existing motion database by applying statistically-based method. Through analysing physical properties of existing motion data, it is able to generate more convincing transition for motion synthesis and blending according

to an animator's directions. It also provides a means for animators re-using existing motion data efficiently;

- Physically-based motion data correction mainly focuses on the artefacts of foot slip. By applying physical laws, the presented method provides animators a method for generating human movement that looks more natural and realistic.

1.4 Thesis layout

The structure of this thesis is shown in Figure 1.1. After a brief introduction of the thesis in chapter 1, the next chapter reviews the related techniques for creating an articulated character animation. The technology that generates and creates high-fidelity human character animation is the most focused area.

Chapter 3 provides the details of motion capture data mapping for the articulated human character animation. Having successfully constructed a human skeleton, the presented method is also extended to map a quadruped animal skeleton.

Chapter 4 presents a statistically-based analysis method for abstracting the potential behaviours of different human movements and constructing different motion patterns from a motion database for character animation.

Chapter 5 describes an incomplete motion data reconstruction method for character motion synthesis that is able to be applied for generating continuous movement according to an animator's directions.

Chapter 6 provides details of abstracting physical properties of human being from existing motion capture data and corrects the artefacts of foot slipping according to certain physical constraints.

Chapter 7 provides a summary of the thesis and reviews the potential of future research.

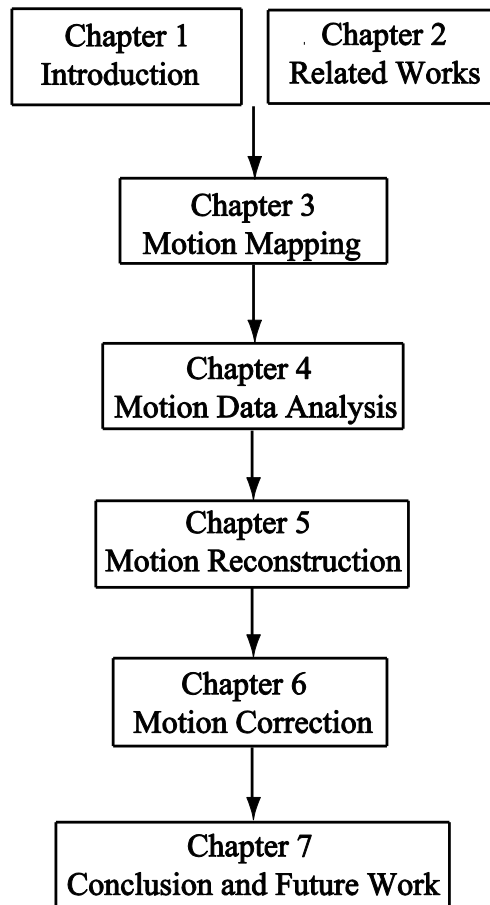


Figure 1.1: Structure of the thesis

CHAPTER 2

RELATED WORK

During the last two decades a number of innovative animations and special effects have been presented in video games, virtual reality, film and television. Behind the aesthetically attractive computer-generated characters, the contributions are not only from the producer (animator), but also from the computer animation techniques utilised. This chapter reviews the techniques for editing and synthesizing human character animation. Firstly, the predominant kinematic-based techniques in the computer graphics industry are reviewed. Subsequently, the motion transition approaches are introduced for creating continuous movement. Then, in order to apply the state-of-the-art technology of motion capture for character animation, the methods for mapping a set of three-dimensional motion capture data to an articulated human character are reviewed. Finally, the techniques for synthesizing realistic motion and re-using an existing motion database efficiently for human-like character animation are focused on.

2.1 Motion Editing for Character Animation

Motion editing and synthesis is the technique used to create motion which controls a character's actions at the appropriate time as realistically as possible. Based on practical experience, the strengths and weaknesses of existing

kinematic-based methods, keyframing and inverse kinematics, are discussed as follows.

2.1.1 Keyframing

In computer animation, the traditional technique is keyframing (Parent 2002) where a number of “key” positions are set at specific times according to the animated sequence. This technique enables animators to control the position of an object easily and produce the desired trajectory of movement. In order to produce a smooth and continuous animation sequence, a cubic spline curve is often used to interpolate the intermediate frames that pass through the set of “key” values supplied by the animator (Parent 2002; Vince 1992). Keyframing, combined with the spline curve technique (Rogers and Adams 1990), is by far the most dominant technique and therefore is at the core of most commercial animation software, such as Autodesk’s Maya, etc (Autodesk 2002). Apart from direct interpolation by cubic spline curve to find the in-between frames, Brotman and Netravali (1988) applied optimal control that models the motion by differential equations. The interpolation is obtained by minimizing control energy and trajectory of the object in space. Combining keyframing and MoCap data, Pullen and Bregler (2002) presented a method to predict the missing degrees of freedom in a given incomplete animation data set.

Keyframing is an efficient approach for controlling position and creating a smooth animation trajectory for a rigid object (Figure 2.1). In comparison, an articulated character has more than forty degrees of freedom (DOFs), therefore it is time-consuming for an animator to set and adjust individual degrees of freedom separately to meet the final postures of an animation production. Furthermore, the quality of character movement may vary tremendously due to the different skills and experience of the animator (Popović 1999).

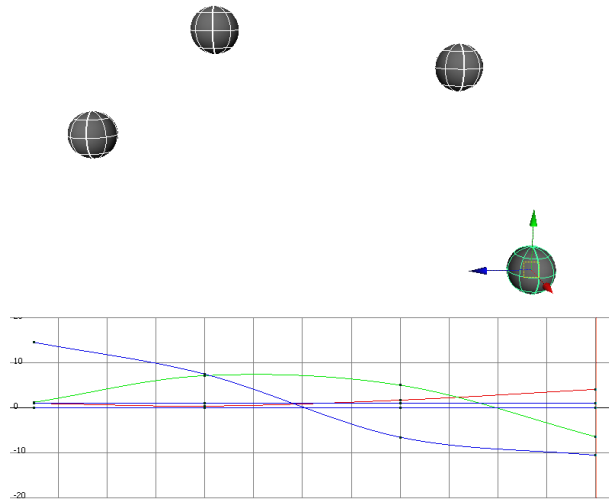


Figure 2.1 Keyframing technique for creating an animation trajectory by setting four key positions of a sphere in a 3D space.

2.1.2 Inverse kinematic control

As keyframing is unable to meet the demands of modern computer-generated character animation, instead of manipulating the DOFs of individual joints of a character directly, the Inverse Kinematics (IK) method has been developed for controlling the postures of an articulated character by positioning the end body part (end effector) of the animated character in three-dimensional space. The intermediate joints within a hierarchical chain will be configured automatically. This technique has been widely investigated in robotics (Craig 1989) and computer animation (Girard and Maciejewski 1985; Boulic and Thalmann 1992; Zhao and Badler 1994; Huang 1997; Monzani et al. 2000; Boulic et al. 2003).

The inverse kinematic approach for a hierarchical linkage is the inverse procedure of forward kinematics. The current state parameters of joints ensure the location of the end joint of a serial chain and the end effector's position and orientation combined with the joints' state in the hierarchy construct a set of non-linear equations. It is possible to invert and obtain the solution for joint angles if the numbers of DOFs in the Joint Space are the same as in the Cartesian Space. However, an articulated structure, such as a

humanoid character, may contain more degrees of freedom in joint parameters which are highly redundant (Huang 1997).

$$\Delta q = J^+ \Delta x + (I - J^+ J) \Delta z \quad (2.1)$$

where Δq are unknown joint variation space vectors; Δx are known positions and orientations of the end effector; J is the Jacobian matrix of the linear transformation; I is the identity matrix ($n \times n$); J^+ is the pseudo-inverse of J ; Δz is the secondary task in the joint variation space which is used to minimize the distance to the attraction posture.

The inverse kinematic method provides animators with an intuitive and friendly tool to control the position of an articulated character. For example, when animating the walk cycle of a human being, instead of controlling and adjusting the hip and knee joint angles explicitly, the animator can define the path of a foot to configure the leg position (Figure 2.2). The intermediate joints within the hierarchical chain can also be adjusted according to the animator's target or the thresholds of each joint. It reduces, to some extent, the burden of creating an articulated character animation.

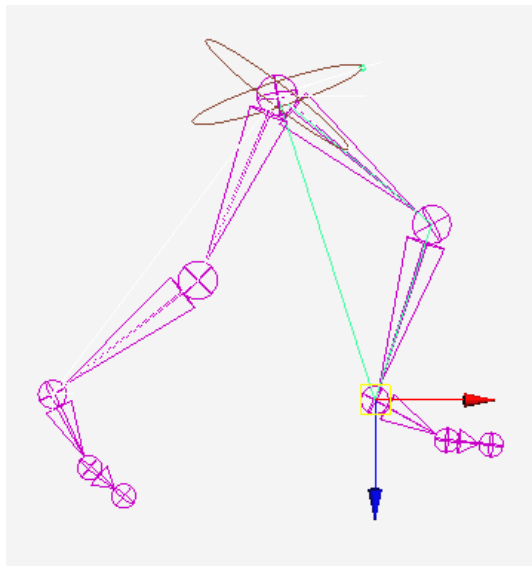


Figure 2.2 Inverse kinematic handle controls for one leg of human character.

2.2 Physics-based Motion Generation

In every-day life people have extensive knowledge and experience of a rigid object moving in space, and have discovered physical laws describing such motions. The human body is constructed from different body parts that are connected by various joints. Although human movement is very complicated, it is still driven by muscle force, ground reaction force, and other internal or external forces (Winter 1990). For a computer-generated character that has more than forty degrees of freedom in joint space, and given the physical properties for each body part, a set of ordinary differential equation can be formed and solved for joint trajectories according to conservation formula governed by external forces and joint torques at individual joints.

The **Forward Dynamics** method (Equation 2.2) had been presented for calculating the spatial parameters of the movement of an articulated character (Armstrong and Green 1985; Mckenna and Zeltzer 1990; Ko 1994).

$$M(q)\ddot{q} + C(q, \dot{q}) = f_{total} \quad (2.2)$$

where q is the generalized parameter, joint angle, \dot{q} , \ddot{q} are joint angular velocity and joint angular acceleration respectively, $M(q)$ is the matrix of inertia, and $C(q, \dot{q})$ includes the centrifugal and Coriolis forces and the damping at the joints in this equation. f_{total} is the total force acting on this system, including muscle force, joint torques and external force.

According to Equation (2.2), a forward dynamic simulation is able to calculate physically realistic movement when interacting with an environment. Baraff (1989) presented analytical methods for rigid body dynamic simulation. However, there are so many unknown parameters that need to be defined for this computation, e.g. muscle force and external force, etc. In addition, due to the complex geometrical shape of a human character, it is also difficult to obtain accurate physical parameters, i.e.

mass distribution, to carry out the computation. The **Inverse Dynamics** method is a practical approach to obtain these forces (Issacs and Cohen 1987). According to the joints' trajectories of an articulated character, the inverse dynamic method is able to compute the desired forces that can be formed into a forward dynamics simulation procedure for computing the physical movement of a character.

From Equation (2.2), a set of differential equations is formed to describe the movement of a character. In order to obtain the positions of the character in the temporal domain, an iterative method is employed for the computation of this system in which the consequence of time t_i depends on the previous results of time t_{i-1} . Animators have very limited control during the computation. They are only available to set the time step Δt , the iteration step n and some initial parameters. Any small disturbance or unexpected change of dynamic properties in previous time t_{i-1} will affect the following calculation in current time t_i (Popović 1999). The consequences of dynamic computation may converge after an expected number of iterations, or they might end up diverging. In addition, only a certain number of forces within the dynamic system can be estimated, the created human character movement by forward dynamics simulation may look rather "robotic".

Due to the difficulty of obtaining external forces to fulfil forward dynamic simulation, researchers try to find alternative means for generating physically correct human character movement. Controller design had made remarkable progress for controlling the movement of legged robots in robotics research (Raibert 1986; Bruderlin and Calvert 1989; Raibert and Hodgins 1991; Honda 2003; Yin et al. 2007); it presents the ability of movement generation in the physical world by estimating the joint torques from the controller. With the utilisation of **Robotic Controller Theory**, a large number of human character animations have been created. Hodgins et

al. (1995) demonstrated various behaviours of an athlete by estimating joint torques as Proportional-Derivative (PD) servos to control the position of all joints according to Equation 2.3.

$$\tau = \kappa(\theta_d - \theta) + \kappa_v(\dot{\theta}_d - \dot{\theta}) \quad (2.3)$$

where τ is a joint torque that drives the movement of a joint, κ and κ_v are gain constants which are defined for the strength of the joint, the θ_d and θ are the desired and actual joint angles respectively, $\dot{\theta}_d$ and $\dot{\theta}$ are the desired and actual angular velocities of the joint respectively.

Hodgins and Pollard (1997) adapted the simulated behaviours for different size human characters after the controller parameters are adjusted by an optimisation search method. Subject to the selected dynamic model and criteria, feed-forward control is used to determine the unique trajectory which yields the best performance. According to the prescribed trajectories of manipulator joints, feedback controllers are used to compute the desired generalized force (Craig 1989; Laszlo et al. 1996). Faloutsos et al. (2001) also proposed a composable controller that is based on Support Vector Machine (SVM) learning theory. They synthesised a set of balanced movements for virtual human character animation. Oshita and Makinouchi (2001) presented a dynamic motion control technique that takes into account comfort and balance control to compute the joint's angular acceleration. Although the robotic controller approach provides an alternative technique for creating a physical plausible human character animation, and more controllers are achieved for character motion synthesis, it has not been widely used because the specific controller design for the actuator that is able to drive the movement of the joint. As opposed to using robotic controllers for generating human character animation, researchers also apply human behaviour or motion to drive a humanoid robot (Dasgupta and Nakamura 1999; Ude et al. 2000; Pollard et al. 2002). Based on a planar physical biped model, Sok et al. (2007) simulated certain walking behaviours by studying human motion from different motion sources.

Dynamic and robotic controller approaches enable animators to generate a series of motion sequences directly and guarantee the generated motion is physically plausible. Instead of generating movements, some researchers present a method based on **Filter Processing** to process and verify the existing motion sequences according to certain physical criteria. Tak et al. (2000) presented a filter that takes balance into account for correcting unbalanced motion to a balanced one by analysing and controlling zero moment point (ZMP). A similar idea presented by Yamane and Nakamura (2003) employed a similar PD control as Equation (2.3) in their online dynamic filter that converts reference motion to a physically feasible movement according to an interaction response with the external environment. Their method relies on efficient dynamic simulation of a rigid-body collision and contact model. Similar to the robotic controller, the filtering method has not been widely used because it only modifies a small set of motions rather than the whole motion sequence. Even though the filter approach presented animators with another possible means for generating physically accurate human character animation, inappropriate use of the filtering may result in the loss of certain valuable information relating to character movement.

Research in the field of **Biomechanics** focuses on the analysis of human body movement. They endeavour to identify the relationship between human movement, structure of skeleton, neuron system and muscle response. A few physics-based human models have been studied for simulating the human movement (Winter 1990; Isbweb 2003). The study of Biomechanics provides a wider knowledge for animators to generate physical available human movements in character animation. According to the established muscle models from biomechanics research, Komura et al. (2000) presented a method for creating and retargeting human body motions by employing a musculoskeletal human body model. Sun and Metaxas (2001) presented a gait generator that based on the analysis of human movement from sagittal elevation angles.

According to an animator's sketch draft design on possible movements of a hierarchical object, Witkin and Kass (1988) presented and proved that **Constrained Optimisation Method** is a powerful approach for generating new motions for character animation. The new motions are generated according to their objective function of minimum energy consumption subject to some constraints in the spatial-temporal domain (Equation 2.4).

$$\min_x f(x) \quad \text{s.t.} \quad c(x) = 0 \quad (2.4)$$

where $f(x)$ is an objective function, $c(x)$ are the constraints and x are the independent parameters.

Given a set of key poses in the temporal domain, the constrained optimisation method is able to create a whole sequence of motion that obeys physical laws. Based on a space-time constrained method, Cohen (1992) and Liu et al. (1994) had developed an interactive, goal-directed motion synthesis technique for animators. In order to avoid calculating the joint torques for physically accurate motion, Liu and Popović (2002) enforce patterns of linear and angular momentum of a character's body parts to preserve dynamic effects. Fang and Pollard (2003) presented an efficient algorithm where the first derivatives of physical constraints can be achieved in linear time. In addition, Fang and Pollard's optimisation method scales well for more complex articulated characters that have degrees of freedom in the range from 7 to 22.

The physically-based techniques described above can guarantee animators that the generated motion sequences are readily available for motion editing and motion synthesis. Except time-consuming for generating realistic movement by these physical based methods, for a multi degrees of freedom human character animation, the motion generated may not be the most convincing motions for realistic human character animation.

2.3 Motion Transition

Motion transition, also called motion blending, is a branch of motion editing and motion synthesis which creates intermediate motion from two motion sequences. Perlin (1995) demonstrated the motion blending technique in his real-time animation system and created the in-between transitions via interpolation (Equation 2.5).

$$\theta(t) = (1-w)\theta_i(t) + w\theta_e(t) \quad (2.5)$$

where $\theta(t)$ is the final blended motion curve, $\theta_i(t)$ and $\theta_e(t)$ are the starting and ending motion curves respectively, w is the normalised weight function.

Considering the smoothness of motion transition, linear interpolation is the quickest method for creating transitions from different motions (Abe et al. 2004; Wang and Bodenheimer 2004). Safonova and Hodgins (2005) have shown that the interpolation technique is a practical method for creating physically plausible motion by analysing the physical correctness of human motion.

Rose et al. (1998) implemented a multi-dimensional motion interpolation technique. They parameterized the motion through radial basis functions, and manually set the appropriate time for interpolation. To create a smooth transition from different motion resources, **Time warping**, a signal processing technique has been proved to be a powerful tool for mapping motion curves with different timing arrangement (Bruderlin and Williams 1995; Kovar 2004; Kovar and Gleicher 2003; Kovar and Gleicher 2004; Ménardais et al. 2004; Hsu et al. 2007). By applying the time warping method for motion transition, the generated motion guarantees not only the smoothness of trajectory but also the time accuracy from the aligned motion clips.

Despite the kinematics based method that has mentioned above, based on physics-based human character model, the **Simulation** method has been

studied for generating reactive motions for human character animation (Pollard and Behmaram-Mosavat 2000; Komura et al. 2004). Following their previous work, Zordon et al. (2005) manually selected a set of motions as a transition reference. Then a dynamic simulation model is applied to compute reactions from unanticipated impacts and returns an available motion sequence from selected motion data in the motion repository. Rose et al. (1996) presented a technique that generates seamless and physically feasible transitions between different motion segments by applying space-time constraints and inverse kinematics constraints.

In contrast to offline blending techniques for synthesizing motion from different motion sources, researchers were also interested in on-line motion blending for character animation (Oore et al. 2002; Park et al. 2002; Kwon and Shin 2005).

The interactive control of character movement and the creation of smooth transitions are vital for on-line performance. Hsu et al. (2005) presented a method for stylistic motion translation by analyzing differences between performances of the similar motion from input and output styles. McCann and Pollard (2007) stored a large amount short motion fragments in the repository for an on-line controller to choose. The choice of movement is based on current motion input and the previous fragment.

Recently another technology, known as **Motion Graph**, has been heavily involved in the computer games industry. Motion graphs are expected to construct long, smooth motions online by searching appropriate motion clips in a large data set for motion synthesis.

In early research, researchers manually constructed motion graphs and explicitly decided which motion clips were to be used for a transition (Perlin 1995; Rose et al. 1998). According to the developed informative heuristic motion graph, other automatic searching approaches have been

presented to identify and extract appropriate motion clips for motion transitions (Lee et al. 2002; Arikian and Forsyth 2002, 2003, 2005; Kovar et al. 2002; Safonova and Hodgins 2007).

Motion transition techniques provide a powerful tool to create continuous motions from existing different motion resources and extend the ability for controlling the animation sequences. Even though existing motion resources provide animators with various choices for motion blending, their ability are also limited for creating realistic and convincing motion sequences from existing motion databases or motion graphs. For example, to blend a running sequence from existing motion clips running at speed 2m/s and 7m/s. Traditional motion blending techniques will create a continuous transition according to available motion clips. The physical properties of blended motion will change in a very short period of time, i.e. velocity and acceleration. The character running will appear unrealistic. However, if an intermediate running motion sequence, i.e. running at 4.5m/s, could be generated to support the motion blending, the motion transition will change gradually which appears smooth and realistic not only in trajectory but also in physical properties. According to the scenario for motion blending, a statistical based method is applied to extract motion from existing motion database and an incomplete motion data reconstruction method is applied to generate transitional motion for motion blending according to the animators' direct in Chapter 5.

2.4 Motion Synthesis from Motion Capture

Motion synthesis is a very important procedure for character animation. Although the technology has been mentioned earlier can generate a certain range of motion, it may not satisfy the animator's demand for creating realistic movement. In this section, the MoCap technique is reviewed for creating high-fidelity, realistic human character animation.

2.4.1 Human-like skeleton mapping from motion capture data

In order to use motion capture data efficiently and to manipulate a hierarchical skeleton for further motion editing, it is important and necessary to obtain joint rotation angles from the recorded motion capture data.

Based on different MoCap systems, several researchers have presented their solutions for this specific problem. Bodenheimer et al. (1997) described a procedure for transforming magnetic motion capture data into an articulated human skeleton movement which requires manual measurements of body segments. O'Brien et al. (2000) also estimated the skeleton from a magnetic motion capture system where each marker has both position and orientation information. By comparison, the data captured from an optical motion capture system only records position trajectory for each marker. The optical motion data are non-rotational, three-dimensional Cartesian positions. To construct an articulated skeleton from a camera-based multi-marker optical motion capture system, Silaghi et al. (1998) presented a method in which the joint position is inferred, based upon adjacent joint link information. Kirk and his colleagues (2005) grouped the markers in different clusters according to assignment of markers to body segments, and then they used a fitting method to estimate the joint position.

Since the captured marker's trajectory describes the movement of the physical body parts of a real performer, Zordan and Van Der Horst (2003) assumed that all the markers attached on an actor are driven by force separately. In order to test their assumption, they mapped the marker data into a fixed human character which includes a physically-based forward dynamic model. Their method for optical marker mapping is relatively time-consuming. The joint angle information of a fixed limb-length skeleton is obtained from the simulation of the equilibrium state.

The optical motion data mapping approaches described above mainly infer the joint position based on multi markers information attached at the adjacent body segments. Even though these methods can generate a reasonable skeleton directly, the methods of Silaghi et al. (1998) suffer from the errors that mapping data into a pre-defined skeleton to achieve joint rotation. The methods of Kirk et al. (2005) suffer from a simple prototype of a human skeleton which lost important end-effectors' information for reconstructing a hierarchical skeleton. According to the knowledge of human anatomy, in this presented method (Chapter 3), the skeleton prototype can be defined automatically and a hierarchical skeleton is constructed by using less marker information. In addition, this automatic method is easy to be extended for the construction of quadruped animal according to the structural similarity, and the per-frame based joint estimation is able to be achieved in real time.

2.4.2 Motion capture based motion synthesis

By using optical or magnetic sensors located on an actor, motion capture became a popular technique for producing highly realistic motions. However, due to the spatial limitation of the MoCap studio, and the fact that there is no motion control by motion capture itself, further research has been undertaken for motion capture based motion data synthesis.

Bruderlin (Bruderlin and Williams 1995) and Witkin (Witkin and Popović 1995) modified and interpolated motion data directly for effective re-use. Unuma et al. (1995) presented the Fourier series expansion method for interpolating motions from different motion clips. Lee and Shin (1999) employed a multi-level B-spline technique for motion editing. By using optimization approach over an entire motion sequence, Gleicher (1998) edited and adapted original motion data to a new character. To edit original captured motion and adapt the motion to different environment, researches also focus on the motion path editing and planning (Van De Panne 1997; Gleicher 2001; Choi et al. 2003; Yamane et al. 2004). These editing

methods described above are based on kinematic properties of motion, which is simple to compute and easy to control.

Considering the physically plausible motion for character animation, Popović and Witkin (1999) mapped the existing motion capture data onto a simplified model that contains the minimum number of degrees of freedom, and then using a space-time optimisation method, a wide range of realistic motion have been created according to the input motion data. Zordon and Hodgins (2002) also presented a method for synthesizing human-like character motion by modifying motion capture data with dynamic simulation.

It is known that human behaviours are unique and distinguishable, the studies of human behaviours has a long history in the area of computer graphics (Philips and Badler 1994; Lau and Kuffner 2005; Safonova 2006; Sok et al. 2007). In a typical MoCap session, there are a large number of similar motion sequences that have been captured for animation productions. In order to re-use MoCap data efficiently, in recent years, many researchers focus on abstracting human behaviours from motion capture data. Prior to motion synthesis from individual motion behaviours, segmentation of motion data is the key for the consequences of final motion transition (Jenkins and Mataric 2002; Barbič et al. 2004; Hsu et al. 2004a; Lee and Lee 2004; Mukai and Kuriyama 2005; Müller et al. 2005).

Currently, statistical based methods for analysing different motion patterns within a motion database and building similar behaviours according to the abstracted motion pattern is more promising. The Hidden Markov Model (HMM) can be trained to learn to synthesize motion data by interpolation or extrapolation from a motion database (Brand and Hertzmann 2000; Ren et al. 2005). Alexa and Müller (2000) proposed an approach to represent an animation sequence based upon principal component analysis on three-dimensional geometry. Troje (2002) provided a framework for recognizing

and classifying human biological motion using principal component analysis. Lim and Thalmann (2002) described a method that constructs parametric motion and skeleton data by interpolating a set of labelled motion capture data in low dimensional space. Glardon et al. (2004) generate a generic real-time walking engine for human character animation. Urtasun et al. (2004) also presented the power of a statistical method. They extrapolated motions in real time animation at different speeds for the same person walking and running. After extracting behaviour patterns in low-dimensional space from large sets of motion capture database, Safonova et al. (2004) synthesise a continuous motion sequences by employing physical constraints in their optimisation methods. According to the pre-recorded human database, Chai and Hodgins (2005) reconstruct a full-body human motion by applying low-dimensional control signals.

By applying Principal Component Analysis (Chapter 4), a similar motion is able to be synthesized but for a different size character. Such an analysis enables animators to construct a motion pattern in low dimensional space from existing motion database. The method for generating intermediate motion is able to be applied for incomplete motion data reconstruction in the following chapter. In this study, similar data acquisition approach has been used for obtaining a wide range of motion database like other researchers; the weakness of using a treadmill for motion data collection is also discovered.

2.5 Summary

Most of the techniques described above are offline process for generating and synthesizing realistic movements for three-dimensional human character animation except for the motion graph technique. Each of them has its own strengths and weaknesses. The utilisation of MoCap data overcomes the complexity of the articulated human skeleton and achieves realistic and usable motion data. The application of statistical analysis of

human movement will enforce and extend the ability of animators to re-use MoCap data efficiently and persuade animators to create convincing motion heuristically for similar behaviour movements. The physically-based constrained optimisation methods will guarantee that the synthesized motion is physically accurate. Thus exploring advantages from each of the technologies for synthesising realistic and physically accurate human character animation is extremely important for the computer graphic industry.

CHAPTER 3

ESTIMATION OF SKELETAL

MOVEMENT

Increasingly, motion capture techniques are becoming more popular in the character animation pipeline. However, deriving accurate joint positions and orientations from captured data is still a potential concern for motion capture studios.

3.1 Introduction

Motion capture is a technique and a process that digitally records the movements of a live “performer” such as a human or an animal in a specific 3D environment (Menache 1999). Once recorded, the motion data are used to animate a computer-generated character. Since motion capture techniques are, currently, the most effective means of achieving realistic and convincing motion from a “performer”, they are widely used in computer games, television and for film special effects where human-like character animation is heavily involved and other areas where motion analysis is performed.

In computer animation the motion of an articulated skeleton character is defined by a world coordinate transformation at the root joint. Rotational

angles are then associated with other joints to establish an articulated hierarchy. However, optical MoCap systems record only a set of three-dimensional Cartesian points that describe the trajectories of markers attached to the live performer. The recorded three-dimension motion data cannot themselves be used for manipulating a character directly because the output data does not explicitly involve a hierarchical skeleton. It is difficult to construct a proper skeleton based upon the captured optical motion data, as the transfer of the three-dimensional position data to a skeleton joint rotational data is non-trivial. The conversion of the Cartesian positional data into a hierarchical skeleton movement in joint space is an important procedure associated with motion capture. The mapping of marker movements to an articulated skeleton's joint rotation form, create a format that permits it to be further edited and eventually used for animating a character. Such a procedure makes further motion data editing more convenient and shortens the production pipeline of computer animation.

Currently, passive and active optical motion capture systems are the two mainstream systems for human-like characters. Both systems use markers attached to the performer: in a passive system, each marker has to be identified continuously during a capture session in order to obtain the location of each marker, whereas in an active system the markers are individually identified using pulsed light-emitting diodes (LEDs).

Compared to the tracking techniques used in a camera-based motion capture system, the Ascension ReActor2 (Ascension 2003; AccessMocap 2005) is a signal-processing-based, active, optical motion capture system. The current capture space is restricted to a $3 \times 3 \times 3$ cubic meter volume. This system has fewer occlusion problems than the passive systems as the markers are instantly recognized after an occlusion. This system uses only 28 infrared markers for tracking a live performance, each of which is attached at the related joint position based on the skeleton hierarchy of a human character. During calibration the detectors are placed at specified

positions in 3D space and receive flashing signals from each marker to determine the location of the markers. The recorded marker positions are discrete three-dimensional position points in the time domain.

Autodesk's Motionbuilder (Autodesk 2007), is a popular commercial software package that uses a pre-defined skeleton for mapping the optical motion capture data to a hierarchical character. During the calibration procedure, the motion capture producer manually adjusts the prototype of the pre-defined character and groups the locations of markers for data mapping. This is a tedious, time-consuming procedure and the quality of output motion data is dependent upon the skill and experience of the operator.

In this chapter, a method that will automatically group marker sets to find accurate joints location in the hierarchy is presented. The motion capture is conducted by ReActor2, a signal-based, active motion capture system from Ascension Technology Co. USA, and there are less markers attached to the performer than used with a passive optical system. In order to compute accurate joint positions and obtain precise joint orientations, the presented method relies heavily on biomechanical information associated with human motion (LeVeau 1992; Winter 1990). Although inverse kinematic techniques (Zhao and Badler 1994; Bodenheimer et al. 1997) are used for configuring intermediate joints relative to end joint positions, they suffer from singularity and initial starting point problems. Similar to the global technique presented in (Silaghi et al. 1998), a recursive method with joint angle minimization is applied to calculate the joint rotational angles in the hierarchy. Comparing to manually marker mapping, the presented automatic skeleton estimation method aims to satisfy the following criteria:

1. Efficiency: Automatic marker mapping technique enables a MoCap producer to achieve skeletal movements quickly.

2. **Simplicity:** Automatic marker mapping technique simplifies the procedure of a MoCap pipeline and enables a MoCap producer to achieve high-quality motion.

3.2 Motion capture and skeleton fitting

In an ideal MoCap session it is assumed that the markers attached on the body stay relatively static. The marker locations represent the best estimation of joint position according to the anatomical topology of a human performer. From the marker locations, an approximated human skeleton is constructed according to the knowledge of human anatomy (Xiao et al. 2009b). Figure 3.1 depicts the procedure for constructing such a skeleton from the motion capture data.

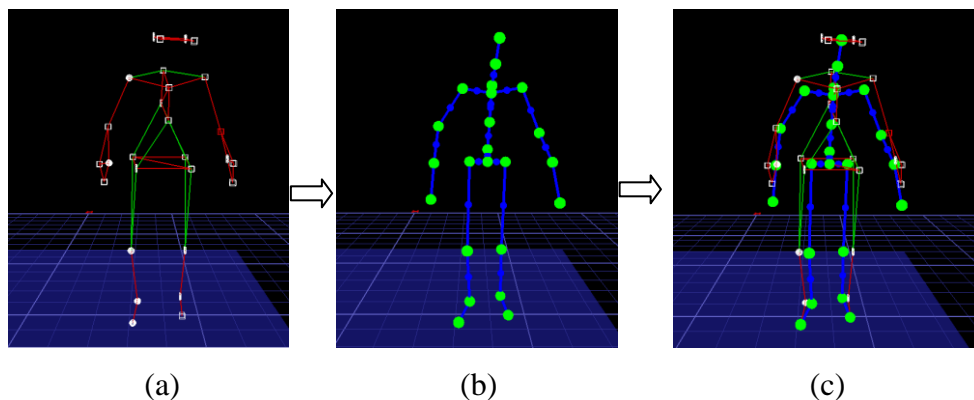


Figure 3.1 The procedure for creating a human skeleton from motion capture data. (a) Stick figure constructed from marker positions; (b) The prototype of a human skeleton; (c) Data mapping from marker positions to a human skeleton in joint space.

To avoid the manual adjustment of a pre-defined rigid body skeleton for optical marker mapping, the presented method automatically estimates joint positions from biomechanical information (LeVeau 1992). Figure 3.2 shows the front view of a human character model with a skeleton beneath the skin and some marker locations for motion capture arrangement. The procedure for skeleton identification includes two stages: a) Identify each

marker and group of markers for individual rigid-body parts, then calculate the actual joint rotation centre based upon the anatomical knowledge of the human body. b) Calculate the joint rotation angles according to the fitted hierarchical skeleton.

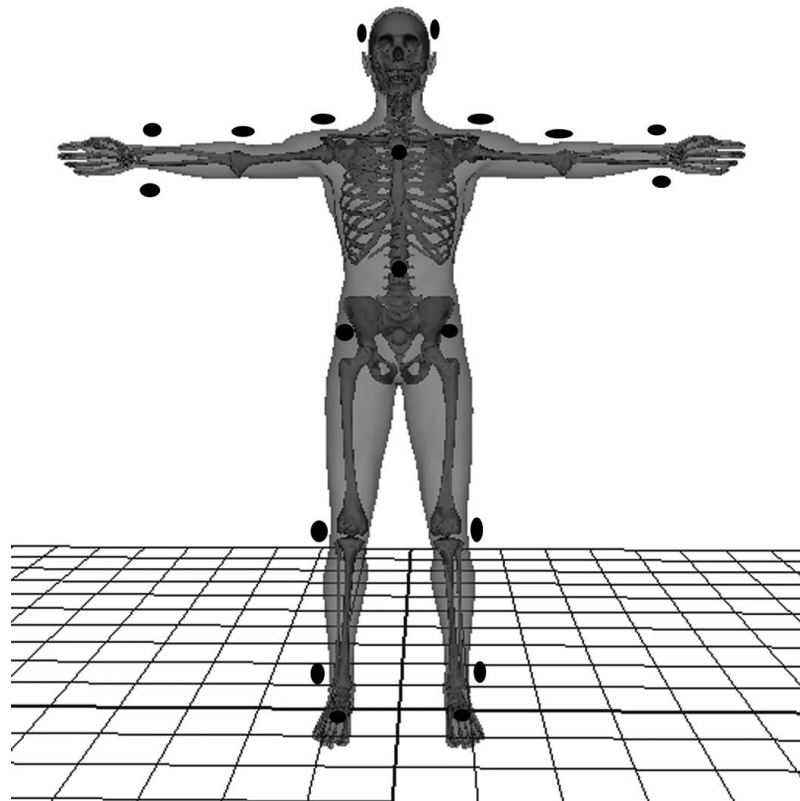


Figure 3.2 Topology of a human character and some marker positions shown with a black dot.

3.2.1 Marker identification and joint determination

Unlike a passive system, the markers in an active system are able to “communicate” directly with the tracking system, e.g. signal detectors. The communication is achieved by means of LED pulses. The LED marker signals can be affected more or less by local environment lighting levels. As this illumination noise can affect the quality of the motion capture data, it is important that this is kept to a minimum for obtaining precise marker position. Infrared radiation has a spectrum range between 10^{11} to 10^{14} Hz.

Since the frequency spectrum of infrared is well defined, the removal of any noise can be solved by means of band-pass digital filtering.

To construct a hierarchical human skeleton, it is essential to find the proper joint positions that connect adjacent body segments. Prior to a motion capture session, the MoCap producer usually attaches each marker on the performer in a position that is regarded as a reference of the performer's joint. Once the detector has received the LED marker pulses, the precise marker position that describes the performer's movement in three-dimensional space is recorded. In an ideal motion capture session, the distance between every two markers that are attached on the same rigid-body part should remain constant, which means that the two adjacent markers would define one rigid-body link in the hierarchical skeleton. From the biomechanical information of human motion (LeVeau 1992; Winter 1990), the distance between the actual human joint location and the markers in the referential marker group should be unchanged during the movement. This is shown in Figure 3.3.

For a hierarchical skeleton, each joint has a local coordinate frame located at the centre of the joint with the \hat{y} axis lying along the corresponding rigid body part pointing toward the child joint. For example, the knee joint shown in Figure 3.3, the \hat{x} axis is the main rotation axis. The \hat{z} axis is obtained with the cross product $\hat{z} = \hat{x} \times \hat{y}$. Therefore, the joint position along $-\hat{x}$ axis is able to be defined according to the marker position. To guarantee that the joint location is constant for motion data mapping, the joint location is formulated as follows

$$P_i^k = P_r^k + d \cdot \hat{x} \quad (3.1)$$

where P_i^k is the k^{th} joint position in i^{th} frame, P_r^k is the referential marker group position relative to the k^{th} joint in the i^{th} frame, d is the constant distance between marker group and actual joint, \hat{x} is the unit vector along the X axis as illustrated in Figure 3.3.

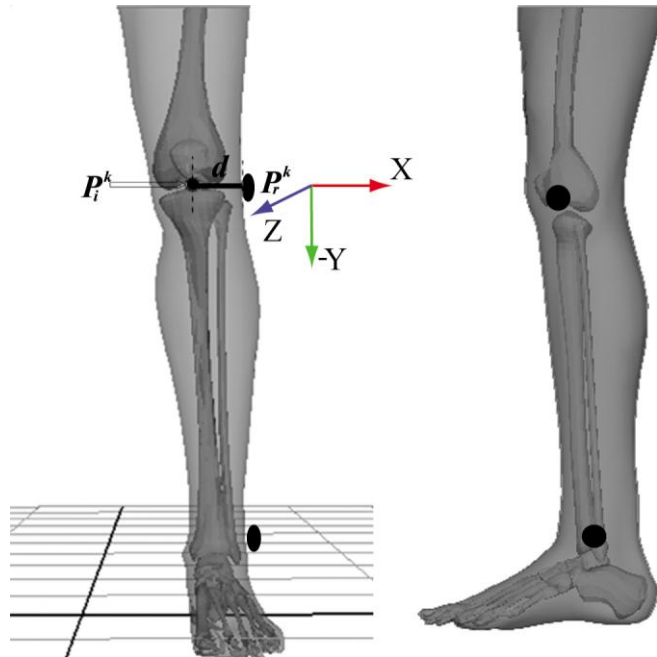


Figure 3.3 Human character joint position calculation at knee joint.

The following code is the algorithm to compute the position of a point that corresponds to a surface in space:

```
inline Vector SpacePoint(int mode, float len, Vector &a, Vector &b, Vector &c)
{
    Vector virPos, virPos1, virPos2, direct;
    float l1, l2, l3, l4, l5, l6, A, B, C, M, N, H, A1, B1, C1, D1, test;

    l1 = b.x - a.x; l2 = b.y - a.y; l3 = b.z - a.z;
    l4 = c.x - a.x; l5 = c.y - a.y; l6 = c.z - a.z;
    M = DotProd( (b-a), a);
    N = DotProd( (c-a), a);
    H = len*len - Length(a);

    A1 = (l4*M - l1*N)/(l2*l4 - l1*l5);
    B1 = (l1*l6 - l3*l4)/(l2*l4 - l1*l5);
    C1 = (l5*M - l2*N)/(l1*l5 - l2*l4);
    D1 = (l2*l6 - l3*l5)/(l1*l5 - l2*l4);

    A = D1*D1 + B1*B1 + 1;
    B = C1*D1 - a.x * D1 + A1*B1 - a.y * B1 - a.z;
    C = C1*C1 - 2 * a.x * C1 + A1*A1 - 2 * a.y * A1 - H;

    direct = CrossProd((c-a), (b-a));

    test = B*B - A*C;
    if( test < 0 ) { virPos = a; }
    else
    {
        virPos1.z = (-B + sqrt(B*B - A*C))/A;
    }
}
```

```

virPos1.y = A1 + B1*virPos1.z;
virPos1.x = C1 + D1*virPos1.z;

virPos2.z = (-B - sqrt(B*B - A*C))/A;
virPos2.y = A1 + B1*virPos2.z;
virPos2.x = C1 + D1*virPos2.z;

switch(mode)
{
case 0:
    if(sign > 0 )           * modified from original code
    { virPos = virPos1; }
    else
    { virPos = virPos2; }
    break;
case 1:
    if( sign > 0 )         * modified from original code
    { virPos = virPos2; }
    else
    { virPos = virPos1; }
    break;
default :
    break;
}
}

return virPos;
}

```

Having calculated the joint positions, the next step is to find the rigid-body segments. Theoretically, the body segment between two adjacent joints should have the same length within all frames. Constant skeleton length for each rigid-body segment can be computed over time. Ideally, the standard

deviation $\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (l_i - \bar{l})^2}$ for each body segment is zero. Where

l_i and \bar{l} are length at i^{th} frame and mean of length over time respectively for each body segment.

3.2.2 Joint angle calculation

Once a hierarchical rigid-body skeleton has been fitted according to the joint position and rigid-body segments, the joint's information in local and global coordinates is obtained based upon the recursive transformation

matrix $X_w = R(q)X_l$. Where X_l, X_w are the joint positions in local and global coordinates system respectively, and $R(q)$ is the recursive transformation matrix calculated from the global coordinates to current joint coordinates.

To better understand the transformations within a hierarchical chain, the computation of the world coordinate position P of a foot point P' within the skeleton is demonstrated. (See Figure 3.4)

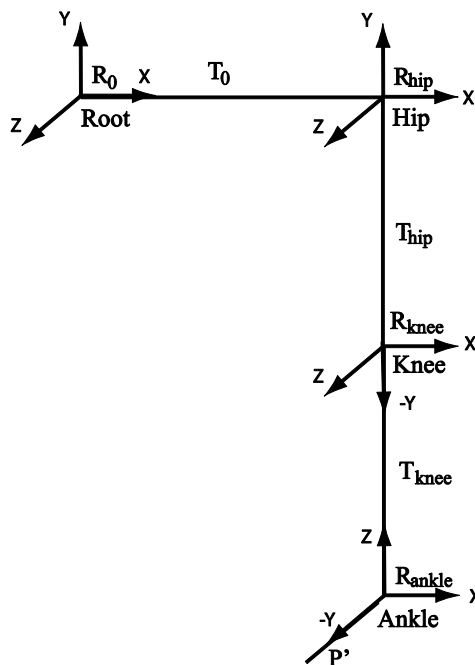


Figure 3.4 Hierarchical chain

The Euler angle notion is used for the implementation in this research. For example, here is a rotation sequence associated with a turn by rotating Z, X, Y axis $R_z(\alpha), R_x(\beta), R_y(\gamma)$. The final rotation $R_{zxy}(\alpha, \beta, \gamma)$:

$$R_{zxy}(\alpha, \beta, \gamma) = R_z(\alpha)R_x(\beta)R_y(\gamma) \quad (3.2)$$

$$\begin{aligned}
R_{xy}(\alpha, \beta, \gamma) &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\beta & -s\beta \\ 0 & s\beta & c\beta \end{bmatrix} \begin{bmatrix} c\gamma & 0 & s\gamma \\ 0 & 1 & 0 \\ -s\gamma & 0 & c\gamma \end{bmatrix} \\
&= \begin{bmatrix} c\alpha c\gamma - s\alpha s\beta s\gamma & -s\alpha c\beta & s\alpha s\beta c\gamma + c\alpha s\gamma \\ s\alpha c\gamma + c\alpha s\beta s\gamma & c\alpha c\beta & s\alpha s\gamma - c\alpha s\beta c\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix} \quad (3.3)
\end{aligned}$$

where $c\alpha = \cos(\alpha)$ and $s\alpha = \sin(\alpha)$. Therefore, the final translation will be:

$$P = R_0 T_0 R_{hip} T_{hip} \cdots R_i \cdots R_{ankle} P' \quad (3.4)$$

where R and T denote rotation matrix and transformation matrix respectively.

The same recursive transformation matrix had been discussed in (Silaghi et al. 1998). In the recursive method, the current i^{th} joint information within the fitted hierarchical skeleton is represented by its parent $(i-1)^{th}$ joint orientation matrices. The i^{th} joint rotation angles are calculated by minimizing the orientation error (Equation 3.5).

$$\Delta O(q) = \sum_{i=1}^3 (P_i^* \cdot P_i(q) - 1)^2 \quad (3.5)$$

where P_i^* ($i = 1$ to 3) are unit vectors along the desired joint orientation.

$P_i(q)$ are the unit vectors along the current joint orientation. The representation of global and local coordinates of a linked system is shown in Figure 3.5.

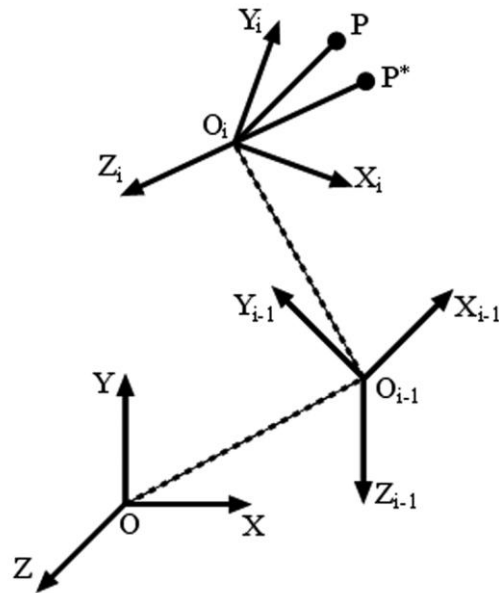


Figure 3.5 Coordinates transformation of linked system. P is the current joint end position in i^{th} joint coordinates, P^* is the desired end joint position in i^{th} joint coordinates.

3.3 Implementation and results

During the implementation, at the marker grouping stage, marker groups are set automatically according to the prototype of a human skeleton. For example, the four markers around the performer's waist are used to define the root joint of skeleton and the hip joints in the hierarchy. The constructed hierarchical skeleton includes 20 revolution joints, each, of which, has three degrees of freedom, and the root joint has six degrees of freedom which are three rotations and three translations. In addition, the output of motion is the standard Biovision hierarchical data format in which the rotation sequence of each joint is ZXY .

To compute the joint position, the method relies on biomechanical information of human motion. The standard deviation for each body length over time is less than 2%. However, for a standard human skeleton, an

empirical formula is able to be found for the calculation of the body segment length according to human heights as shown in Figure 3.6 (Winter 1990).

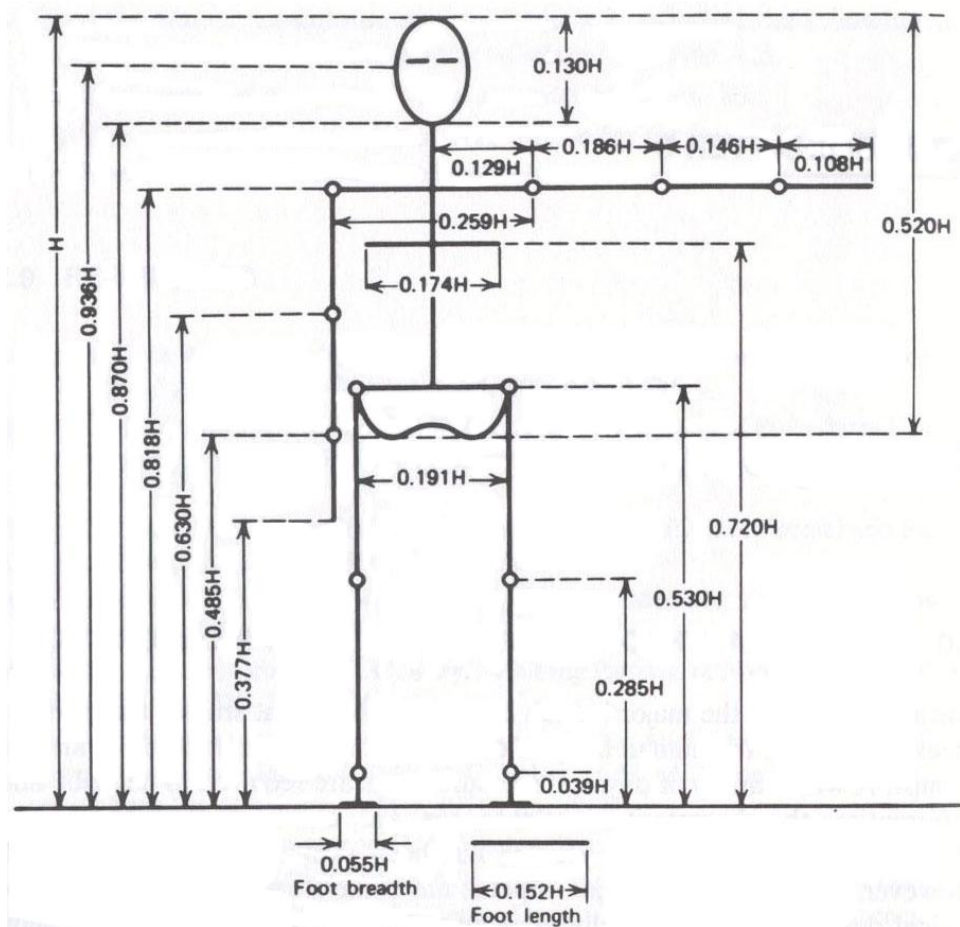


Figure 3.6 Anthropometric measurements from Winter (1990).

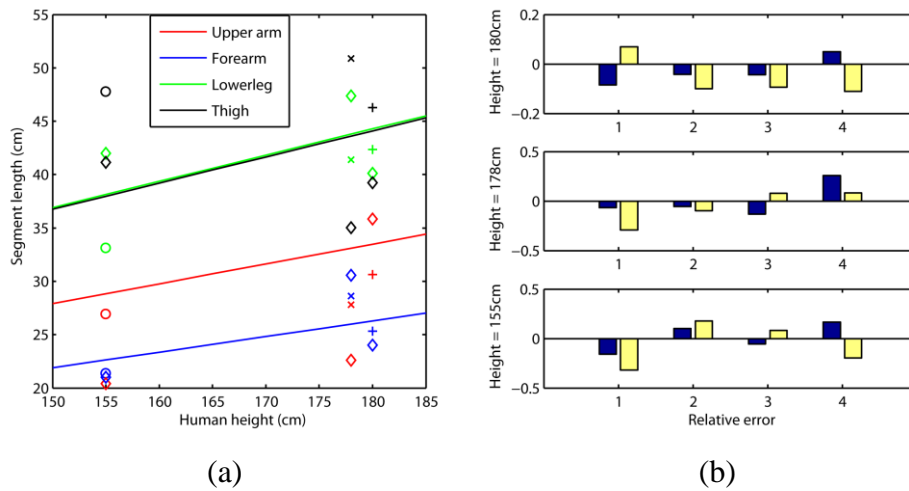


Figure 3.7 (a) Human body segment as fraction of human height, + and x from two male performers, O from a female. ◇ are obtained after the mapping procedure by the MotionBuilder software. (b) Based on anthropometric measurements, relative errors (in percentage) of each body segment from three sets of data obtained from the presented method (blue) and MotionBuilder output (yellow), from left to right, the columns correspond to the upper arm, forearm, lower leg and thigh respectively.

In Figure 3.7, the calculated body segment length from two males and one female vs. standard anthropometric measurements are shown. Moreover, one male dataset (marked as x) is obtained according to the marker set of a Vicon system. The data set obtained by this method are also used for comparison with parameters obtained after the mapping procedure by an experienced motion capture producer using MotionBuilder software (marked as ◇). From the illustrated figure, there are obvious discrepancies between each data set. The biggest is the length of a female's upper leg. In fact, the female does have a longer thigh in proportion to her height. Moreover, there is a 1 - 4 cm difference in other data sets compared with the anthropometric data. These errors are mainly due to the fact that the marker shape in a Reactor2 system is a 3 cm diameter disk. In order to prevent a marker moving from the attached point, the wrist crossbar and the ankle markers must be placed away from actual joint positions by about 1

to 2 cm to guarantee a stable marker position. As long as the movement of the hierarchical skeleton represents the movement of the optical motion data, the mapping procedure is successful. For the final character animation, the skeleton motion can be edited for a specific scene, and the further motion retargeting is able to compensate for any calculation errors.

After the calculation and determination of the joint positions and the length of each body part, the presented method is able to estimate joint orientations over time using the recursive method. In Figure 3.8 and Figure 3.9, the original raw motion data and the constructed human skeleton are shown. It is observed that the calculated position of the centre of mass of each individual rigid body segment is very much related to the extra markers used in a Vicon system. Similar to the techniques presented in (Silaghi et al. 1998), at least three markers on one rigid-body part are required to estimate the joint position. The presented method for mapping optical motion data is not specific to a Reactor2 system, it can fit other optical motion capture data as well. To test the method, various optical motion data is chosen ranging from subtle to more dynamic movements, e.g. dancing and martial arts. In Figure 3.10 a human skeleton with joint orientations is displayed, and it also includes the comparison of the skeleton generated by the presented method with the skeleton produced using MotionBuilder.

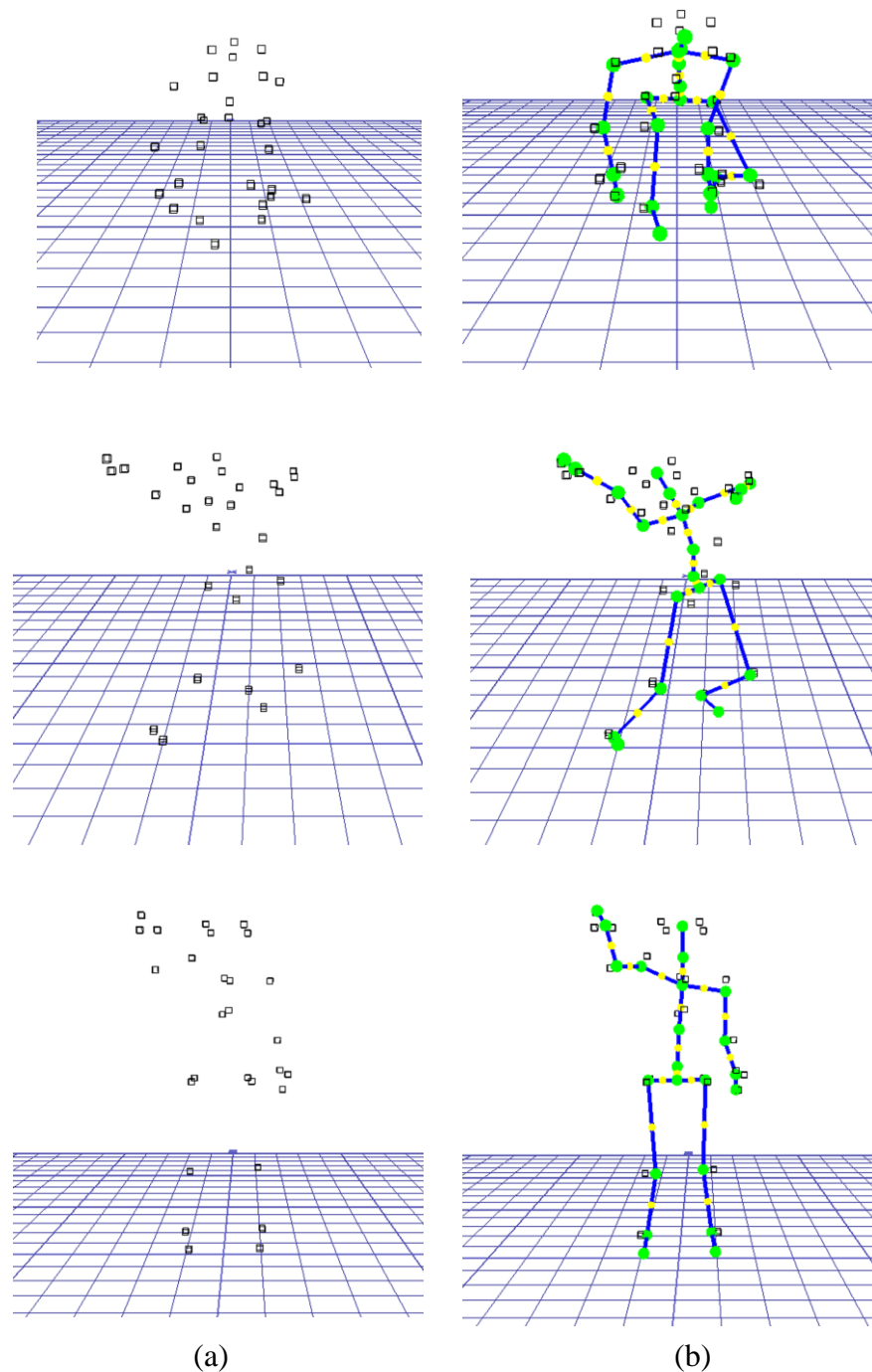


Figure 3.8 Column (a) are the raw optical motion data based on each individual marker set captured from Reactor2. Column (b) are the constructed skeleton (green sphere displays joint, blue bar is the rigid body segment, and yellow dot shows the centre of mass of each rigid body).

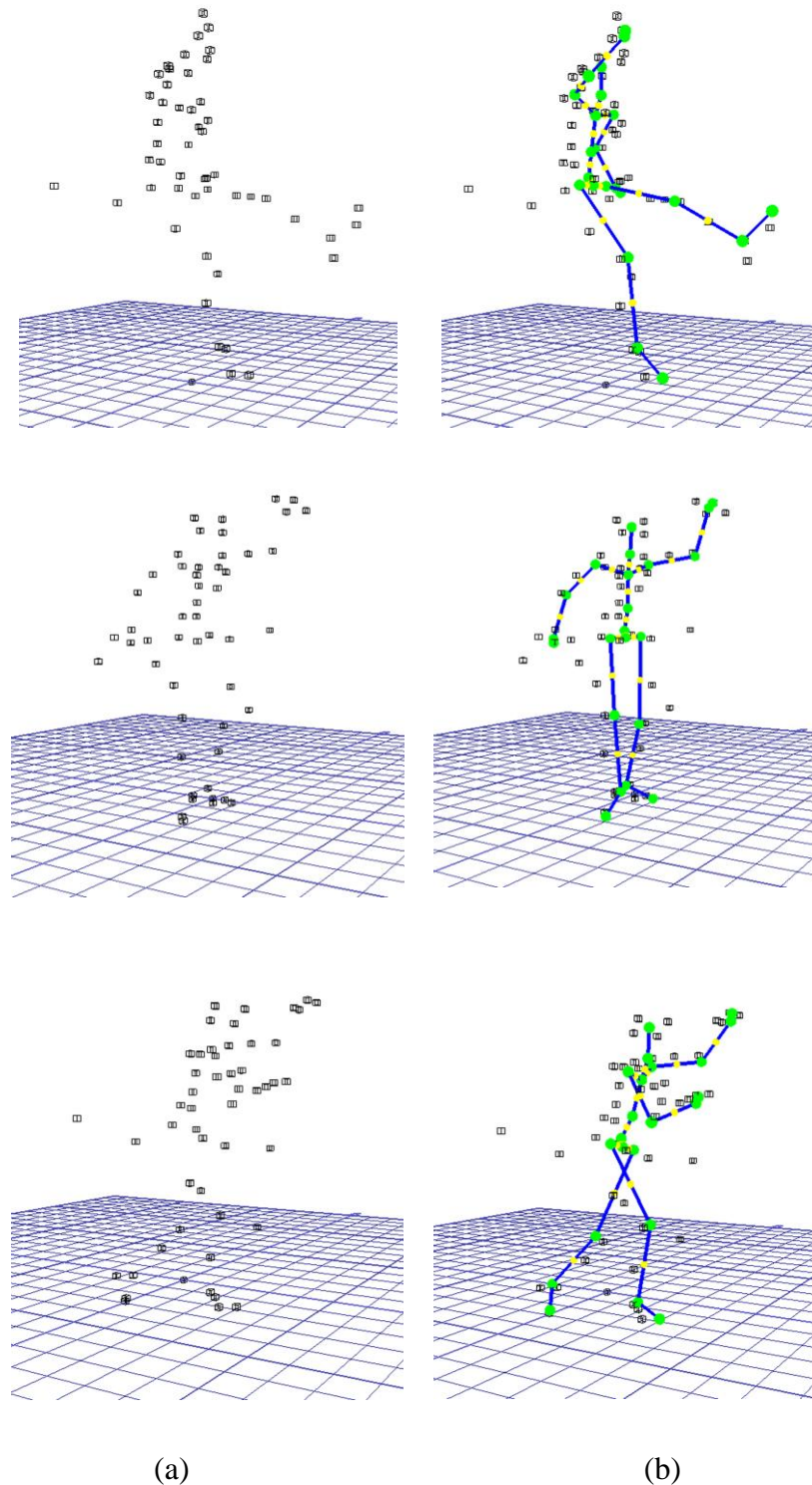


Figure 3.9 Column (a) are the raw optical motion data based on each individual marker set captured from a Vicon system (black square shows the marker position). Column (b) are the constructed skeleton (green sphere displays joint, blue bar is the rigid body segment, and yellow dot shows the centre of mass of each rigid body).

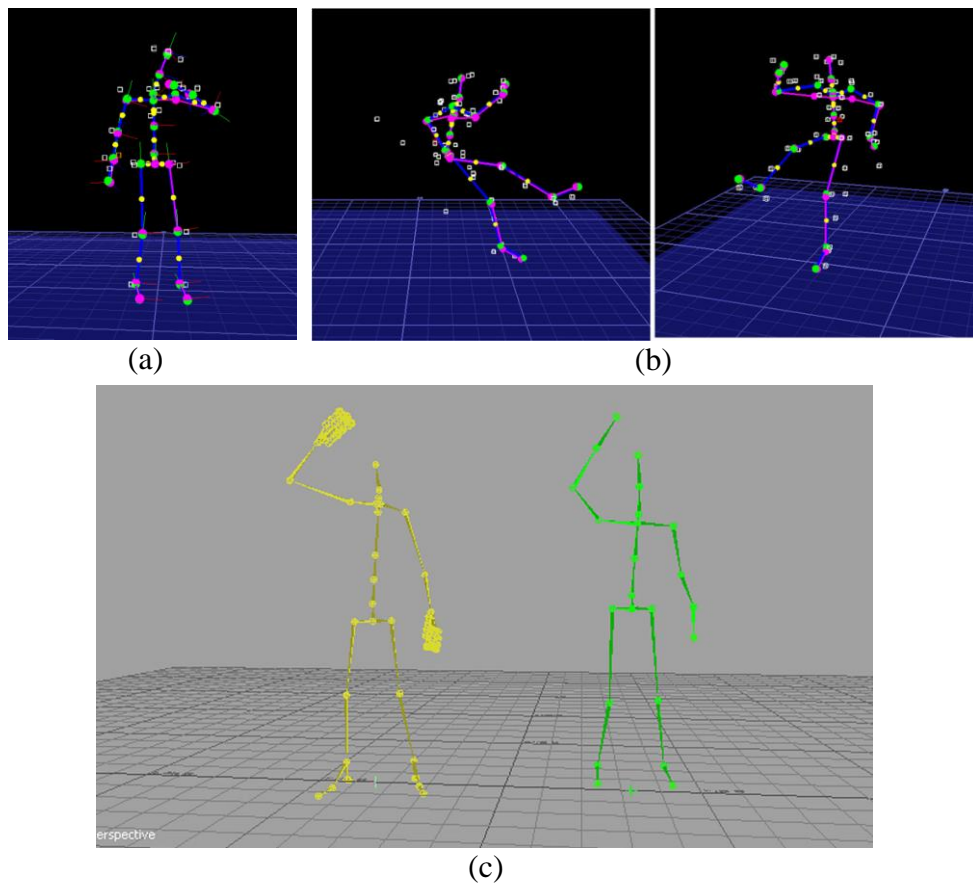


Figure 3.10 (a) and (b) show a human skeleton displayed with joint Axes (pink), which (a) is the skeleton constructed from a Reactor2 system and (b) is from a Vicon system; (c) is the skeleton output from the presented method (green) compares with the skeleton achieved from MotionBuilder (yellow)

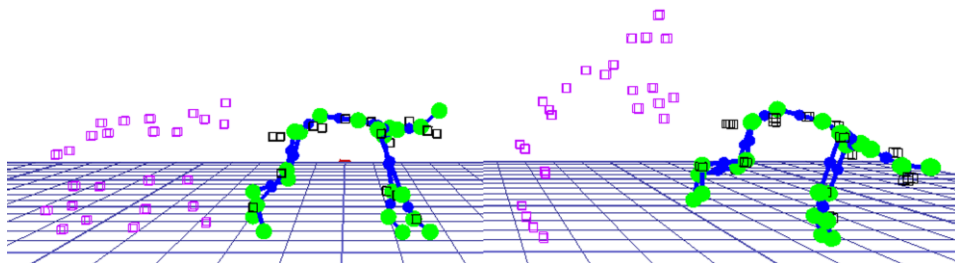


Figure 3.11 Image sequences of a dog running and jumping with an estimated skeleton.

Even though the presented method utilizes human biomechanics and anthropometric knowledge to estimate human skeletal motion, it can be easily extended to quadruped creatures. According to the layout of optical markers, the skeleton joints of a quadruped have been similarly identified. The results of skeleton estimation for a sequence of a running and jumping dog are shown in Figure 3.11. Here, too, the presented method yields satisfactory results.

All the tests were run on a 3.4Gz Pentium IV with 2GB of main memory. Since the estimation of joint orientation of a generated skeleton is per-frame based technology, the computation time for marker mapping and the skeleton joint calculation is about 2ms, and for a simple stick character, the graphic drawing takes around 2ms. The joints estimation method for a human-like skeleton has also been tested online through a network. The display of a simple stick character takes around 5ms. Although there is a computation cost for background rendering and time delay during the online data transferring, the display rate for this online test is achieved by 0.04s, which means it can play at around 23 frames per second. Given the differences of display for a simple stick character and a complex-shape character, the time for graphics displaying may increase. However, it is still fast enough for real-time character animation.

3.4 Discussion

Through the use of a novel per-frame based recursive method with joint angle minimization, human skeleton mapping from optical marker to joint angle rotations are achieved in real time. The output of motion data from a hierarchical skeleton can be applied for further character motion editing and retargeting. However, the accuracy of skeleton joint position and orientation relies heavily on recorded marker positions. There is a limitation for a motion capture system using a small number of markers. If one of the markers is lost or blocked, the signal detector would not record the marker

position. Consequently, the output of the skeleton motion could be compromised. In order to achieve acceptable motion data, the MoCap producer needs to either recapture the motion data or clean up the data after a capture session. The more information derived from optical markers, the more accurate the skeleton mapping can be achieved. After testing the presented method on other optical motion capture data, it is observed that the different naming conventions used by other optical systems would limit the automatic method. To get around this issue, the marker name can be mapped manually, or to define a comprehensive set of marker name for automatically mapping.

The presented approach is useful for mapping optical motion data to the same skeleton character. As demonstrated from test results, the process is able to play in real time. By simply scaling the skeleton and mapping joint orientation to a new character, it is not a precise way to create an online performance. Many more artefacts would appear after the motion data mapping, e.g. penetration. For this special online issue, the mapping approach would consider joint limitation and other physics-based approach for skeleton retargeting (Choi and Ko 1999; Gleicher 1998).

The developed algorithm first captures the raw motion data. It is the key procedure for dealing with optical motion data for character animation. Comparing with MotionBuilder that has a number of extra functions, such as that of motion retargeting, other post-processing techniques can be similarly incorporated should it be developed into a fully fledged package.

3.5 Summary

In this chapter, a method has been presented for automatically estimating and determining the topology of a hierarchical human skeleton from optical MoCap data based upon human biomechanical information. A per-frame based method provides the MoCap producer with an alternative choice for

processing raw motion capture data in real time. Compared to the off-line post-processing procedure for mapping the MoCap data into a hierarchical skeleton by MotionBuilder software, the presented method enables MoCap producer to achieve skeletal movement whilst motion capturing. The automatic estimation of skeletal movement method efficiently solves the skeleton mapping problem in less marker information. And it simplifies the processing procedure and shortens the animation production pipeline for using MoCap data.

CHAPTER 4

MOTION DATA ANALYSIS

Most MoCap data are related to the motion associated with person's everyday movements, and if it were possible to re-use such data, it would greatly improve the effectiveness of an animator. Furthermore, if it were possible to abstract each individual movement as a unique motion pattern from an existing MoCap database, it would be easier to generate a similar motion for a different size character.

4.1 Introduction

The purpose of a MoCap system is to capture as much movement data/information as possible from the performance of an actor within a defined working space. Since everyone has a very unique style in performance, the behaviours from one another can readily be distinguished. Apart from specific movements such as those found in martial arts, most motion data are associated with walking, running and jumping etc. Despite the profile of differences from different performers, i.e. gender, height or weight and so on, the captured data can be interpreted as one of many unique human behaviours. Although there is a mixture of behaviours in a single motion data clip, these behaviours can still be prioritized, for instance, throwing an object whilst walking. Here, there are two behaviours involved in this motion data: one is throwing and another is walking.

Before throwing takes place, walking is the underlying behaviour in this motion data.

During an every-day motion capture session, a large amount of data is captured according to the director's request. Unfortunately, according to the expectation of an animator, only few motion sequences has been chosen and edited for the final character animation production. In fact, as most existing motion data sets describe everyday activities, it would reduce the duplication of a motion capture session if such motion data could be re-used.

As a hierarchical character skeleton often has 50 to 60 degrees of freedom in joint space, the motion data storage is very large and proportional to the capture time. In order to describe the behaviours individually and represent them mathematically, a data set of the motions with the same behaviour need to be abstracted and constructed for further analysis.

One statistically-based method, Principal Component Analysis (PCA) (Jolliffe 2002), is able to construct a low dimensional space while retaining as much as possible of the variations present in the data set which describe the same behaviour. Recently, the PCA technique has been investigated for synthesising motion and extrapolating stylistic motion from a large amount of motion database in a low dimensional space (Glaridon et al. 2004; Safonova et al. 2004; Urtasun et al. 2004; Xiao et al. 2009a). By applying PCA to motion behaviours (patterns), various sets of qualitative and quantitative parameters of individual behaviours are able to be achieved. The analysis of motion pattern with a mathematical description that is able to be applied and to construct similar behavioural animation from an existing motion database. It also extends the ability for generating a new motion with different parameters from this reduced dimensional space.

4.2 Motion data acquisition and analysis

To test the PCA technique, an Ascension's Reactor2 active optical motion capture system is conducted to capture 20 people walking at speeds ranging from 0.5 to 3.0m/s in increments of 0.5m/s, and running at speeds ranging from 3 to 3.5m/s on a treadmill, according to their ability and fitness. After character mapping from the optical motion capture data (Xiao et al. 2008), each behaviour is able to be distinguished and segmented from the motion sequences. Five motion classes are chosen: strolling, normal walking, fast walking, walking to running, jogging and fast running. Using the prototype of motion, in general, the motion classes are classified as walking and running.

The motion sequences are segmented according to cycles (one cycle includes two steps, starting from the left-heel touchdown to the next left-heel touchdown). Due to the fact that all of the processed motion data is described as BioVision format, the joint angles are represented as discrete degrees in three rotational axes. In order to abstract the similarity and reduce the dimensions of motion, as opposed to quaternion spherical interpolation (Urtasun et al. 2004; Shoemake 1985), a linear interpolation method is used to re-sample the cyclic motion data at regular time intervals so that each example can be treated as $N=100$ samples of a motion starting at normalized time 0 and ending at normalized time 1. The properties of the obtained samples should be the same as the original one, or very close to the original motion data after motion data re-sampling. In order to check the consistency of re-sampling, in Figure 4.1 the original and re-sampled angle variation of hip joints in the time domain for a walk and running cycle are illustrated respectively. The variation of angular velocity and angular acceleration according to the walk cycle are also displayed. From the result shown in Figure 4.1, the linear interpolation for the resampled original motion data is applicable.

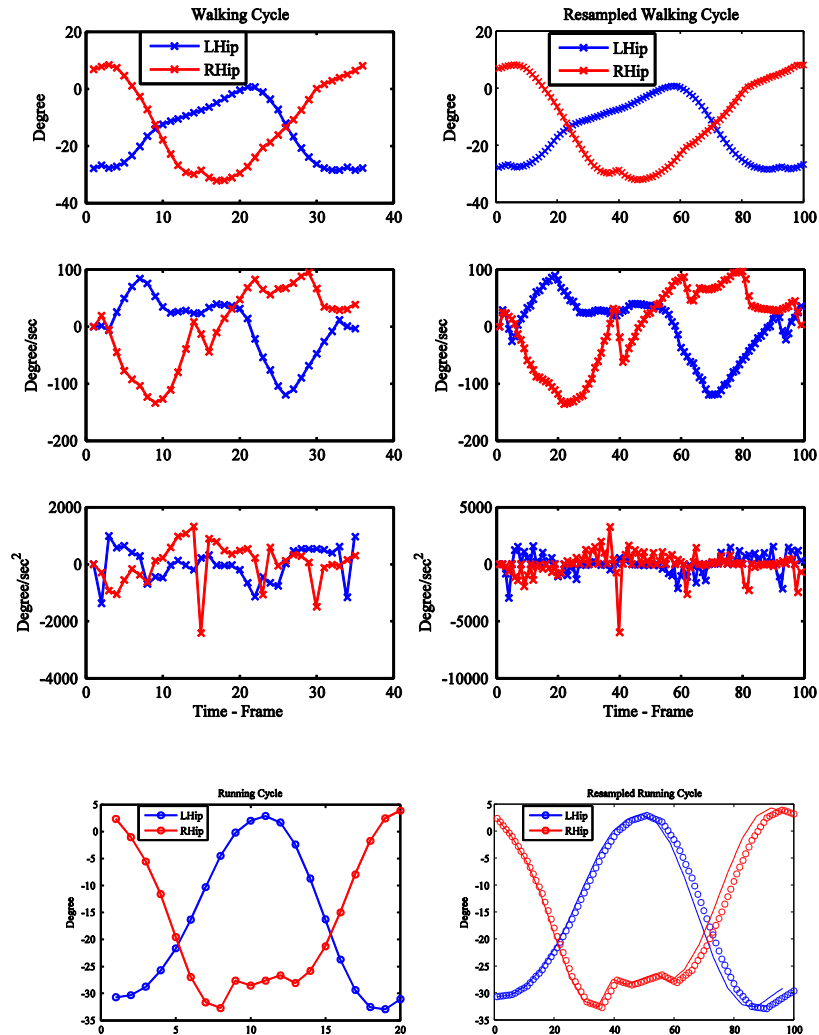


Figure 4.1 Segmentation and re-sampling of walking and running motion data in joint angle position within a cycle. Variation of angular velocity and acceleration is also shown in the walk cycle.

4.2.1 Principal Component Analysis (PCA)

In computer graphics the movement of a human character is described as a sequence of discrete parameters in the time domain as $Q(t) = \{p(t), \theta(t)\}$, where $p(t)$ is the translational position of the root joint, $\theta(t) = \{q_1(t), \dots, q_n(t)\}$ are the joint angles in the hierarchy. Given multi

motion sequences of the same behaviour, the PCA technique computes the estimation of $\theta(t)$ in a full-dimensional space by a linear combination of l basis vectors in a low l -dimensional as (Equation 4.1):

$$\hat{\theta}(t) = \theta_0 + \sum_{i=1}^l \alpha_i \theta_i^e(t) \quad (4.1)$$

where $\hat{\theta}(t)$ is the estimated motion after the PCA process, θ_0 is the mean motion, α_i are scalar coefficients of the eigenvectors θ_i^e of the motion.

To construct the PCA motion model, the mean motion θ_0 of individual pattern needs to be computed by

$$\theta_0 = \frac{1}{N} \sum_i^N \theta_i$$

where θ_i is the sampled motion segment, N is the number of samples.

Prior to obtaining of the eigenvectors coefficients of the motion, the covariance matrix C of the sampled motion data is computed by

$$C = MM^T \quad (4.2)$$

where $M = (\theta_i - \theta_0)$, $1 \leq i \leq N$, is the $N \times n$ motion matrix whose i^{th} row are the mean differences from the original motion samples.

The eigenvectors are obtained by solving matrix C through Singular Value Decomposition (SVD) (Press et al. 1992; Mathworks 2002) as

$$C = UDA^T \quad (4.3)$$

where U to be a $N \times n$ matrix, D as $n \times n$, and A as $n \times n$, both U and A has orthonormal columns, and D is a diagonal matrix contains the non-negative eigenvalues in a descending order. The principal components by projecting all the motions on the eigenvector matrix A is able to be achieved as

$$\theta^e = \alpha_i^T (\theta_i - \theta_0)$$

where α_i^T is the transpose of eigenvectors. Therefore, an approximated motion $\hat{\theta}$ by linear combination of l ($1 \leq l \leq n$) principal components with a mean motion is able to be constructed as Equation 4.1.

In order to evaluate the highest approximation of the sampled motion in low dimensional space, l is chosen as follows:

$$\varepsilon = \frac{\sum_{i=1}^l \lambda_i}{\sum_{i=1}^n \lambda_i} \quad (4.4)$$

where ε is the percentage of the motion data represented in 1-dimensional space, λ_i are the eigenvalues corresponding to the eigenvector computed by PCA. As Shown in Figure 4.2 (c) and Figure 4.2 (d), both walking and running can achieve 93% for the databases by given $l \simeq 10$.

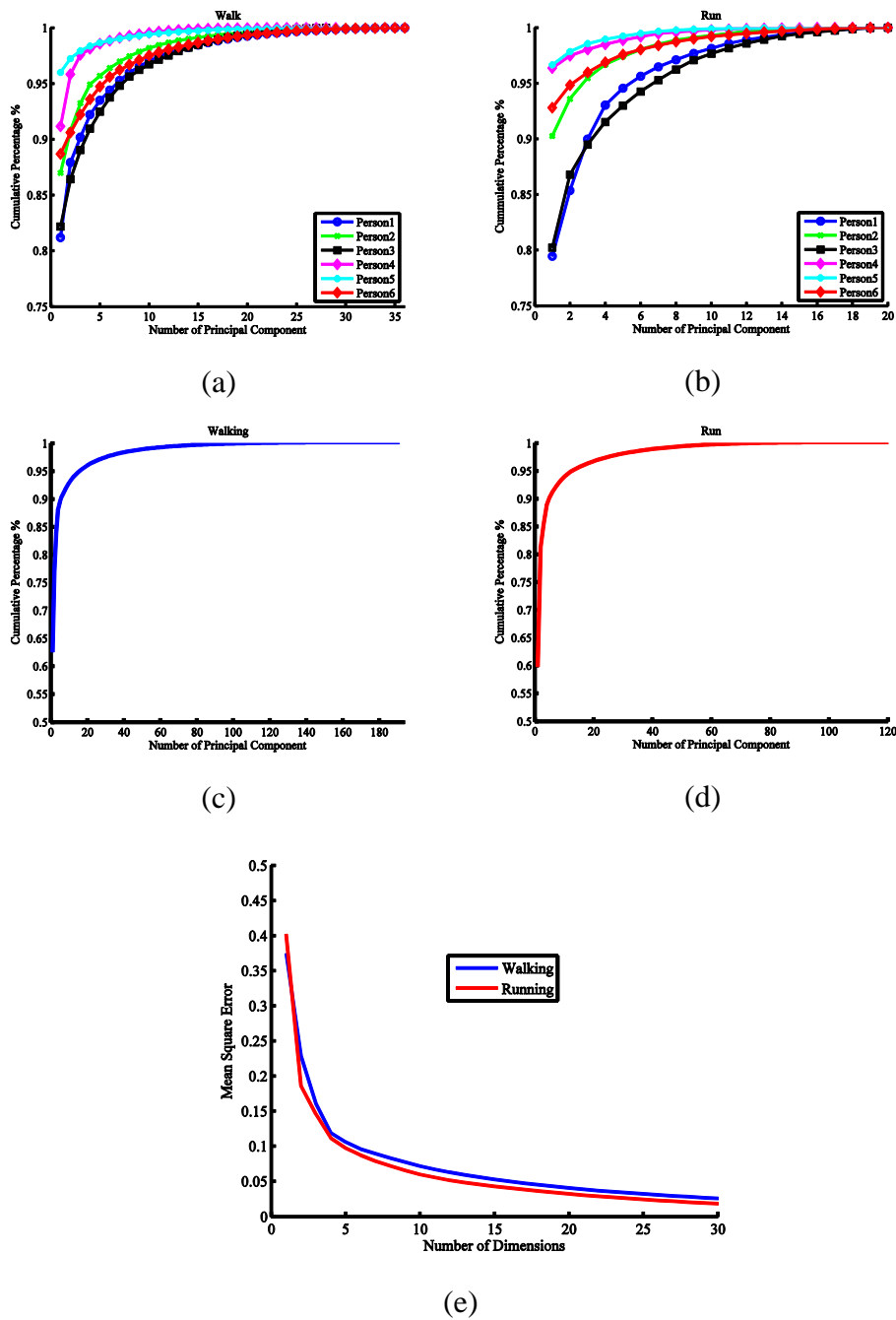


Figure 4.2 PCA analysis of human motion: (a) Walking that includes 6 subjects, 5 speeds and 1 cycle; (b) Running that includes 6 subjects, 4 speeds and 1 cycle; (c) Walking that processes 180 motions together from (a); (d) Running produces 120 samples together from (b); (e) Mean Square Error between a full-dimensional motion and the corresponding l -dimensional representation for walking and running behaviours.

4.2.2 Motion construction from motion database

Generating new motion from existing databases offers exciting new ways of creating character animation, and in this section, the creation of such data sets will be demonstrated.

In general, the existing motion database $\hat{\theta}(t)$ includes the motion data captured from different person p that is denoted by $\theta_i^p(t)$ ($1 \leq p \leq n$). In order to construct a new motion for an unknown person x , as shown in Equation 4.5, the scalar coefficients of the eigenvectors α_i^x of the motion θ_i^e need to be derived:

$$\hat{\theta}^x(t) = \theta_0 + \sum_{i=1}^l \alpha_i^x \theta_i^e(t) \quad (4.5)$$

where $\hat{\theta}^x(t)$ is the constructed new motion, α_i^x are the scalar coefficients corresponding to the existing eigenvectors of the motion.

Given a sequence of motion data from $\theta^x(t)$ performing certain movements such as walking at speed s_1 , a walking motion data in a different speed s_2 is wanted to be generated. First of all, as described earlier, the motion needs to be segmented in the same format for building the database. Then the segmented motion data is projected into an existing PCA space. As Urtasun et al. (2004) suggested using normalised Mahalanobis to take the distance between $\theta_{s_1}^x$ and $\theta_{s_1}^p$ in considering the proportion of each principal component to a corresponding eigenvalue. To maximize accuracy between these motion samples, the similarity to the existing data is measured by Mahalanobis distance as follows:

$$d(\theta_{s_1}^x, \theta_{s_1}^p) = \sqrt{(\theta_{s_1}^x - \bar{\theta}_{s_1}^p)^T C_p^{-1} (\theta_{s_1}^x - \bar{\theta}_{s_1}^p)} \quad (4.6)$$

where $\bar{\theta}_{s_1}^p$ is the mean value of person p walking at speed s_1 , C_p is the covariance matrix computed from the existing motion samples of person p as described in Equation 4.2.

According to the computed distance, how close or similar the tested samples to the sample set in the database can be defined. The similarity between motion θ^x and θ^p is able to be evaluated by a weight function considering the closeness of the tested samples within the database as follows:

$$w^{x,p} = \frac{d(\theta_{s_1}^x, \theta_{s_1}^p)^{-1}}{\sum_{p=1}^n d(\theta_{s_1}^x, \theta_{s_1}^p)^{-1}} \quad (4.7)$$

From the obtained weights of s_1 and considering the similarity in the existing database, the approximate scalar coefficients α_i^x for speed s_2 are able to be obtained as follows:

$$\alpha_i^x = \sum_{p=1}^n w^{x,p} \alpha_i^{p,s_2} \quad (4.8)$$

And the constructed new motion can be achieved by applying Equation 4.5.

4.2.3 Cyclic motion synthesis

After applying motion data analysis of an individual person, a relationship between various motion speeds and coefficients of the Principal Components is able to be constructed. Figure 4.3 to Figure 4.6 show two types of motion: walking and running, with the speed corresponding to the coefficients of the Principal Components. As illustrated in Figure 4.3, it is observed a very clear linear relationship between speed values and corresponding coefficients.

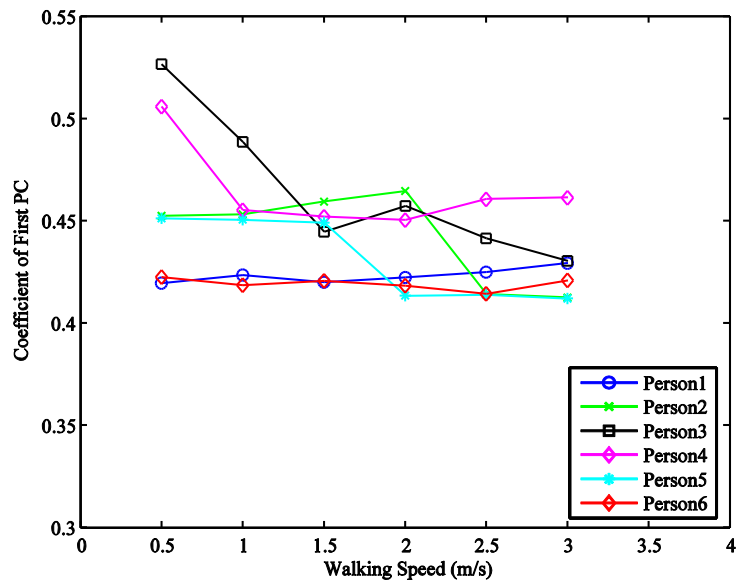


Figure 4.3 Walking speed corresponding to the coefficient of the first Principal Component from 6 people's motion data.

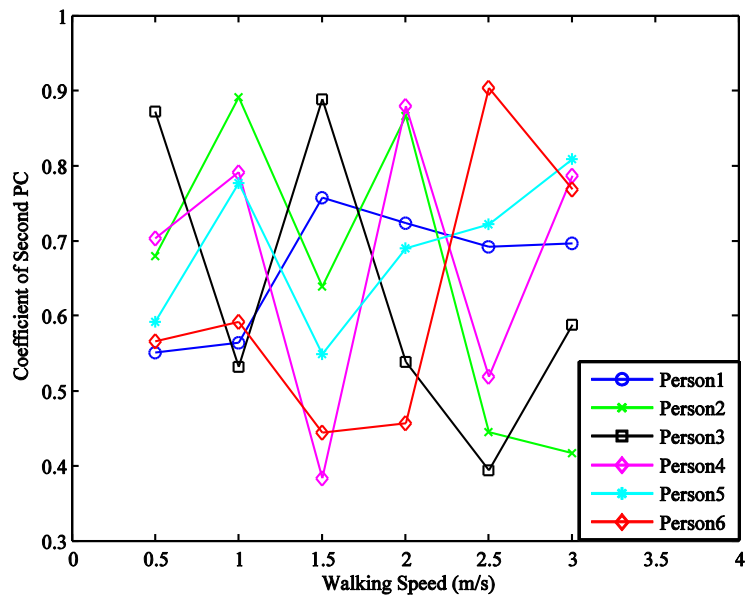


Figure 4.4 Walking speed corresponding to the coefficient of the Second Principal Component from 6 people's motion data.

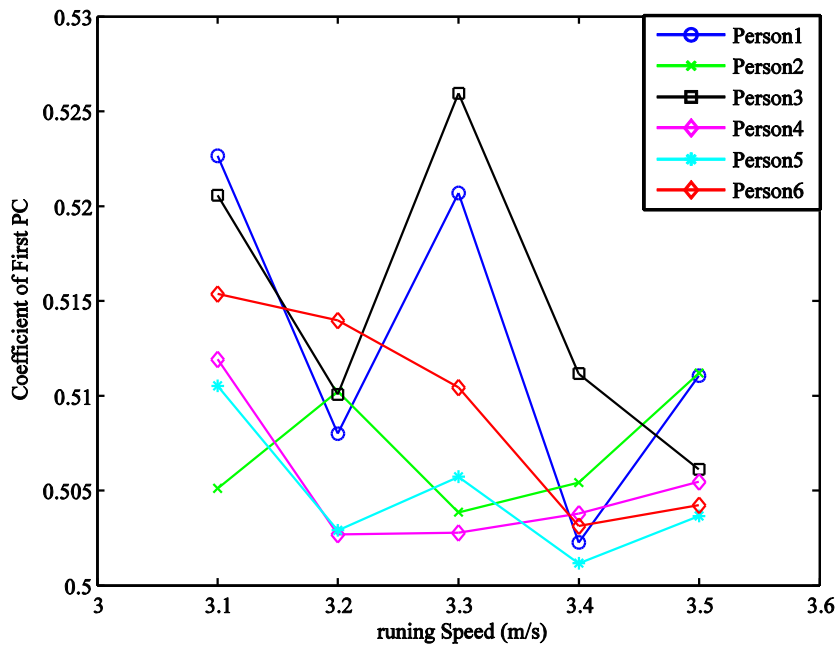


Figure 4.5 Running speed corresponding to the coefficient of the first Principal Component from 6 people's motion data.

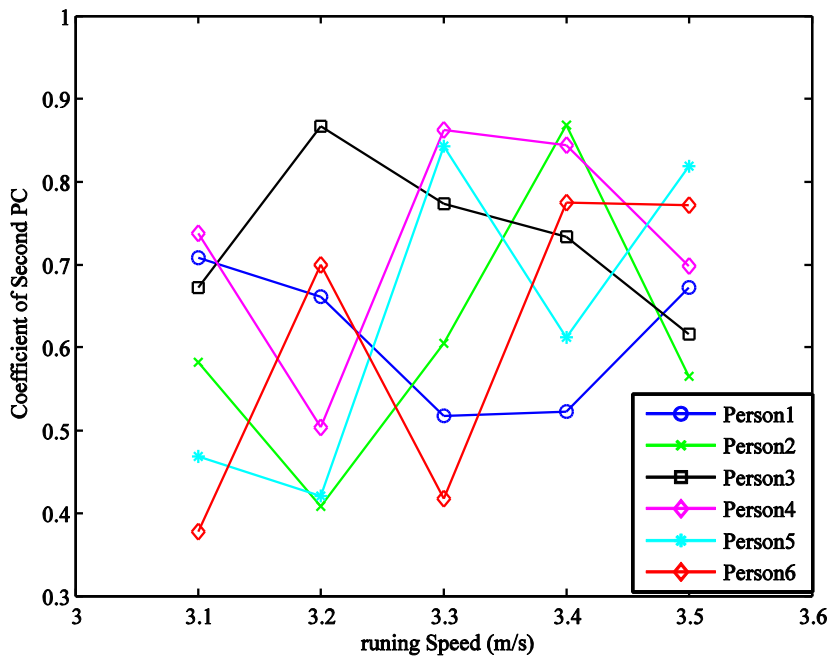


Figure 4.6 Running speed corresponding to the coefficient of the second Principal Component from 6 people's motion data.

In theory, after achieving an approximation function, a motion can be formed by interpolation and extrapolation of the coefficients of each PCs. These obtained coefficients can then be applied to Equation (4.1) to construct a new motion according to the motion database. However, the constructed new motion is limited to the current database. To extend the method for an arbitrary human character, further analysis of the motion data is needed.

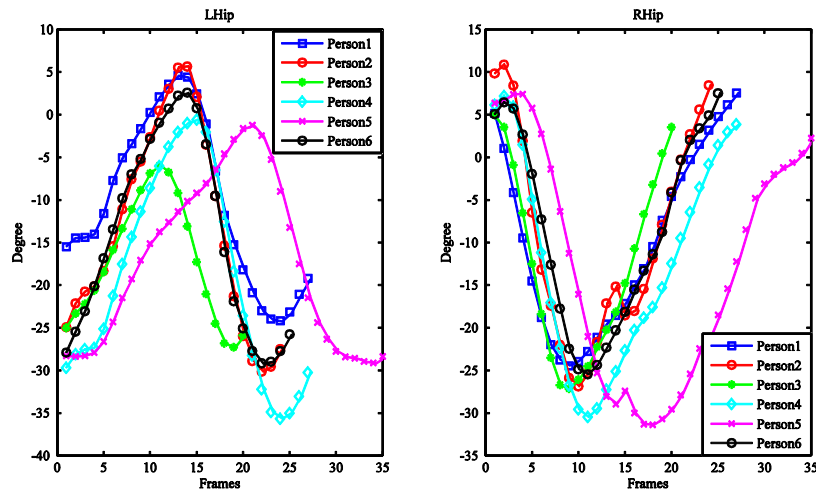


Figure 4.7 Walk cycle: two joints angle comparison from 6 different persons.

As shown in Figure 4.7, two hip angles in the sagittal plane show very similar trajectories in a walk cycle from six people. For a similar movement, all of the people have the same pattern. Regarding the joint angle trajectory as a wave, the obvious differences are the phase, amplitude and wavelength. For a cyclical wave, frequency f is defined as

$$f = \frac{v}{\lambda} \quad (4.6)$$

where v is the velocity of wave, and λ is the wavelength.

For human motion, Inman et al. (1981) have presented a polynomial equation as follows:

$$f = 0.743\sqrt{V} \quad (4.7)$$

where V is the normalised speed, which can be achieved by applying Equation 4.8

$$V = \frac{\tilde{v}}{H} \quad (4.8)$$

where \tilde{v} is the walking speed, and H is the hip joint height.

Similar to Glardon et al. (2004), Inman's formula is also adapted in this research experiment but in the form aV^b . During the data acquisition procedure, each walk cycle was segmented and the motion data was re-sampled using a normalised time step. After curve fitting by function aV^b , the frequency function as $f = 0.878V$ is achieved. Figure 4.8 shows the frequencies with respect to normalised speed.

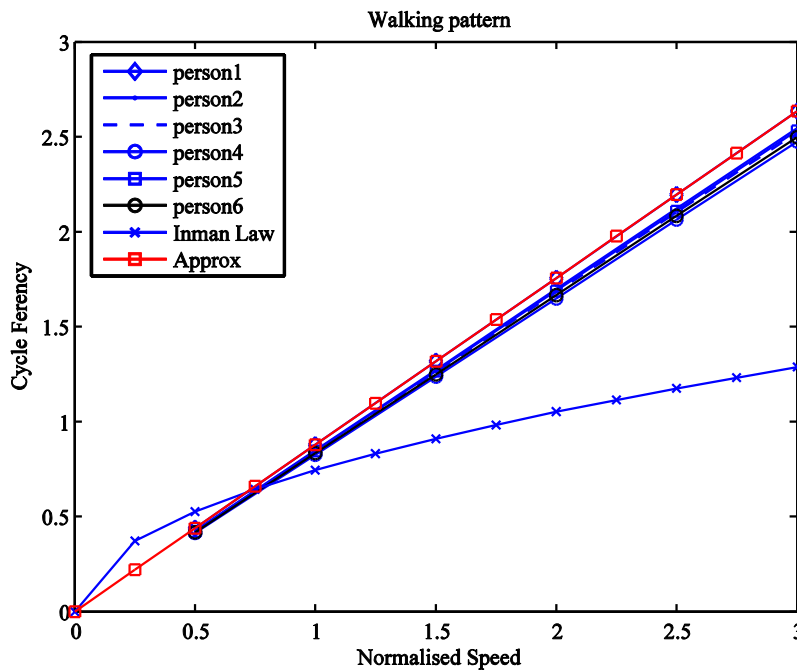


Figure 4.8 Walking frequency function corresponding to normalised speed.

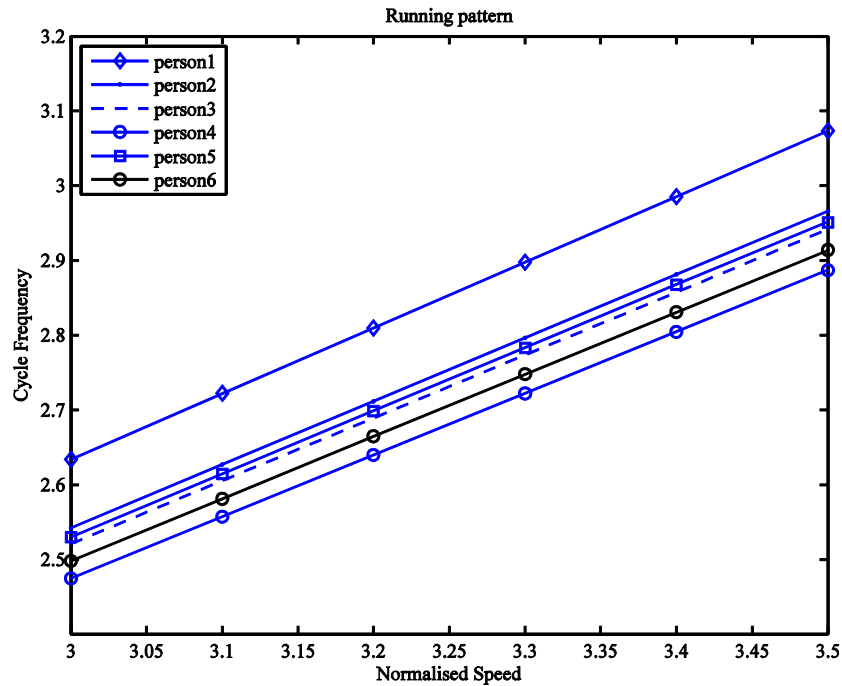


Figure 4.9 Running frequency function corresponding to normalised speed.

To construct a cyclic motion according to a frequency function, the period T of cyclic motion is obtained using $T = \frac{1}{f}$. Furthermore, phase variation $\Delta\theta = \frac{\Delta t}{T}$ in one cycle of movement also can be obtained.

Comparing the research output with Equation (4.7), the obvious difference is the factor of velocity. As the walk cycles shown in Figure 4.7, each individual walk cycle has a very similar trajectory. They are performed in a very stylistic and standard form. As shown in Figure 4.9, the frequency of running cycle shows linear variation as well. The reason for the deviation from Inman's law is that the motion data are captured from a treadmill where the movement is limited to $\approx 1.2m$ long space, whereas the person on the treadmill always walks or runs inside the $1.2m$ space based on a similar pace. Although different movements can be captured from a treadmill for motion analysis, the result is far different from the result of motion data is

captured from normal performance, e.g. no space limitation for the movement. In order to overcome such a drawback in the future, finding an appropriate method for motion capture is demanding so that it will help animators to achieve high quality motion data for motion analysis or other applications.

4.3 Transition from walking to running

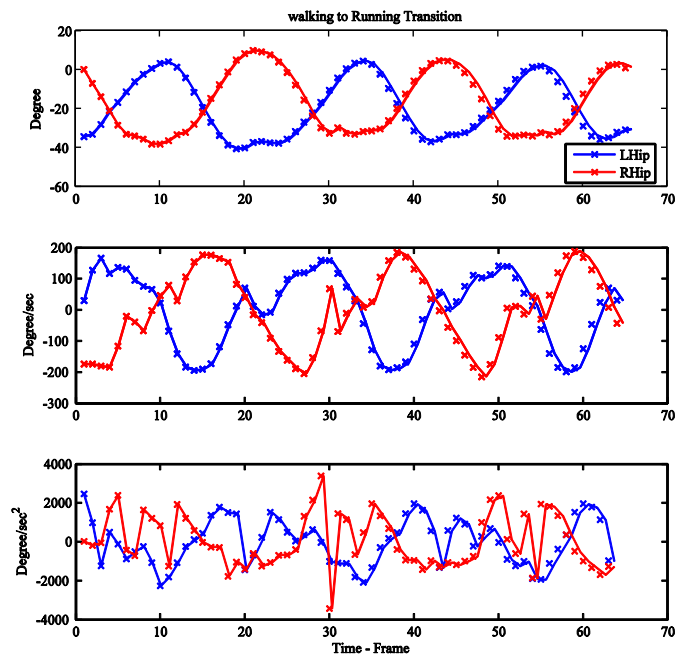


Figure 4.10 The analysis of joint angle position, angular velocity and angular acceleration during the transition from walking to running.

During motion data acquisition, a sequence of movement is captured from walking to running. Unfortunately, the segmented transition from walking to running cannot be applied by Principal Component Analysis method since it only takes place within one single period of time. However, such a transition is able to be analyzed according to its pattern based on each individual movement. This transition happens in the period of time in

which the movement changes from two support points during walking to one support point when running is started. Figure 4.10 shows the parameters of joint angles during this transition. From the motion data point of view, it is difficult to distinguish running data from walking data. Both of them perform a very similar style. Although this transition motion is unable to be applied by PCA, based on the research finding, diagrams which illustrate the speed variation according to the PC analysis of walking and running is able to be constructed as shown in Figure 4.11 and Figure 4.12. From the illustrated Figure 4.11, it shows that the speed of motion is continuous, but the motion pattern changed completely according to the first principal component.

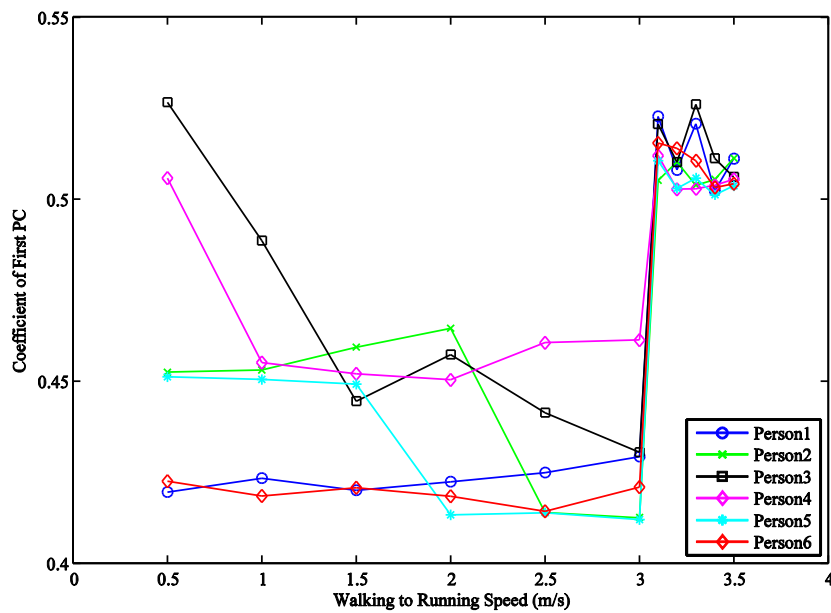


Figure 4.11 Speed variations from walking to running corresponding to the coefficient of the first Principal Component from 6 people's motion data.

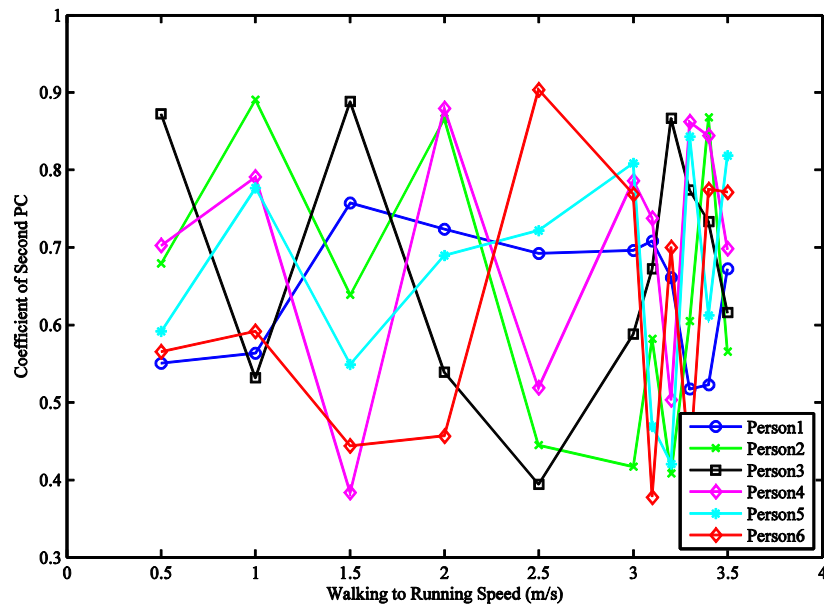


Figure 4.12 Speed variations from walking to running corresponding to the coefficient of the second Principal Component from 6 people's motion data.

4.4 Experiment result

In this research experiment, the PCA space includes walking and running motions that are captured and processed from six people out of a possible twenty. After carefully segmenting the motion clips, the motion data is re-sampled and normalised to 100 frames. For each individual subject, the PCA space has 20 samples for running and 36 samples for walking. The walking data includes six different speeds from 0.5m/s to 3.0m/s in increments of 0.5m/s and running from 3.0m/s to 3.5m/s. The built PCA space of individual subjects contains up to 2MB of motion data for running and 3MB for walking.

The motion database is built from different people in different speeds. According to the speed variations or the personal behaviours in the database, the same motion is also able to be constructed from different principal component spaces as shown in Figure 4.13.

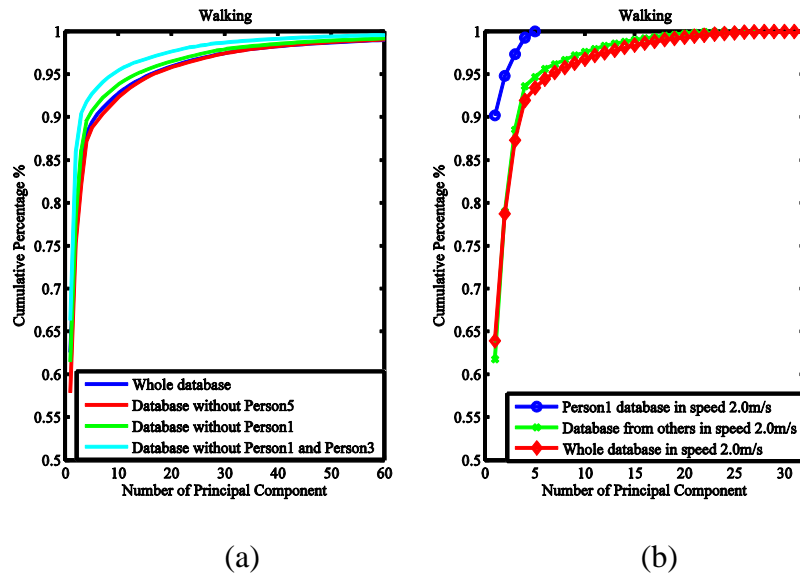


Figure 4.13 Principal Component Analysis of walking motion: (a) Walking motion construction from different motion database; (b) Construct same speed walking motion according to different resources.

As illustrated in Figure 4.2, about 10 PCs can represent more than 95% of the total information. For an individual subject, two to four PCs can represent over 90% of the total information. According to the analysis, constructing different motions by PC coefficients could be easily interpolated and extrapolated. Also, a similar motion for different size skeletons could be constructed.

In this experiment, two different size skeletons have been chosen in which the tall skeleton has a 1 meter hip joint height and the short skeleton has 0.7 meter hip joint height. Both human skeletons walk at a speed of 2.5 m/s and run at the speed of 3.5m/s. Figure 4.13 shows the examples constructed for different size skeleton movements from the motion database. Figure 4.14 (c) shows the frequency change between these two walking skeletons. In order to walk at the same speed, the short skeleton has to walk faster to catch up with the taller skeleton.

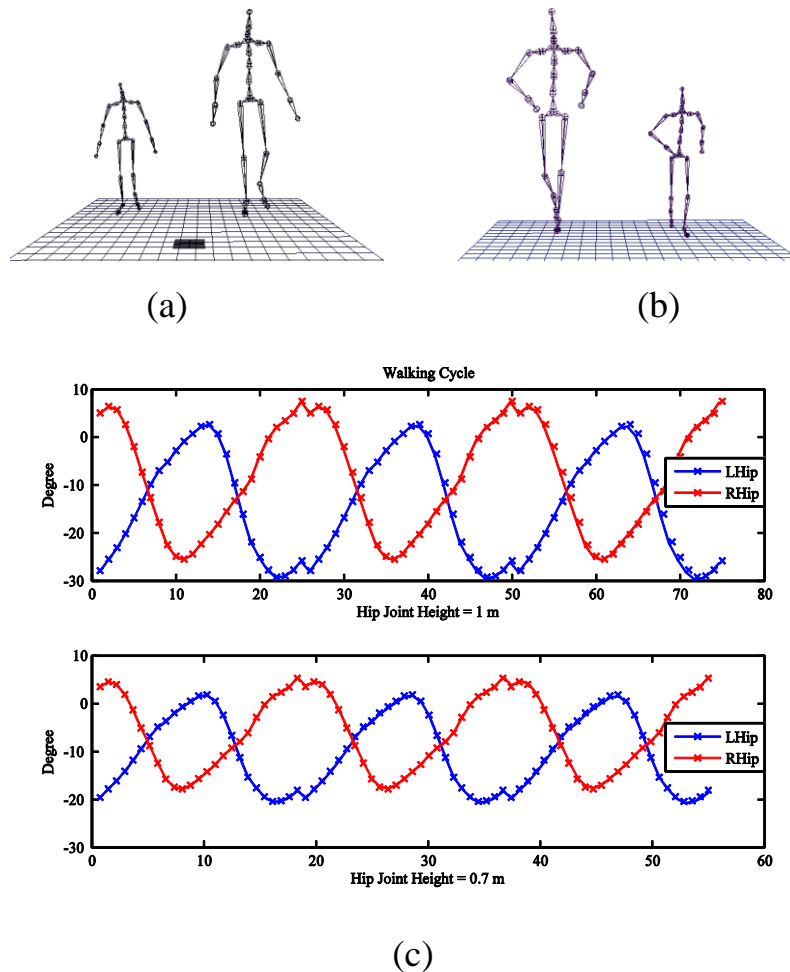


Figure 4.14 Two different size skeletons with identical movement in same speed: (a) Walking cycle; (b) Running cycle; (c) Walking cycle analysis at the same walking speed from top left.

4.5 Discussion

As described above, there are only two cyclic motion patterns that have been analysed. The presented method is statistically based and it approximates the solution according to similarity and probability from an existing population. By employing an existing motion database, a similar animation of different size human characters is able to be generated. In addition, for the same person, the motions with different speeds can also be interpolated and extrapolated from the database.

Although there is a transition from walking to running in a captured motion sequence, it only takes place in a short period of time in which human movement changes from two support points during walking to one support point when running begins. Such transitions from walking to running within each motion sequence can be segmented and can be analysed according to the analysis of walking and running motion data separately. However, this transition behaviour is unable to be analyzed by using PCA method. If such a pattern could be built into the PC space, it would help animators to create smoother transitions from one movement to another. In order to extend the utility of re-using previous motion capture data, a database that includes more patterns should be built into the PC space.

From the experiment, it has been discovered that by using a treadmill, similar behavioural movement is able to be captured in a wide range of speeds. However, the person on the treadmill is limited by the moving space. His or her reaction is different from their performance during normal movement. In order to obtain accurate information of motion analysis, the motion data should be obtained from people's normal performance.

4.6 Summary

In this chapter, Principal Component Analysis method has been examined and applied for abstracting human motion in low dimensional. It provides a generic method that mathematically describes different human motions and is able to be used to construct similar behavioural motion from existing motion database. It shows that only a few Principal Components could be represented nearly 90% of a unique motion pattern. It provides animators with a technique to generate similar behaviour motions by setting various parameters from the generic motion pattern. It also extends the ability of re-using an existing motion database efficiently. By applying PCA, a stylistic motion is able to be interpolated and extrapolated from an existing motion database. It provides animators with an alternative method to

generate any intermediate motion from a given motion database. Such a technique will support a further research in the following chapter.

As stated above, it is very important to improve the technique for data acquisition as well as for motion segmentation, since manually segmenting each motion sequence and building corresponding motion database is time-consuming. On the other hand, if the treadmill is the only available facility that can be relied on for certain motion data collection, more careful monitoring should be carried on to the person during motion capturing since their characteristic might change, and this might affect the quality of the motion data.

CHAPTER 5

MOTION TRANSITION BY

INCOMPLETE MOTION DATA

RECONSTRUCTION

To create a final animation production, animators need to edit and synthesize motion capture data according to a specific scene or storyboard. Normally, they consist of several short motion data sequences that arise due to the limitation of the motion capture system. To connect such short motion clips together for the final character movement, a transition technique is required to generate a smooth motion sequence. The consequence of generated transition motion should not only guarantee the smoothness of motion data but also satisfy the variational consistency of physical properties of a moving human character.

5.1 Introduction

Realistic human-like character animation depends upon many elements, such as a good story, attractive character(s), realistic movements, etc. The storyboard shown in Figure 5.1 (Parent 2002), shows how animators describe every motion that appears in each shot, which lends itself to motion capture techniques. However, connecting these movements together

to create a smooth continuous motion transition is vital for high-quality animation.

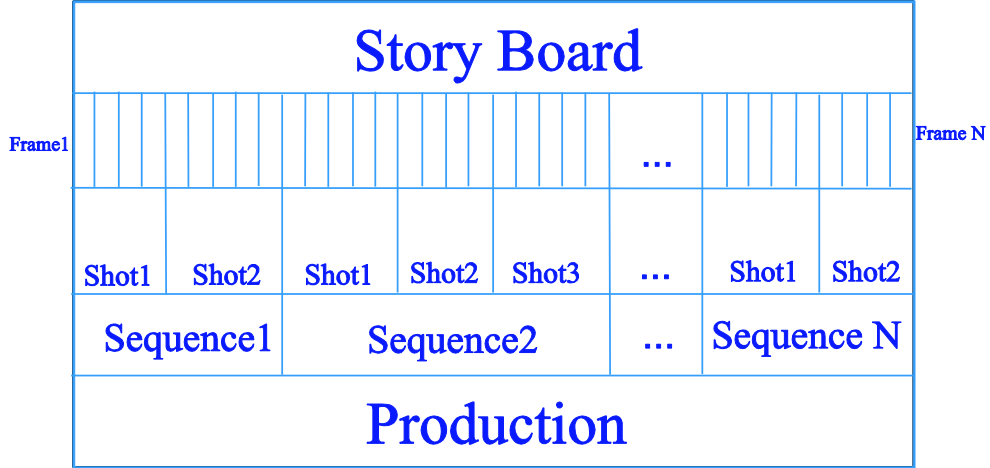


Figure 5.1: Story board organizations for a 3D Animation product from Parent (2002).

To integrate different shots, especially when they involve different movements, animators employ an ease-in and easy-out linear function to interpolate the motion clips according to different weight coefficients. The motion transition is interpolated as a sum of two motion curves as described in (Perlin 1995; Witkin and Popović 1995; Kovar et al. 2002a; Safonova and Hodgins 2005):

$$\theta(t) = w(t)\theta_1(t) + (1-w(t))\theta_2(t) \quad (5.1)$$

where $\theta_1(t)$ and $\theta_2(t)$ are the motion curves being blended, $w(t)$ is a normalized weight function and $\theta(t)$ is the new motion.

Traditionally, when interpolating between different movements, animators have to manually find a contact point in order to constrain two movements together and choose the duration in time domain where the two motion curves are aligned (Witkin and Popović 1995). For example, setting left foot as a constraint for both walking and running motion sequences (Figure 5.2). Then a dynamic time warping method is employed to find the closest

distance between these two curves and match the duration within motion transition (Bruderlin and Williams 1995; Kovar and Gleicher 2003).

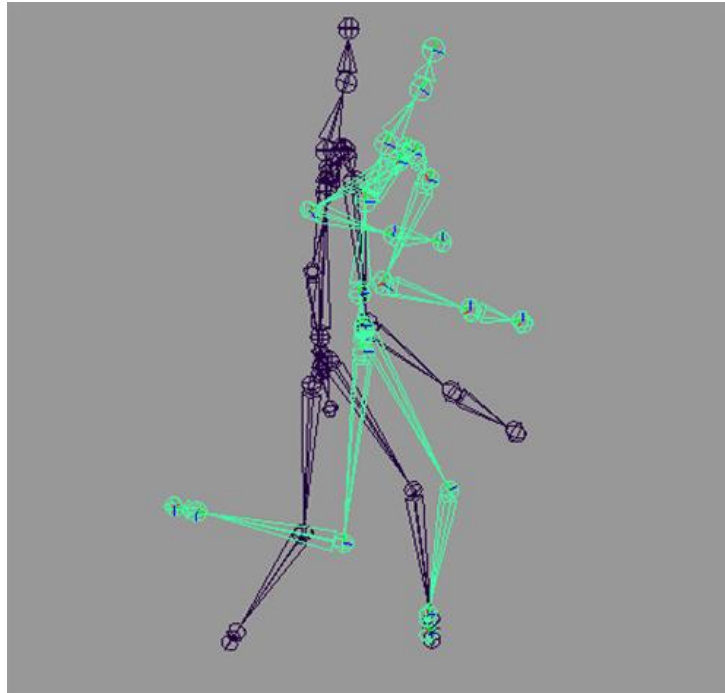


Figure 5.2 Pose constraints for employing motion blending.

5.2 Problem description

Even though linear blending can produce smooth and continuous motion data, the blended motion may introduce sudden or unrealistic movements, especially for unusual dynamic movements. One of the main reasons that affect the final blending result is that these blended motions have very obvious differences in physical properties, i.e. velocity. A sudden movement has been introduced to a blended running motion sequences begin from frame 25 to frame 37 see (Figure 5.3 and Figure 5.4).

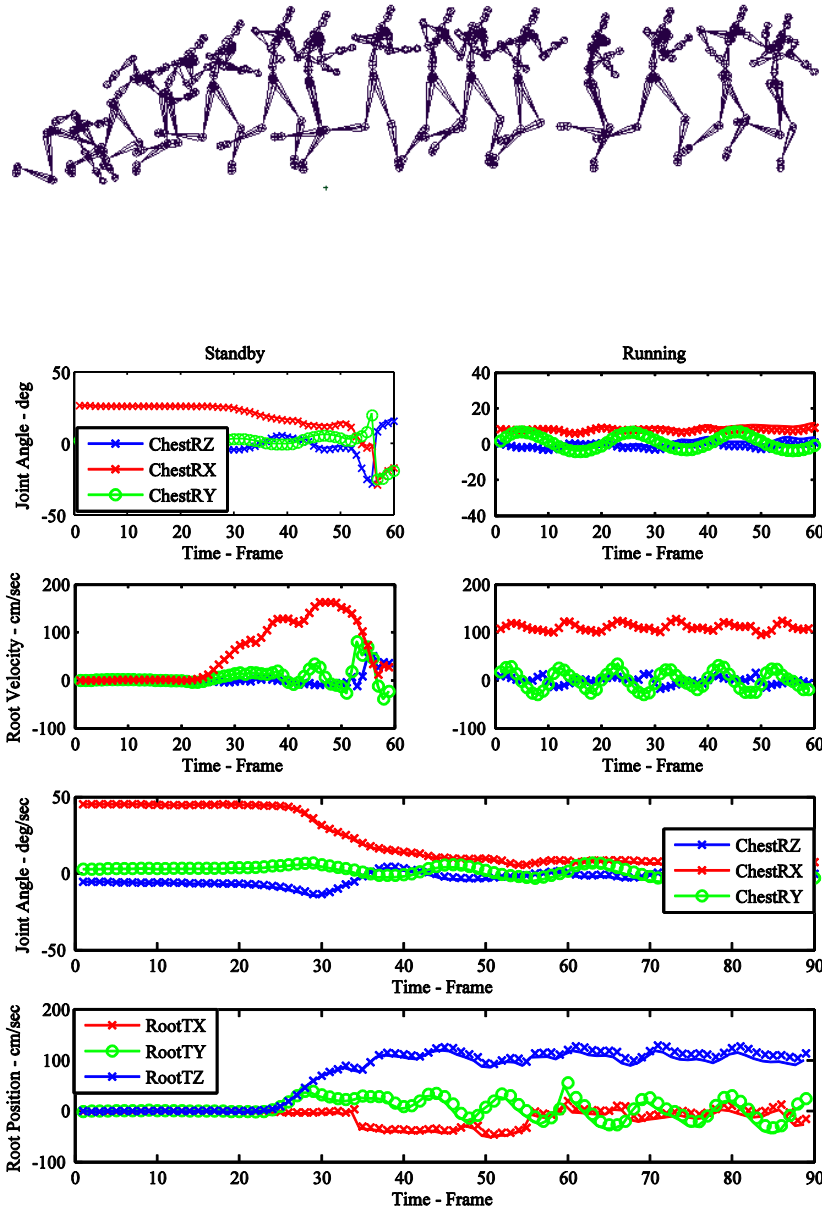


Figure 5.3 (Top) Running motion sequences from blending of standby and run motion clips. (Bottom) Motion analysis before and after motion blending from these two motions.

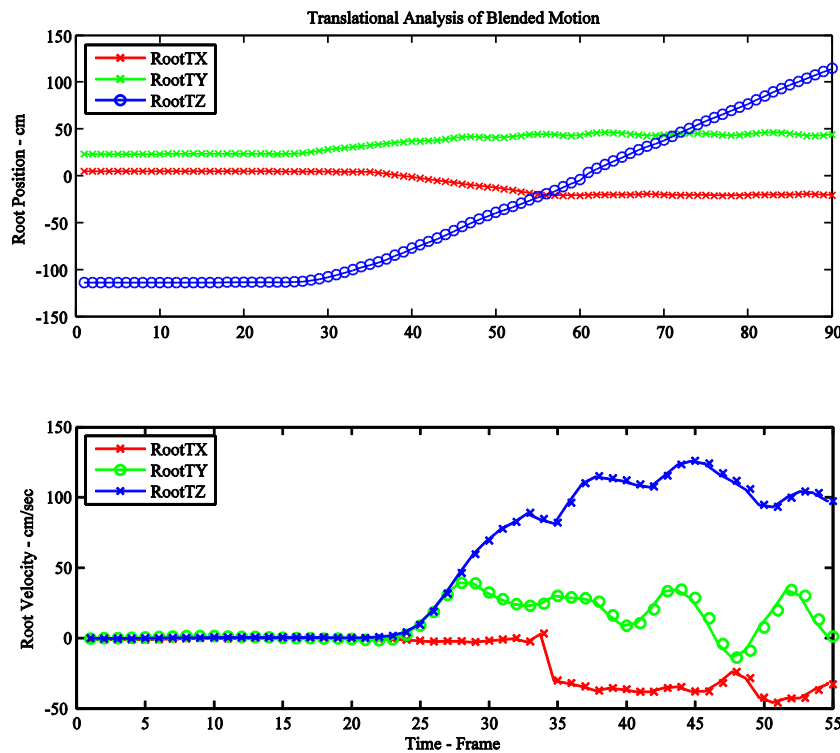


Figure 5.4 Translation analysis of root joint after motion blending includes position and velocity.

Even though results show that analyzing the physical properties of human movement by Safonova and Hodgins (2005), the interpolation function is able to produce more natural and realistic looking motion subject to a certain physical corrections and modifications. Rather than correcting blended motion by modification of physical properties at a later stage, it is very convenient for an animator to blend a motion or create a transition motion according to existing physical available motions.

The concept of Incomplete Motion Data Reconstruction (IMDR) originates from Image Inpainting and Texture Synthesis which consider isophotes as an objective function to restore a damaged image, or the removal of unwanted elements from a picture (Bertalmio et al. 2000). Individual people possess very unique styles of motion, and the behaviours from each

other can be easily distinguished such as walking, running, jumping, dancing etc. During every-day life people easily and smoothly transition from one motion to another. It is imaginable that the motion transition is based on some physical condition such as velocity, acceleration etc. To construct a long and continuous motion sequence from several short motion clips, it is necessary to build convincing motion transitions between each motion clip. By applying PCA it is possible to reduce a high dimensionality of motion to a low dimensional motion without losing too much information. Despite the fact that PCA is used for synthesising motion and extrapolating stylistic motion from a motion database (Safonova et al. 2004; Urtasun et al. 2004), abstracting appropriate motion sequences and re-using a motion database efficiently for motion data reconstruction still relies upon the analysis of motion data.

From the description of motion analysis in Chapter 4, a large amount of information can be obtained from different resources, e.g. velocity of starting and ending motion clips, etc. In contrast to linear blending for motion transition, the presented method is to find proper motion data or a motion prototype from a motion database to reconstruct the transitional motion according to an animator's requirements.

5.3 Incomplete motion data reconstruction

The presented IMDR technique is based upon the analysis of motion data and requires a trajectory path defined by an animator (Xiao et al. 2009a). It provides animators with a very convincing technique for seamless three-dimensional character motion synthesis. To fulfil the incomplete motion data reconstruction for a final animation production, there are three steps: first, the given motion sequence along with the animator's plan is analyzed for obtaining the character information such as height of skeleton, prototype of motion; second, a new motion is constructed from an existing motion database according to the animator's direction; third, a time alignment

method is applied to smooth the conjunction of original motion data with new generated motion data.

5.3.1 Motion analysis from a given motion sequence

A motion sequence defined by an animator contains motion information that has to be interpreted for the further motion reconstruction, i.e. speed, length of stride, etc. Also, further information about the planning of an animation sequence is needed, for example, the constructed path of the final sequence.

5.3.1.1 Motion data interpretation

To interpret motion data comprehensively, different methodologies are applied to complete the task. As Figure 5.5 shows, three stages are included before the final motion data analysis: computer graphic display, text file information and motion data plotting.

From the computer graphic display, the movement and the structure of the character as a skeleton profile are illustrated. In this example, the behaviour of its movement can be distinguished such as walking. From the text file of the motion data, joints that have been applied for construction of the character and its sequence in the hierarchy are able to be clarified; and the length of the each link between adjacent pairs of joints and the type of each joint are also able to be identified. Before plotting the motion data, the information of how many frames of the movement and the duration between each frame also can be obtained from the file. From the plotting of three-dimensional position data at the root joint, it shows the travel direction is almost along $+z$ direction. Further extra information can be achieved from the data plotting, i.e. cycles, time, etc. By applying all of the available information and knowledge, it allows animators to interpret a motion comprehensively.

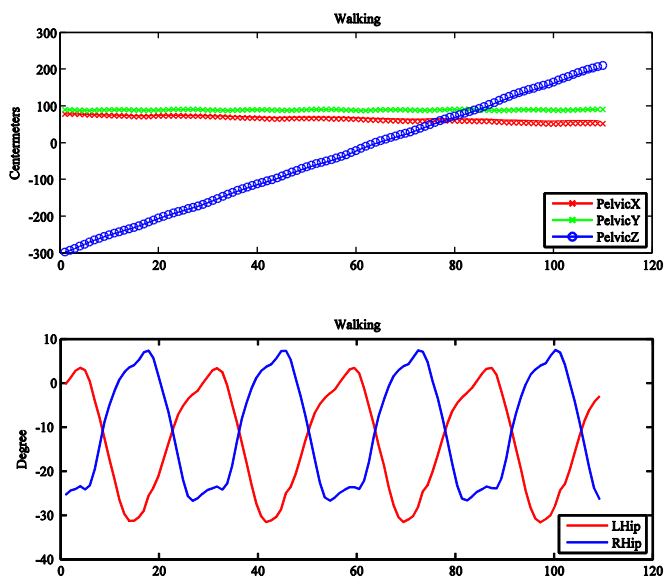
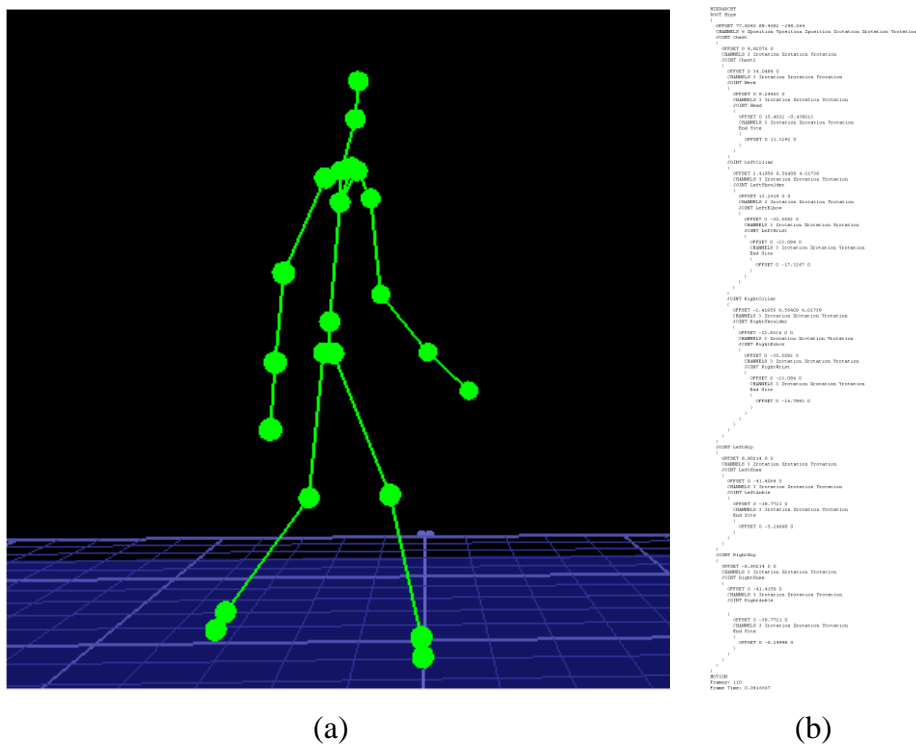


Figure 5.5 Motion data interpretation from three areas. (a) Computer graphics skeleton display. (b) Motion data file format. (c) Motion data display.

5.3.1.2 Motion path design and analysis

The path design or analysis of the animation path planning is a very important procedure to complete the presented IMDR. Since the approach is to construct a missing data sequence between two motion clips, in order to find an appropriate motion data, an animator has to define the sketch of the movement between the given motion clips. An example of path design is shown in Figure 5.6.

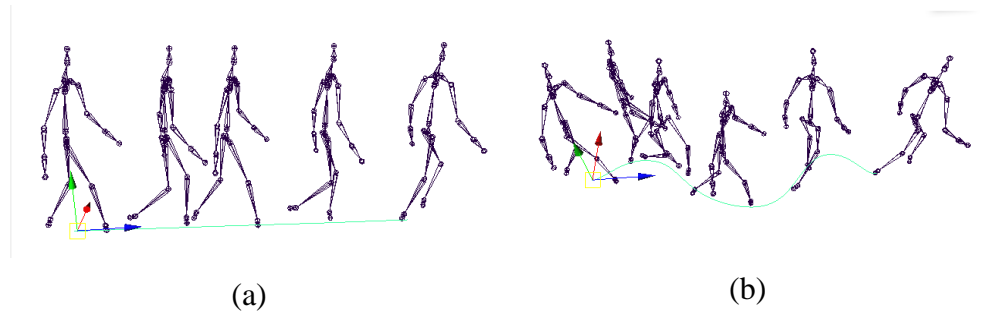


Figure 5.6 Motion path planned for IMDR. (a) Walking along a direct line; (b) Walking along a curve.

From the sketch of the defined path, the starting velocity v_s , position p_s and ending velocity v_e , position p_e are able to be found from the given motion clips. In order to build an appropriate motion data for this sketch, it is necessary to know further information about the movements involved in the motion sequence such as the time (frame) of this motion sequence, type of movement, etc.

5.3.2 Motion reconstruction

According to the given motion data $\theta^x(t)$ supplied by animators such as type of movement, velocity v_{s_1} , position p_{s_1} , velocity v_{e_1} etc, combining with the information for the path design, it is able to build an intermediate motion sequence v_{s_2} or even change the speed from v_{s_1} to v_{e_1} . Recall the techniques described in early chapter, the given motion data is able to be projected into an existing PCA space $\theta^p(t)$ built from different resources.

Mahalanobis distance is applied to measure the similarity among the existing data; and it provides animators with a criterion for choosing the proper motion:

$$d(\theta_{v_{s_1}}^x, \theta_{v_{s_1}}^p) = \sqrt{(\theta_{v_{s_1}}^x - \bar{\theta}_{v_{s_1}}^p)^T C_p^{-1} (\theta_{v_{s_1}}^x - \bar{\theta}_{v_{s_1}}^p)} \quad (5.2)$$

where $\bar{\theta}_{v_{s_1}}^p$ is the mean value for person p walking at speed s_1 , C_p is the covariance matrix computed from the existing motion samples of person p .

The computed distance provides animators a means to evaluate the closeness of the given data within the database by a weight function

$$w^{x,p} = \frac{d(\theta_{v_{s_1}}^x, \theta_{v_{s_1}}^p)^{-1}}{\sum_{p=1}^n d(\theta_{v_{s_1}}^x, \theta_{v_{s_1}}^p)^{-1}} \quad (5.3)$$

Then, the obtained weights in velocity v_{s_1} enable it to be applied into PCA space where motion data contains velocity v_{s_2} in the same motion. The approximate scalar coefficients α_i^x are obtained as follows:

$$\alpha_i^x = \sum_{p=1}^n w^{x,p} \alpha_i^{p,s_2} \quad (5.4)$$

The new motion based on the motion analysis and path design can be constructed from

$$\hat{\theta}^x(t) = \theta_0 + \sum_{i=1}^l \alpha_i^x \theta_i^{p,e}(t) \quad (5.5)$$

where $\hat{\theta}^x(t)$ is the constructed new motion in joint angle space, θ_0 is the mean motion of the data set, $\theta_i^{p,e}$ is eigenvectors of the motion, α_i^x is the scalar coefficients correspond to the existing eigenvectors of the motion.

5.3.3 Time alignment

In order to generate a smooth transition between two motion clips, it is important to find the spatial differences between them. A time alignment process, dynamic time warping (Bruderlin and Williams 1995; Kovar and Gleicher 2003), is taken to compare these two motion clips frame by frame.

To calculate the distance $d(q_1^i, q_2^i)$ between motion θ_1 and θ_2 at i^{th} frame, a time path minimization is performed to corresponding frame pairs for the aligned two motions:

$$D(\theta_1, \theta_2) = \min \sum_{f=1}^n d(q_1^i, q_2^i) = \min \sum_{f=1}^n \|q_1^i - q_2^i\|^2 \quad (5.6)$$

where $D(\theta_1, \theta_2)$ is the sum of distance between aligned two motions, f is the frame index from 1 to n .

According to the chosen motion clips, it is able to constrain them together and create a grid where columns and rows correspond to frames from first and second motion respectively. The dynamic time warping algorithm will automatically calculate the distance of corresponding frame pairs in each cell for every DOFs in the hierarchy. An example of dynamic time warping is depicted in Figure 5.7. The red line within dark area corresponds to an optimal time alignment between given motions.

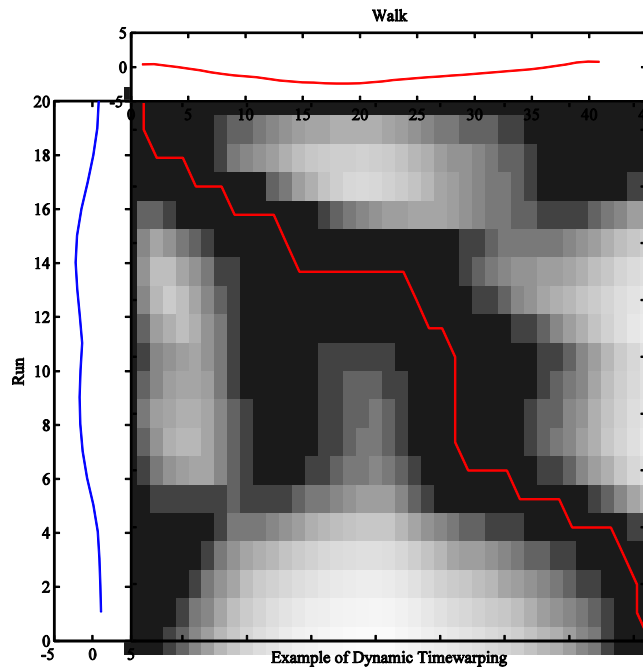


Figure 5.7 Example of dynamic time warping applied to a walking motion and a running motion.

5.4 Experiment results

5.4.1 Walking along a curve path

In this experiment, a motion sequence of person5 with a walking speed of 1.5m/s is given as a reference motion, and a walking motion at a speed of 2.5m/s is needed to be constructed from the database without the data being captured from person5. The final animation plan is shown in Figure 5.8. There is missing data according to the animator's path design. In order to construct a long and continuous motion sequence, the existing motion need to be analyzed and the proper motion data need to be constructed from a previously built motion database to fill the gap.

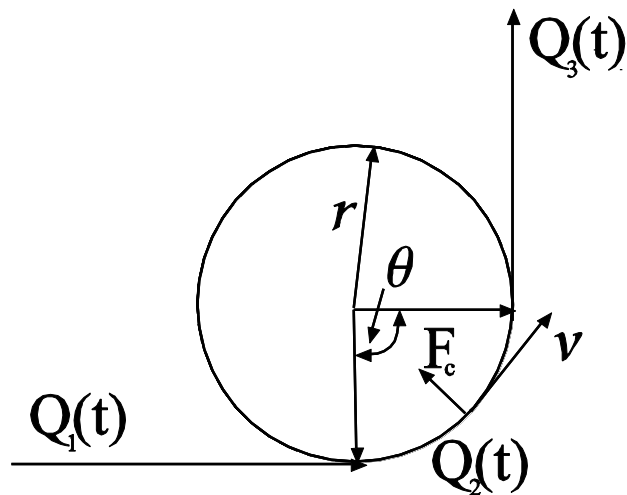


Figure 5.8 Construct a turnaround motion, where $Q_1(t)$ is the first motion clip; $Q_2(t)$ is the required construct motion clip; $Q_3(t)$ is the second motion clip, r is radius of circle; θ is the rotation angle covering the missing gap, F_c is the centripetal force and v is the velocity.

First, the prototype of the given motion data is analyzed, including the velocity and foot step, etc. According to the trajectory path designed by the animator or the walking speed and root trajectory of a human character in

the first and the second motion clips, an approximate path for the root trajectory for the missing data gap is able to found by

$$r = \frac{1}{\kappa}$$

where r is the radius of curvature, κ is the curvature can be obtained as follows:

$$\kappa = \frac{d\phi}{ds}$$

where ϕ is tangential angle, s is the arc length.

Second, according to the given motion data at a walking speed of 1.5m/s, the weights according to the Mahalanobis distance is able to be calculated. Calculated weights see Table 5-1. Then the obtained weights are able to be applied for estimating approximate scalar coefficients of walking data in speed of 2.5m/s. The appropriate walk cycle is constructed from the existing database by applying different number of principal components as shown in Figure 5.9.

Table 5-1 Obtained weights correspond to the existing data

$w(\theta_{v_{s1}}^x, \theta_{v_{s1}}^p)$	$\theta_{v_{s1}}^{p_1}$	$\theta_{v_{s1}}^{p_2}$	$\theta_{v_{s1}}^{p_3}$	$\theta_{v_{s1}}^{p_4}$	$\theta_{v_{s1}}^{p_6}$
$\theta_{v_{s1}}^x$	0.56	0.01	0.361	0.055	0.014

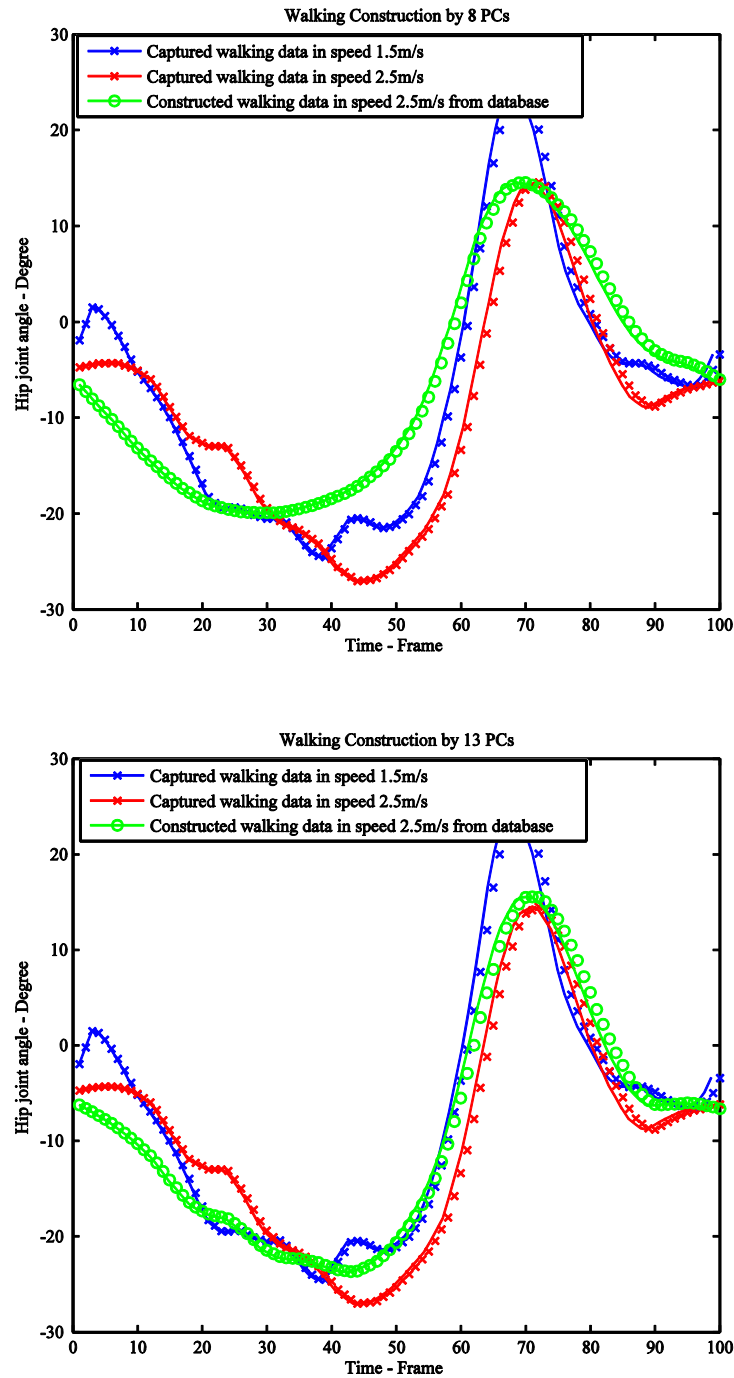


Figure 5.9 Joint angle constructions by applying different number of principal components: (Top) 8 PCs; (Bottom) 13 PCs, and joint angle trajectories comparisons between a given motion in low speed (blue), original captured motion in high speed (red) and constructed motion in high speed (green).

Finally, the original motion data has been modified and adapted to a motion sequence walking along a curve. Parts of these results have been implemented by using a Maya API that includes motion path trajectory definition, frame supplement, etc. Figure 5.10 shows a sequence of the still images for this incomplete motion data reconstruction.

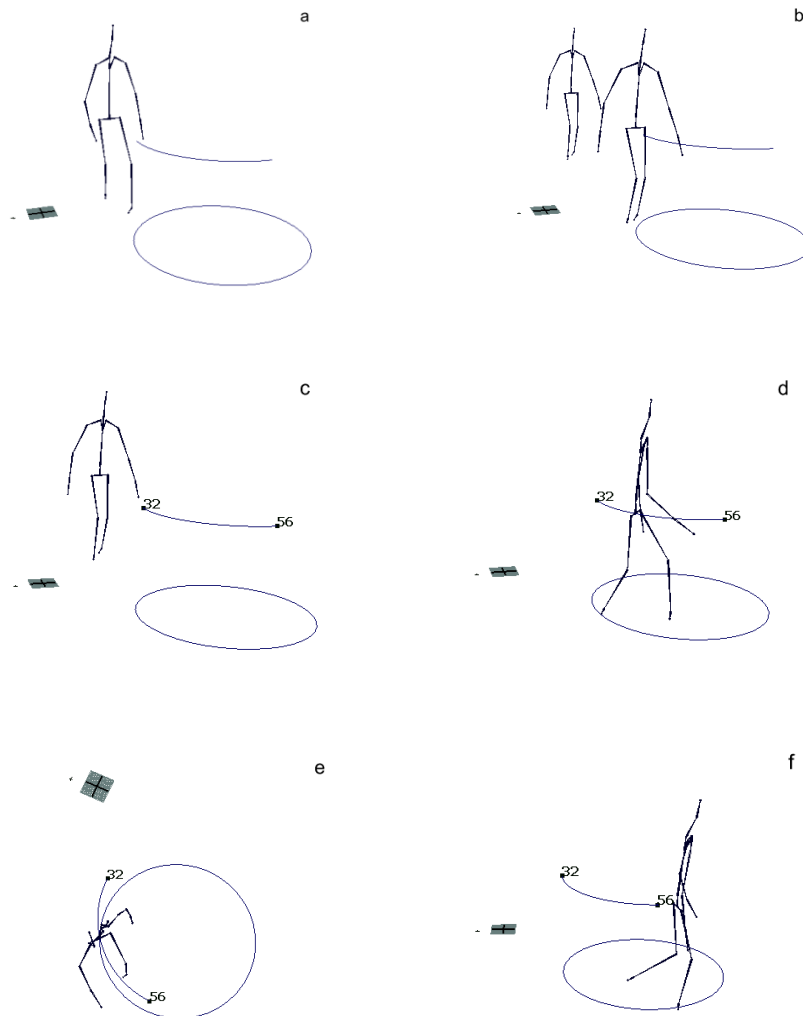


Figure 5.10 Image sequences of IMDR. (a) Is the original motion clip. (b) Search a proper motion for motion construction. (c) Define a proper path in spatial and temporal space. (d) Adjust and retarget input motion for motion reconstruction. (e) Top view of step (d). (f) End of motion data reconstruction.

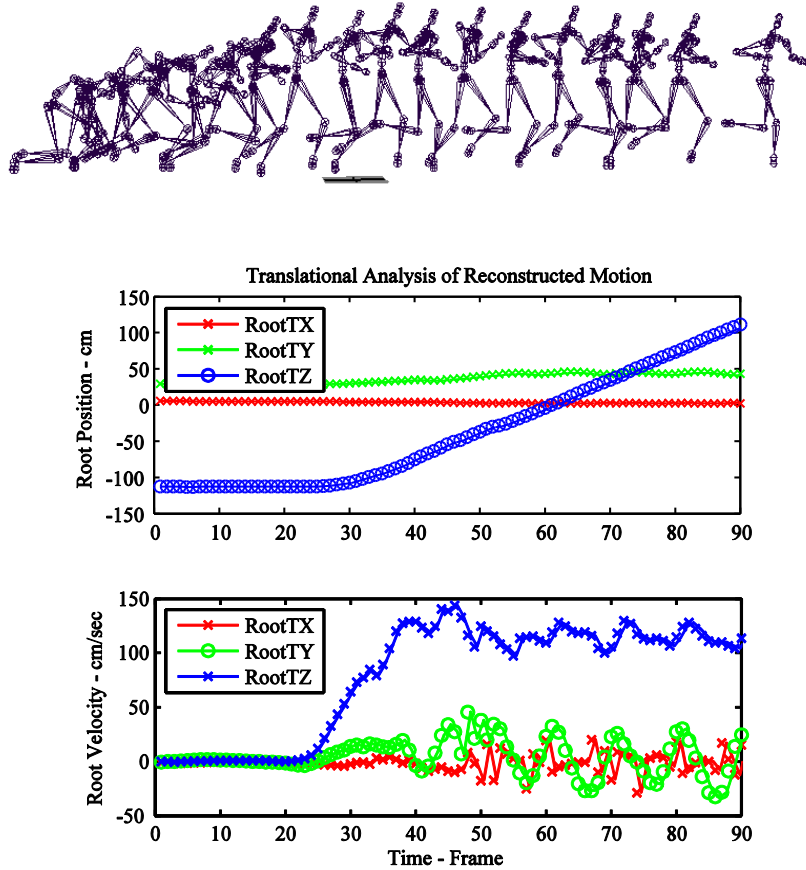


Figure 5.11 Sprint motion sequences constructed by employing IMDR method. (Top) Motion sequences of sprint from standby to run; (Bottom) Velocity analysis of Root joint.

5.4.2 Sprint

Due to the spatial limitation of capturing space, a whole sequence of sprint motion is unable to be captured. However, it is possible to generate such motion sequence by applying a motion blending technique. According to the previous analysis of blended motion shown in Figure 5.3, a sudden movement has been generated due to the velocity difference between standby and running motion clips. In order to eliminate such an artefact and construct a convincing motion sequence, an appropriate intermediate transition need to be constructed from an existing motion database for

generating such a movement. As the analysis results shown, the translation velocity has been built up to 1.6m/sec from standby motion clip and the running motion clip is 1.1m/s average. Rather than blending motion in a low speed see Figure 5.3, it would be better to find an intermediate transition motion, running in speed 1.4m/s, for generating the complete motion sequence. The final constructed sprint motion sequence by applying an intermediate motion clip is shown in Figure 5.11.

From the velocity comparison in Figure 5.12, it shows that the IMDR method is able to generate more convincing motion sequence for motion transition by employing motion analysis prior to motion blending.

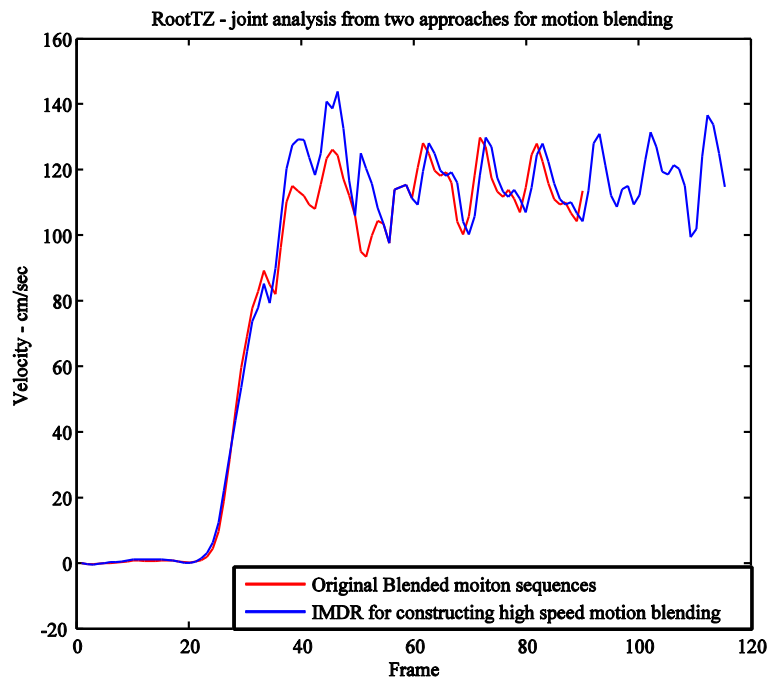


Figure 5.12 RootTZ joint comparison from original blended motion sequences and the constructed motion sequences by IMDR method.

5.5 Discussion

The presented incomplete motion data reconstruction method extends the use of existing motion data for animators to create wide range of motion sequences. The method is statistically based and only approximates the solution according to similarity and probability from the existing population.

Comparing the IMDR method with traditional motion blending techniques, the presented technique is to employ motion analysis and construct an appropriate intermediate motion sequence(s) for generating continuous movement. Considering the procedure for motion blending, the presented method is relatively time-consuming than direct motion blending from available motion sources. However, the presented method provides flexibility for animators to control the blending sequence by their desire through animation path design. And it provides animators with a way that enables them to guarantee the variation consistency of physical properties during a character's movement. Both IMDR and traditional methods have strengths and weaknesses for motion blending. The prospective of using IMDR and traditional technique for motion blending is listed in Table 5.2.

Table 5-2 Comparison of IMDR and traditional blending method

	IMDR	Traditional method
Motion database/ Motion sources	Both	Motion sources only
Motion analysis	yes	no
Method	Statistics-based	Direct interpolation
Flexibility	can pre-define path	No path define
Time-consuming	yes	no
Smoothness	yes	yes
Realistic	yes	Not always

From the results shown in Figure 5.9, the reconstructed motion data is close to the original captured data but has an obvious offset. There are a few factors that should be considered to correct the errors and improve the result, one of which is to build a large and comprehensive motion database that will enable animators to abstract even more precise motion patterns for motion synthesis. Another is to consider modifying the physical properties after the creation of transition motion.

5.6 Summary

When generating a motion transition from a limited range of motion sources, although the synthesized motion appears smooth in trajectories, the quality of movement might appear unrealistic and unconvincing as a result of the variation of physical properties, i.e. velocity etc. According to an animators' direction, based upon the technique that has been studied in Chapter 4, an intermediate movement is able to be generated from the existing motion database for the purpose of motion blending. This generated intermediate movement will be applied in the middle of those two motion clips as a transitional motion for motion blending. The presented IMDR method will provide animators with an additional and reliable technique for motion blending to create a continuous human character movement for computer animation production.

CHAPTER 6

MOTION DATA CORRECTION

Accurate and realistic motion is a vital feature of computer animation. The utilisation of motion capture data provides animators with a physically-based and plausible source of realistic motion. However, the main concern with motion capture involves motion data editing and synthesizing. When motion sequences are adjusted to accommodate a new scene or environment, it is very easy to introduce dynamic artefacts that make the motion appear unrealistic. Consequently, it is essential that any editing minimizes unwanted artefacts and maintains the integrity of the original motion sequence.

6.1 Introduction

MoCap systems provide animators with a valuable source of high-fidelity motion sequences, however, there are serious concerns associated with the process of motion data editing and synthesizing.

Even though great care is taken to capture a motion sequence according to an animator's script, MoCap data mapping and motion retargeting can introduce unwanted errors that can make the final animation appear unrealistic. e.g., the animated character body penetrates the ground or a foot slides along the floor. The reasons for these errors might be that the sensors

were located incorrectly on the actor's body, or it could be due to one or more sensors were not attached firmly to the actor's body. Apart from recapturing the motion to obtain more accurate motion, the correction of motion data is also an important procedure to produce convincing and realistic motion.

Motion synthesis is a technique used to generate a series of motion sequences which perform human actions at the appropriate time as realistically as possible. In general, the equation of motion for an articulated system can be described by the following differential equation (Craig 1989):

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G + F \quad (6.1)$$

where τ is the joint torque, θ is the joint degrees of freedom, $\dot{\theta}$ and $\ddot{\theta}$ are the joint angular velocity and angular acceleration respectively; $M(\theta)$ is an $n \times n$ inertia mass matrix, n is the number of degrees of freedom; $C(\theta, \dot{\theta})$ is a $n \times n$ matrix, including the centrifugal and coriolis effects; G is the gravitation force and F represents the external force.

Rearranging Equation (6.1), the joint angular acceleration can be obtained:

$$\ddot{\theta} = M^{-1}(\theta)[\tau - (C(\theta, \dot{\theta}) + G + F)] \quad (6.2)$$

Equation (6.2) defines a set of second-order ordinary differential equations:

$$\ddot{\theta} = f(\theta, \dot{\theta}, t) \quad (6.3)$$

where t stands for time and the boundary values for θ and $\dot{\theta}$ are defined. The motion synthesis is assumed to be the numerical solution of a boundary problem.

Due to the computing difficulty of the force function F , it is impossible to apply Equation (6.2) directly for generating an articulated human character movement. In practice, a motion synthesis technique aims to use 'reasonable' motions or an animator's sketch to find a set of satisfied internal states between a given initial state $(\theta_o, \dot{\theta}_o)$ and a desired final state $(\theta_f, \dot{\theta}_f)$. These satisfied internal states are represented through

constraints. Accordingly, the motion synthesis is presented as a constrained optimization problem.

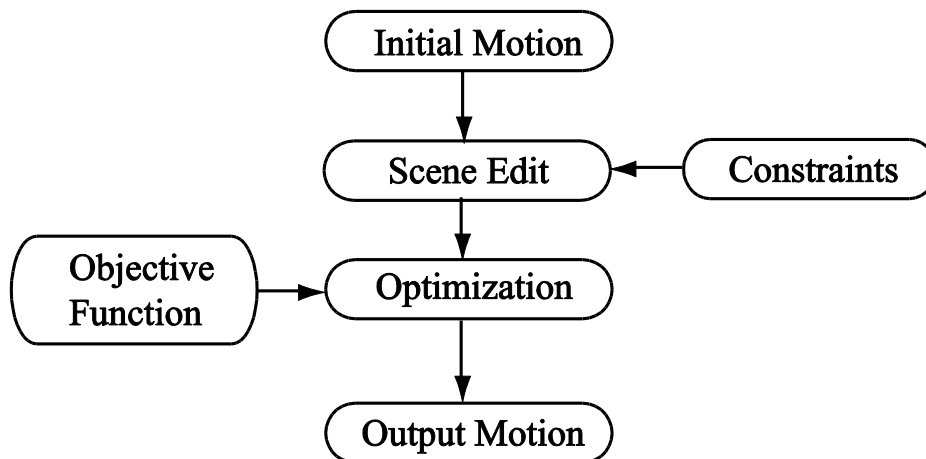


Figure 6.1 The structure of motion data correction by using constrained optimisation method.

Constrained optimization techniques have proven to be a powerful approach for generating new motions (Witkin and Kass 1988; Popović and Witkin 1999; Liu and Popović 2002; Fang and Pollard 2003). In this chapter, a physically-based optimization method is presented to correct and synthesize motions from existing motion data. The remainder of the chapter provides detailed information for applying such a method for motion data correction. According to the knowledge of biomechanics, the physical properties of a human character is defined and estimated. In considering the constraints employed for motion data correction, certain physical models are investigated for human character animation. The presented motion data correction technique is like a motion editing process where the procedure of the motion synthesis is from coarse to fine according to the objective function and the physical constraints which are appropriate for motion correction. By this means, unrealistic motion artefacts can be corrected and eliminated. The diagram of the motion data correction procedure for character animation is illustrated in Figure 6.1.

6.2 Human Character Model

In this experiment, the original motion data is captured by a Reactor2 system, an optical motion capture device. Using the knowledge of human anatomy, the human character was represented by a hierarchical skeleton, which has 16 rigid links (body parts) and 15 revolute joints (Figure 6.2). The total number of degrees of freedom (DOF) of the human character is 51, including 6 DOF (3 rotations and 3 translations) at the root joint (waist). The root is an unactuated joint which is defined using a reference coordinate system.

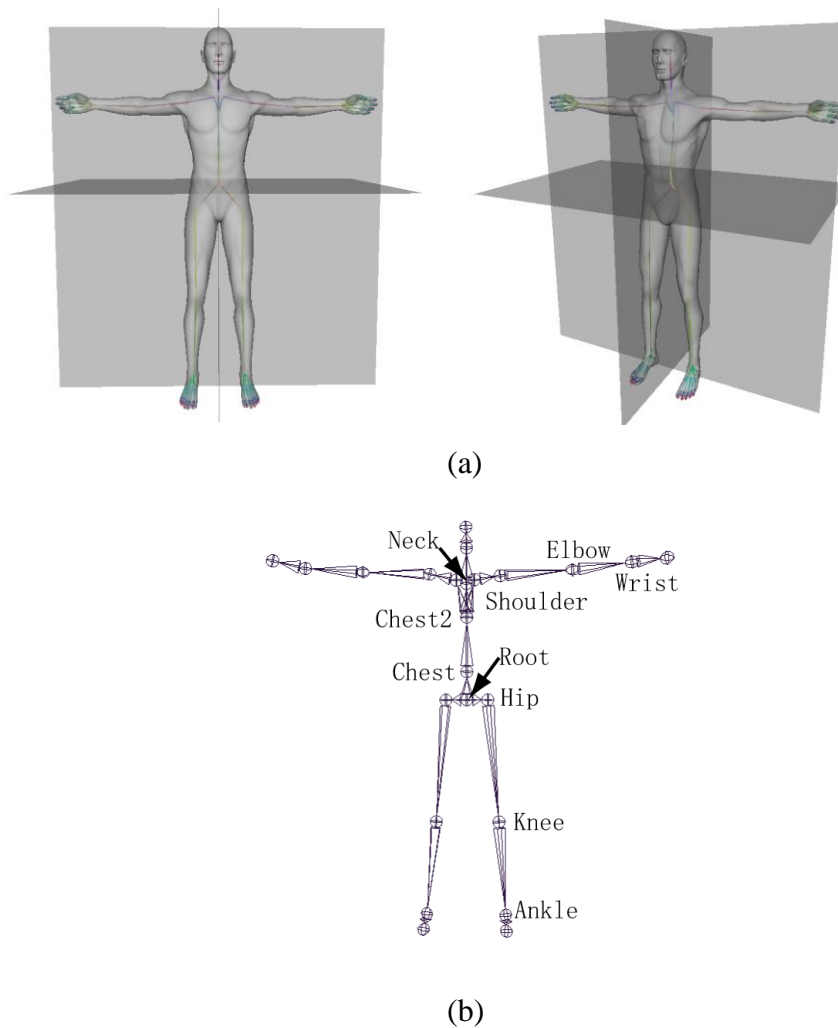


Figure 6.2 (a) CG Human character model with skeleton. (b) Skeleton of human character obtained after Mocap mapping.

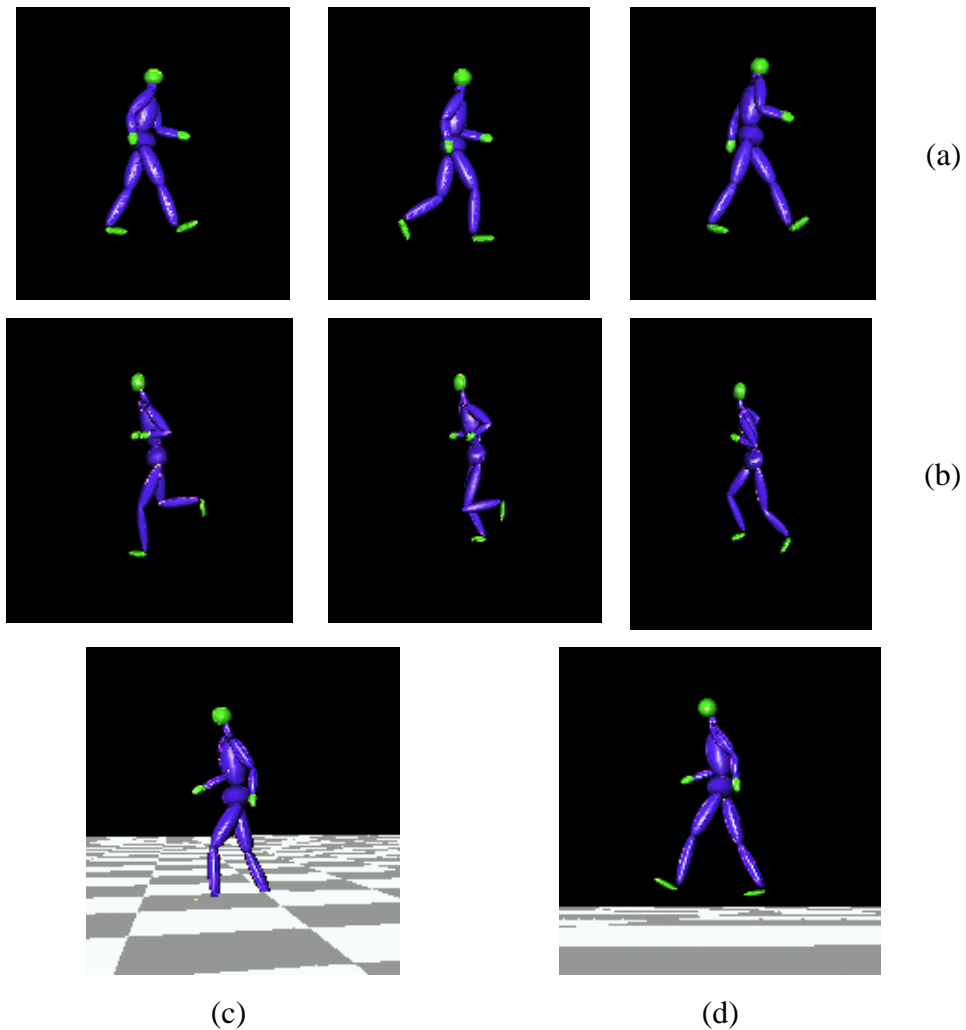


Figure 6.3 Human character animation based on motion capture data. (a) walking; (b) running; (c) feet penetrate ground; (d) airborne walking.

Motion data obtained from MoCap systems provide kinematic information for animators, which are in the form of discrete spatial points recording the position of the joint angles related to a hierarchical human character skeleton. As a result, most existing motion synthesis techniques mainly consider the smoothness of kinematic parameters, making the generated motions apparently natural-looking. However, if in the editing process important physical properties are neglected, the resulting motions can be unconvincing and unrealistic. By using C++ and the OpenGL Utility Toolkit (GLUT) library, a kinematics-based approach is implemented, i.e. manipulating the motion curve directly for motion synthesis and editing. In

the subfigure (a) and (b) of Figure 6.3, it illustrates the still images of human character walking and running respectively. By manipulating the motion curve directly, a certain unrealistic motions results have been generated are shown in Figure 6.3(c) and 6.3(d).

From the results demonstrated above, by using kinematics-based method, the motion curve can be manipulated directly for motion editing. However, the synthesized motion may appear unnatural. Therefore, it is necessary to consider employing different techniques, i.e., embed relevant physical properties, into motion capture data during the process of motion synthesis to help create physically correct motion. According to kinematic information of captured motion data, achieving physical properties becomes a rather difficult task since there are no dynamic properties included such as mass, moment of inertia and torque. However, by ways of estimating and pre-computing geometric shapes of the human body, some of the dynamic properties of the body part that relates to the corresponding skeleton are able to be obtained.

6.2.1 Physical properties of human character

The physical properties of animated characters are dependent on the mass distribution of the human character – an actor. However, there is no explicit information about mass and moment of inertia of individual body parts from the captured motion data. The only known parameters are the hierarchical structure of the animated human character and the total mass of the real performer. Therefore, a method has to be presented to fit the mass distribution to the character, in order that the physical laws are satisfied.

Shin et al. (2003) presented an optimization method to determine the mass distribution using physically plausible motion, constrained by the zero moment point method, while most researchers estimate the mass distribution of human characters from biomechanics literature in considering the reliability of biomechanical experiments (Hodgins et al.

1995; Tak et al. 2000). In this experiment, based on studies undertaken by Winter (1990), the primitive geometric model of human body parts is able to be estimated and the dynamic properties of the human character is able to be computed separately with empirical data.

There is an empirical function which can be used to calculate and obtain the estimated parameters for the mass distribution of body parts. Drill and Contini (1966 cited Winter 1990) developed an expression for body density ρ as a function:

$$\rho = 0.69 + 0.9c \quad (6.4)$$

where $c = H/W^{1/3}$, H is body height (meter) and W is body weight (kilogram).

Therefore, the body part of character can be achieved from the empirical function

$$m_i = w_i M \quad (6.5)$$

where m_i is the mass of the i^{th} body part; w_i is the proportional coefficient and M is the total mass of a character. Further details about the empirical equation for calculating the mass of body parts and how to choose the proportional coefficient can be found in Winter (1990 chap. 3).

For a single DOF system, physical property is often referred to the mass of a rigid body. However, in a rotational motion system, the notion of the moment of inertia is another important property of a rigid body. In order to present the mass distribution of a rigid body in a complete way, a set of quantities which preserve information about the mass distribution of a rigid body relative to a reference rotation axis should be defined.

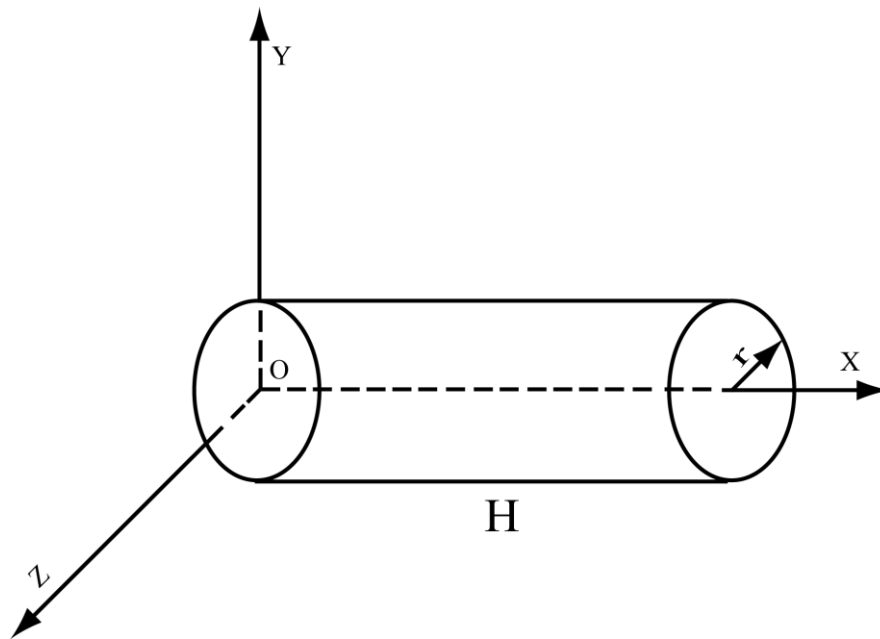


Figure 6.4 Cylinder with uniform density in a reference frame.

An example of a cylinder with an attached frame is shown in Figure 6.4. The inertia tensor relative to the reference frame is expressed in a matrix form as:

$$I_o = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (6.6)$$

where I_o is the moment of inertia of a rigid body in reference frame O . For a rigid body with uniform density, the scalar elements are given by

$$I_{xx} = \iiint (y^2 + z^2) \rho dx dy dz$$

$$I_{yy} = \iiint (x^2 + z^2) \rho dx dy dz$$

$$I_{zz} = \iiint (y^2 + x^2) \rho dx dy dz$$

$$I_{xy} = \iiint xy \rho dx dy dz$$

$$I_{yz} = \iiint yz \rho dx dy dz$$

$$I_{xz} = \iiint xz \rho dx dy dz$$

The moment of inertia of a rigid body is dependent on the geometrical shape and rotation axis. For different geometric objects, the reader is referred to Craig (1989). Using the same approach, Popović (1999) in his thesis calculated these physical properties according to the geometry of an animated character by using the following equations:

$$m_i = \iiint_V \rho_i dx dy dz \quad (6.7)$$

$$c_i = \frac{1}{m_i} \iiint_V P \rho_i dx dy dz \quad (6.8)$$

$$I_i = \int r^2 m = \int \rho_i r^2 dV = \iiint_V \rho_i r^2 dx dy dz \quad (6.9)$$

where m_i , c_i and I_i are the mass, centre of mass and moment of inertia of i^{th} body part respectively, P is the position of the mass particle, ρ_i is the density of the primitive, r is perpendicular distance from the axis of rotation.

In order to compute the moment of inertia, the geometric shape of individual body parts has to be estimated in advance. The moment of inertia is calculated in the centre of mass with respect to the X, Y and Z rotation axes respectively. As shown in Table 6-1, the main physical properties of each body part of a male performer are pre-computed prior to the motion synthesis process.

Table 6-1 Physical properties of human character from a male performer

Name (Body Parts)	Mass (kg)	Length (m)	Primitive	Moment of inertia ($\times 10^{-3} \text{ kg} \cdot \text{m}^2$)
Hip	10.65	0.335	Rectangular	(6.03,8.353,6.03)
Chest	10.425	0.18	Rectangular	(7.838,8.1766,7.838)
Chest2	17.625	0.338	Rectangular	(9.321,13.8,9.321)
Thigh	7.5	0.441	Cylinder	(130.134,130.134,4.412)
Lower leg	3.487	0.42	Cylinder	(56.935,56.935,1.5173)
Foot	1.087	0.26	Rectangular	(3.153,3.153,0.631)
Upper arm	2.1	0.335	Cylinder	(15.651,15.651,0.653)
Lower arm	1.2	0.263	Cylinder	(5.853,5.853,0.135)
Hand	0.45	0.18	Rectangular	(0.216,0.216,0.407)
Neck & Head	6.075	0.32	Cylinder	(2.008,2.008,0.878)

6.2.2 Joint limitation of human character

In order to achieve a realistic character movement, joint limitations of 3D human characters are the most common constraints that are defined in character animation. For example, the hip joint has 3 DOFs, the constraints for a hip joint include (detailed joint limitation refer to Table 6-2):

$$\begin{aligned}
 \theta_{\min x} < \theta_{hipx} < \theta_{\max x} \\
 \theta_{\min y} < \theta_{hipy} < \theta_{\max y} \\
 \theta_{\min z} < \theta_{hipz} < \theta_{\max z}
 \end{aligned} \tag{6.10}$$

where θ_{hipx} is the hip joint angle rotate around x axis, $\theta_{\min x}$ and $\theta_{\max x}$ are the minimum and maximum joint rotation angles around x axis respectively.

Table 6-2 Configuration of Joints¹

Joint Name	Degree Of Freedom	Limitation (degree)
Root	$R(x, y, z) T(x, y, z)$	NoN
Hip joint	R_x	(-120, 30)
	R_y	(-25, 25)
	R_z	(75, -35)
Knee joint	R_x	(0, 105)
Ankle joint	R_x	(-20, 40)*
	R_y	(-5, 5)*
Waist joint	R_x	(-10, 95)
	R_y	(-35, 35)
	R_z	(-60, 60)
Chest joint	R_x	(15, 0)*
	R_y	(-20, 20)*
	R_z	(-5, 5)*
Shoulder joint	R_x	(-175, 60)
	R_y	(-5, 5)
	R_z	(-30, 120)
Elbow joint	R_x	(-135, 0)
	R_y	(0, 20)
Wrist joint	R_x	(-40, 40)
	R_z	(-5, 15)
Neck	R_x	(-10, 30)
	R_y	(-15, 15)
	R_z	(-20, 20)

¹ Parts of data derived from the sports centre of Bournemouth University, some of data estimated from the personal test (marked as *).

6.3 Physics-based human character

In computer graphics a human character is formed from a hierarchical chain through revolute joints. Before the demonstration of the physically-based motion synthesis for human character animation, it is better to understand the construction of dynamic formulation of a human character. To consolidate the knowledge of physics – Newton-Euler equations (6.11), a rigid body is always considered first. These equations describe how forces, moments of inertia, and accelerations relate to the movement of a single rigid body:

$$\begin{aligned} F &= ma_c = m\dot{v}_c \\ N &= I_c\dot{\omega} + \omega \times I_c\omega \end{aligned} \quad (6.11)$$

where F is a resultant force acting at the centre of mass of rigid body, m is the total mass of the rigid body, a_c and \dot{v}_c are accelerations of the centre of mass, N is a resultant moment acting on the body, I_c is the moment of inertia of the rigid body, ω and $\dot{\omega}$ are rigid body angular velocity and angular acceleration respectively.

Bones, muscles, ligaments and other organs are all parts of the human body, and human movement is difficult to be described as a single mathematical model. However, given the consequences of human movement, it is the muscle force acting on bones and presented as joint torques that drive motions. The subject of an articulated character movement has been studied for many years in biomechanics and robotics (Featherstone 1987; Craig 1989; Winter 1990; Isbweb 2003).

Assuming the joint position θ , velocity $\dot{\theta}$, and acceleration $\ddot{\theta}$ in Equation (6.1) are known, with the mass distribution information of the human character, the joint torques required to cause this motion are able to be calculated. This technique is known as ***Inverse Dynamics***. Compared with two popular dynamic formulations (force-balance approached Newton-Euler method and energy-based Lagrangian method), the computation cost

of Newton-Euler is $O(n)$ whereas the Lagrangian dynamic formulation costs $O(n^4)$ (where n is the total joints' DOFs) (Featherstone 1987; Craig 1989). More details of recursive Newton-Euler dynamic formulation for series link can be found at Appendix A.

6.3.1 Dynamic balance constraint

In character animation, balance is always regarded as one of the most significant properties. Aydin and Nakajima (1999) control a balanced character by manipulating a certain joint torques. Some researchers consider controlling the centre of mass of articulated character within a supporting area (Boulic et al. 1995; Popović 1999). The constrained criterion of controlling centre of mass within a supporting area is regarded as static balance. In order to accurately describe the balance of a moving human character, a proven criteria of dynamic balance has been studied, i.e. the *ZMP* within the supporting polygon (Tak et al. 2000; Komura et al. 2000; Shin et al. 2003). When a walking or running character is in contact with the external environment, e.g. ground, each body part will contribute to the moment at the supporting polygon (foot) by $m_i(r_i - Z) \times \ddot{r}_i$. From the theory of equilibrium of moments, there exists a point where the net torque contributed from each body part, due to ground forces, is equal to zero. In other words, the sum of the torques from each body part equals the torque due to gravity as Equation (6.12):

$$\sum_i m_i(r_i - Z) \times \ddot{r}_i = \sum_i m_i(r_i - Z) \times g \quad (6.12)$$

As illustrated in Figure 6.5, following equation can be solved to obtain the *ZMP*:

$$\sum_i m_i(r_i - Z) \times (\ddot{r}_i - g) = 0 \quad (6.13)$$

where m_i and r_i are the mass and the position of the mass centre of the i^{th} link respectively, \ddot{r}_i is the acceleration of the mass centre of i^{th} link, Z represents the position of *ZMP* in a referenced coordinate system.

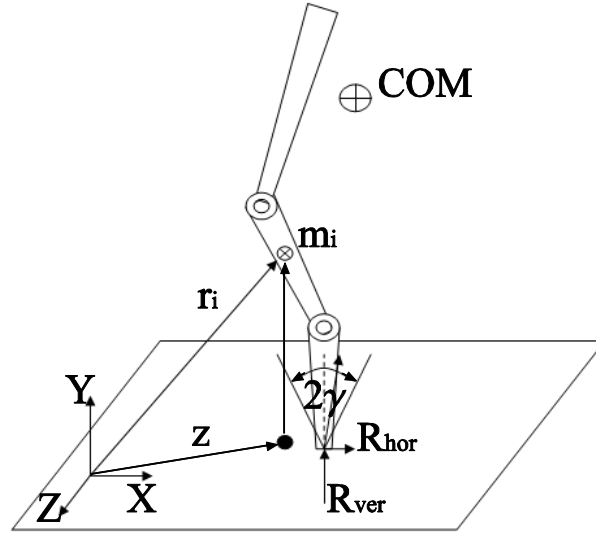


Figure 6.5 Efficient dynamic constraints of human character's motion (Dynamic balance constraint: ZMP calculation and Coulomb's Contact Model).

There are an infinite number of solutions for Equation (6.13), and also, these solutions will change under the different terrain in the virtual environment. A unique and explicit equation of the *ZMP* from Equation (6.14) therefore can be given by fixing $ZMP_y = const$:

$$ZMP_x = \frac{\sum_i m_i r_{i_x} (\ddot{r}_{i_y} - g_y) - \sum_i m_i r_{i_y} (\ddot{r}_{i_x} - g_x)}{\sum_i m_i (\ddot{r}_{i_y} - g_y)}$$

$$ZMP_y = const \quad (6.14)$$

$$ZMP_z = \frac{\sum_i m_i r_{i_z} (\ddot{r}_{i_y} - g_y) - \sum_i m_i r_{i_y} (\ddot{r}_{i_z} - g_z)}{\sum_i m_i (\ddot{r}_{i_y} - g_y)}$$

The resulting motion is treated as a lack of balance when the *ZMP* is outside the support polygon. In theory, by correcting the *ZMP*, a character's posture could be adjusted to gain a new balance. However, it is difficult in practice to choose a proper way to adjust *ZMP*, since an articulated character has a large number of DOFs. In the presented method, in order to keep the synthesized motion as similar to the original input motion data as

possible, the chest joint is adjusted to keep the *ZMP* inside the supporting polygon.

6.3.2 Ground contact force

In computer animation an animated human character is supposed to interact with different parts of the virtual environment, i.e. ground, objects etc. During motion synthesis, the animator explicitly defines some positional constraints of the character. In order to control the position of a supporting leg, a position detector is used to define the position of the foot. Coulomb's contact model (Abe et al. 2007; Fang and Pollard 2003; Vukobratovic et al. 1990) is used for contact between the character and the virtual environment. The total reaction force \vec{R} is within the cone of angle 2γ , and its horizontal component opposing the sliding will be of sufficient intensity to prevent an unwanted horizontal motion for the supporting foot over the ground surface. Coulomb's contact model is shown in Figure 6.5.

$$\frac{\vec{R}_{hor}}{\vec{R}_{ver}} \leq \tan \gamma = \mu \quad (6.15)$$

Where \vec{R}_{hor} and \vec{R}_{ver} are horizontal and vertical components of reaction force respectively, μ is the friction coefficient of the ground. The inequality constraint for the contact friction force is therefore:

$$\sin^{-1}\left(\left|\frac{\vec{R}_{hor}}{\vec{R}}\right|\right) < \tan^{-1} \mu \quad (6.16)$$

Equivalently, the constraint without the inverse trigonometric function is:

$$\frac{|\vec{R}_{hor}|}{|\vec{R}|} < \frac{1}{\sqrt{(1+\mu)^2}} \quad (6.17)$$

The magnitude of the horizontal friction force can be constrained as follows:

$$0 \leq |\vec{R}_{hor}| \leq \frac{|\vec{R}|}{\sqrt{(1+\mu)^2}} \quad (6.18)$$

6.4 Constrained optimization method

The constrained optimisation problem is to minimize the objective function subject to various constraints at each stage as follows:

$$\min f(\theta(t)) \quad \text{subject to} \quad c_i(t) = 0, i = 1 \dots n, t \in [t_0, t_f] \quad (6.19)$$

where $f(\theta(t))$ is objective function, $\theta(t)$ is joint angle in time domain, and $c_i(t)$ are the constraints, and t represents the individual time from starting time t_0 to final time t_f .

To solve such a constrained optimisation problem for character animation, a sequential quadratic programming (SQP) has been presented and proved to be a reliable method (Abe et al. 2004; Cohen 1992; Fang and Pollard 2003; Gleicher 1997; Popović 1999; Safonova et al. 2004; Witkin and Kass 1988). The interested reader can also refer to (Barclay et al. 1997; Heath 1997; Nocedal and Wright 1999).

6.4.1 Cubic B-spline

In computer graphics, human character movement is described by a set of joint variations over time. The unknown $\theta(t)$ in Equation (6.19) is a set of trajectory functions. In order to solve such an optimization problem, it's better to represent the minimization in terms of unknowns. To fit the trajectories of joint angle over time, a cubic spline is a promising method that guarantees the curve has C^2 parametric continuity. A spline employs piecewise polynomials that approximate functions numerically. The interested reader can refer to the book (Rogers and Adams 1990).

In this experiment, cubic B-spline is the method for fitting a curve. In general, a B-spline curve is defined by

$$\theta(t) = \sum_{i=0}^m B_{i,k}(t) x_i \quad t_{\min} \leq t < t_{\max} \quad 2 \leq k \leq m \quad (6.20)$$

where $B_{i,k}(t)$ is a set of normalized B-Spline basis functions, k is the order (degree = $k - 1$) of the basis function, x_i is the i^{th} control point of the $m + 1$ defining polygon vertices – the coefficient of basis function, t is called knots.

The derivatives of a B-spline curve at any point on the curve are obtained by formal differentiation. The first and second derivatives of cubic B-spline function are as follows:

$$\dot{\theta}(t) = \sum_{i=0}^m \dot{B}_{i,k}(t)x_i \quad (6.21)$$

And

$$\ddot{\theta}(t) = \sum_{i=0}^m \ddot{B}_{i,k}(t)x_i \quad (6.22)$$

Substitute Equation (6.20) into Equation (6.19), the constrained optimization problem is adapted as follows:

$$\min_x f(B(t)x) \quad \text{subject to} \quad c_i(t) = 0, i = 1 \dots n, t \in [t_0, t_f] \quad (6.23)$$

According to the trajectory of character movement, given a function $f(\theta, \dot{\theta}, \ddot{\theta})$, it is able to

$$\begin{aligned} \frac{\partial f}{\partial x} &= \frac{\partial f}{\partial \theta} \frac{\partial \theta}{\partial x} + \frac{\partial f}{\partial \dot{\theta}} \frac{\partial \dot{\theta}}{\partial x} + \frac{\partial f}{\partial \ddot{\theta}} \frac{\partial \ddot{\theta}}{\partial x} \\ &= \frac{\partial f}{\partial \theta} B + \frac{\partial f}{\partial \dot{\theta}} \dot{B} + \frac{\partial f}{\partial \ddot{\theta}} \ddot{B} \end{aligned} \quad (6.24)$$

Since the motion data correction or motion synthesis only considers the trajectory in very short period of time, motion of 1 or 2 second or even less. Figure 6.6 is an example of 3 second long knee joint trajectory fitting by a cubic B-spline according to different number of control points. If more control points are applied, the motion will be smoother. However, if too many control points are applied, the computational cost will increase

according to the number of unknowns. In the experiment, for a short movement, a very smooth joint trajectory is able to be represented by $\simeq 8$ control points.

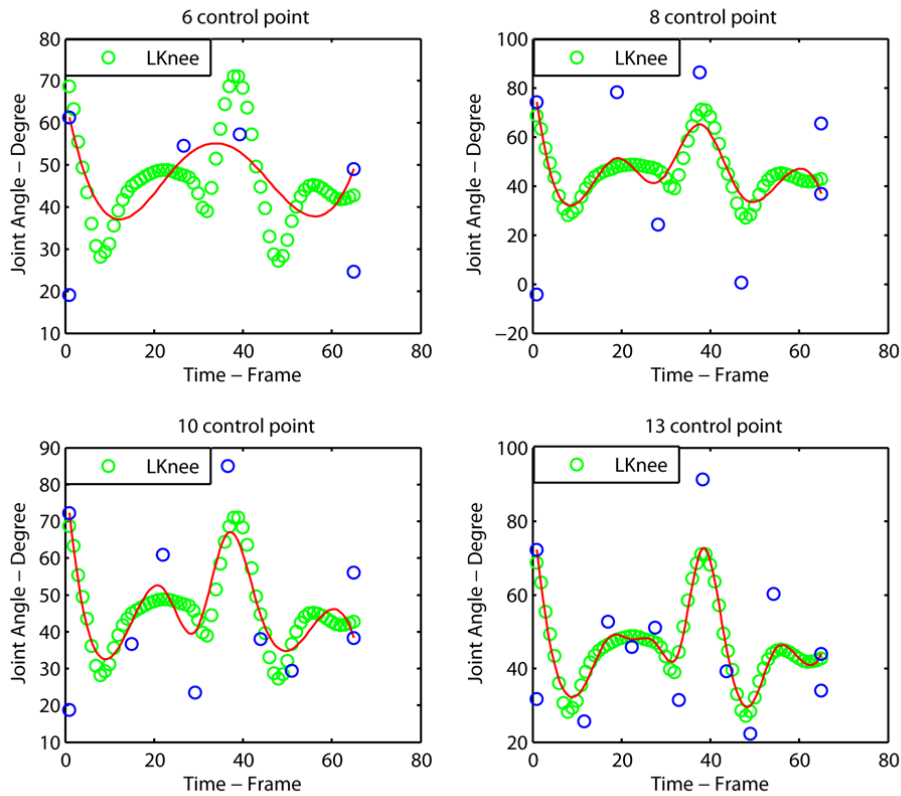


Figure 6.6 Comparison of curve fitting by applying different number of control points of Cubic B-spline.

6.4.2 Objective function

One objective function that considers displacement mapping, which is to minimize the offset from the reference motion $\theta_r(t)$ (Gleicher 1998; Tak et al. 2000):

$$f(\theta_i(t)) = \int_{t_0}^{t_f} (\theta(t) - \theta_r(t))^2 dt \quad (6.25)$$

As a result, the synthesized motion is kept as similar to the referenced motion as possible. However, this works successfully only if no errors appear in the original data, which in many cases is not true.

Thus, physical properties are taken into account, and the energy consumption is chosen as the objective function, which is the integral of the sum of squared joint torques (Popović and Witkin 1999):

$$f(\theta_i(t)) = \int_{t_0}^{t_f} \sum_{i=1}^n \tau_i^2(\theta, t) dt \quad (6.26)$$

6.5 Experiment results

In the experiment, a walking example of an animated human character is demonstrated. The original motion includes 28 frames and this motion is less than 2 seconds. The editing procedure of motion synthesis is from coarse to fine. The time complexity for each iteration is $O((mn)^2)$, m is the number of degree of freedom, n is the number of B-spline coefficients for each degree of freedom. Due to the fact, the original motion data is employed as a starting initial condition and the data correction takes place if artefact appears, it converges less than 0.5 second within a small number of iterations. The demonstration of motion data correction is implemented by applying Maya API in a PC with Intel(R) PentiumIII 794MHz and 512MB memory. The structure of node connection of the experiment from Maya API is shown in Figure 6.7.

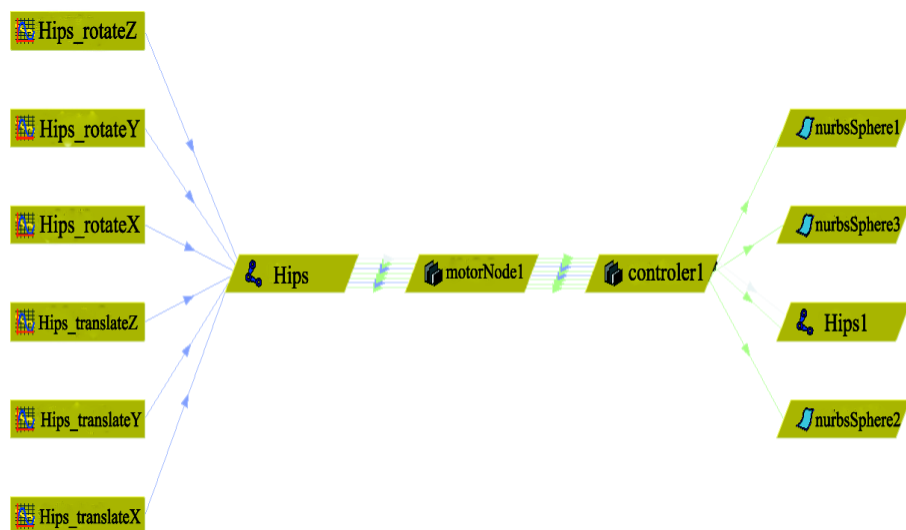


Figure 6.7 Example of Maya API node structure

According to the obtained pre-computed results from Section 6.2, following the allocation of mass and moment of inertia of each body part to corresponding link within the articulated character skeleton, and then the degrees of freedom of each joint have been set for joint limitation constraints. From the mass distribution of the articulated character, mass centre of whole body, upper body and lower body have been computed during the character movement. Such information is used to monitor the movement trajectories of the character.

In order to fulfil the motion correction process by physics-based optimization method, a series of nodes have been designed for the implementation. According to the node structure of Maya API, the trajectories of each joint are known. The joint trajectories provide the necessary information for carrying out the iterative Newton-Euler dynamic formulation. As shown in Figure 6.7, a motorNode is designed for each joint that constructs a serial link from root joint to leave joint. It provides the main computation of each joint and transmits the output to the next joint. To apply ground contact force to the character, a position detector has been integrated into the end joint of each leg that provides the area information of supporting polygon and detects the time when foot joint touch down the floor. In this experiment, the friction coefficient of the ground μ is equal to 0.9. Further trajectory information of ZMP has also been computed in controller node during the run-time for monitoring the dynamic balance during the character walking.

6.5.1 Foot slipping correction

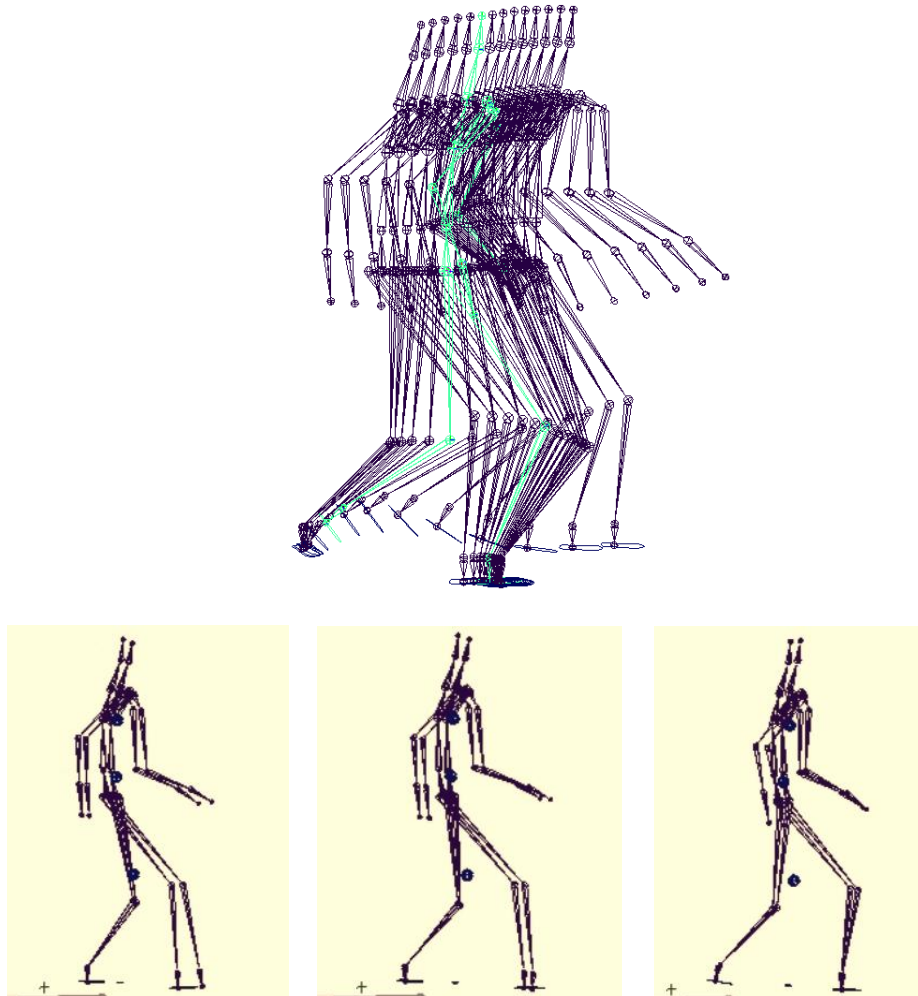


Figure 6.8 Eliminate foot slip during human character walking motion. (Top) Original walking motion, right foot slip during touch-down period; (Bottom) Comparison of original motion (skeleton with foot print) and corrected motion (skeleton without foot print).

It has been found that the original motion is unrealistic due to the right foot slipping during its supporting period (see top image of Figure 6.8). This is possibly due to the inaccuracy of the position of the motion capture marker, which was located too far away from the toe joint. The movement of the marker corresponds with the period when the heel of the foot touches the ground to the point when the toe leaves the ground. The position detector

located at the end joint was used to detect the position of the foot. In order to resolve the slip effects, the final position of the toe joint as the point when it leaves the ground has been defined and constrained. Figure 6.8 shows a walking motion including the original motion capture data and the synthesized motion.

6.5.2 Walking upstairs

In Figure 6.9, it shows the synthesized motion of a character walking upstairs. The original input motion data was similar to that in Figure 6.8. According to the virtual environment, a proper set of states for a character walking upstairs have been defined. The final trajectory of the root was obtained by mapping the height of the stairs with the original motion data. The trace of the root is defined as a parabolic function. In order to maintain dynamic balance for upstairs walking, the upper body tilted forward to keep the ZMP in the foot support area.

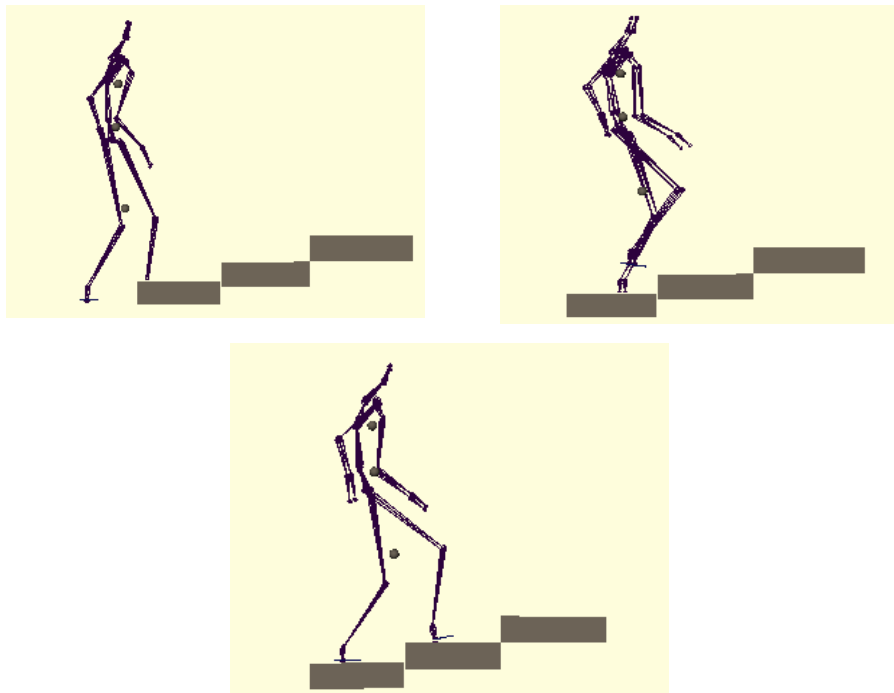


Figure 6.9 Comparison of human character walking upstairs from original motion and corrected motion.

6.6 Discussion

Time-consuming animation techniques concern animators, especially physically-based techniques. In this research, instead of generating a new motion by setting different constraints during spatial-temporal domain like other research that has been discussed earlier, a physically-based optimization method for motion correction has been applied whenever the MoCap data appears unrealistic. Among a wider range of artefacts in character animation, i.e. body part penetrating, reaching and sliding etc, in the experiment of the motion correction procedure, the artefact of foot sliding in skeleton mode of human character animation only be considered.

In the presented optimization approach for motion data correction, the muscle force imposed on the joint is considered. Therefore, the only objective function is to consider the energy consumption, which is to minimize the sum of the square of the joint torques. The choice of the objective function is the essence of producing a realistic and convincing motion sequence. Despite the corrected motion that has been obtained, there are certain limitations to this technique which suggest areas of future research. First of all, more sub-objective functions should be added into the objective function, and the functions should be derived from different knowledge. It should obey not only physical laws, but also other frameworks, e.g. similarity. Furthermore, more convincing tools should be provided for animators to synthesize motion.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Generating realistic and convincing motion for feature character animation has become a major area of interest for the computer graphics industry and research community. Motion capture technology provides a powerful tool to record detailed movements of performers, which can be replicated to drive computer-generated characters. Extending the power of motion capture technology will assist animators in creating realistic character animation.

In this thesis, various techniques have been investigated for creating articulated character animation. A number of novel algorithms have been presented for generating realistic human movements by employing captured motion data. In this final chapter, the study of post-processing techniques of motion capture data are concluded for human-like character animation and a number of challenging areas are suggested for extending the research in the future.

7.1 Conclusion

To obtain controllable and editable motion data for character animation, the mapping of motion capture data to a hierarchical skeleton is a very important procedure to achieve realistic and physical available motion for

-
- [Glardon et al. 2004] Glardon, P., Boulic, R. and Thalmann, D., 2004. PCA-Based Walking Engine Using Motion Capture Data. *CGI '04: Proceedings of the Computer Graphics International*, pp 292-298. ISBN: 0-7695-217-1, ISSN: 1530-1052, DOI: 10.1109/CGI.2004.73.
- [Gleicher 1997] Gleicher, M., 1997. Motion Editing with Spacetime Constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pp 139-148. ISBN: 0-89791-884-3, DOI: 10.1145/253284.253321.
- [Gleicher 1998] Gleicher, M., 1998. Retargetting Motion to New Characters. *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp 33-42. ISBN: 0-89791-999-8, DOI: 10.1145/280814.280820.
- [Gleicher 2001] Gleicher, M., 2001. Motion Path Editing. *I3D '01: Proceedings of the 2001 Symposium on Interactive 3D Graphics*, pp 195-202. ISBN: 1-58113-292-1, DOI: 10.1145/364338.364400.
- [Heath 1997] Heath, M.T., 1997. *Scientific Computing: An Introduction Survey*. International editions, McGraw-Hill, ISBN: 0-07-027684-6.
- [Hodgins et al. 1995] Hodgins, J.K., Wooten, W.L., Brogan, D.C. and O'Brien, J.F., 1995. Animating Human Athletics. *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp 71-78. ISBN: 0-89791-701-4, DOI: 10.1145/218380.218414.
- [Hodgins and Pollard 1997] Hodgins, J.K. and Pollard, N.S., 1997. Adapting Simulated Behaviors for New Characters. *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp 153-162. ISBN: 0-89791-896-7, DOI: 10.1145/258734.258822.
- [Honda 2003] Honda Motor Co. Ltd., 2003.
Available from: <http://www.honda.co.jp/english/technology/robot/>.
[Accessed March 2003]
- [Hsu et al. 2007] Hsu, E., da Silva, M. and Popović, J., 2007. Guided Time Warping for Motion Editing. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 42-52. ISBN: 978-1-59593-624-4.

ability of an animator for motion blending. On the other hand, it also extends the availability of re-using existing MoCap data efficiently.

Although motion capture is able to record accurate and detailed motion from a performer in three-dimensional space, the movement of a computer generated character in joint space might contain certain artefacts after post-processing, e.g. foot slip etc. The motion data correction method in Chapter 6 employs the knowledge from biomechanics and physics. After presenting a detailed procedure for constructing physical properties of human character from the existing information of motion capture data, a series of physical constraints have been applied on the model to generate realistic movement for character animation in a computer-generated environment. This technique extends the editing ability for an animator to generate realistic and convincing animation.

7.2 Future Work

Based on the current research results, there are some new areas of research for the future:

7.2.1 Automatic motion data segmentation and searching

So far the motion database, which includes walking and running motion data, is constructed manually according to the current knowledge of human motion. It is rather time-consuming to obtain such a large range of data. To build a comprehensive motion database, it needs not only to capture a large amount of motion data, but also to present a method that enables motion data to be automatically segmented from the captured motion data sequences. The segmented motion data can be built into the database based upon a certain criteria, which will be utilised for searching appropriate motion data in the later procedure of motion synthesis.

7.2.2 Motion data acquisition

Considering the data acquisition approach described in Chapter 4 for motion analysis, it is necessary to address an alternative means for data collection. The motion that a person performs everyday is like an “active” movement, which means people control their behaviours directly and naturally. However, when a person performs on a treadmill, the performance is regarded as a “passive” movement. Such a “passive” movement might not appear natural and might change the quality of motion data. To address this issue, it is important to employ an appropriate method for massive amount of motion data acquisition.

7.2.3 Physically based motion reconstruction

The method of incomplete motion data reconstruction is to generate intermediate transitions between different motion clips. This statistically-based method requires a comprehensive motion database to support motion synthesis and generate similar movements from abstracted motion patterns. To employ certain physical methods, the presented incomplete motion data reconstruction method will achieve more detailed movements from very basic motion patterns. The physically-based motion reconstruction will extend the ability of generating convincing and realistic transitions for motion synthesis from different motion clips.

7.2.4 Motion blending in marker space

Further investigation of the motion blending techniques for creating continuous movement from multi motion clips has been undertaken. Due to the fact that marker trajectories describe a detailed movement from a performer, instead of blending motion for character animation in joint space after post-processing procedure of captured data, blending and generating a new motion in marker space from captured raw motion data is an alternative approach. It means that animators will be able to achieve a long, continuous and blended motion for character animation directly after the process of MoCap data mapping.

7.2.5 Dynamic reaction with virtual environment

The aim of motion synthesis is to create a realistic motion sequence during the off-line editing procedure. For the purpose of the on-line procedure, e.g. computer game, there are unexpected interferences with computer generated character from virtual environment. To achieve realistic reaction motion from external impact, the mathematical-physical human model that has been studied is able to be applied to mimic the human behaviours in the virtual environment.

Appendix A

Iterative (or recursive) Newton-Euler Dynamic Formulation

This iterative Newton-Euler has two steps to fulfil dynamic formulation as shown in Figure A. The outward step is to compute inertial forces from root link to leaf link according to the rotational velocity and linear and rotational acceleration of the centre of mass of each link of the hierarchical chain. The inward step is to calculate the joint torques that will result in these net forces and torques being applied to each link. The standard notations from physics and robotics have been applied for formulating the equations and the algorithm presented is based upon Craig (1989), the interested reader also can refer to book (Featherstone 1987)

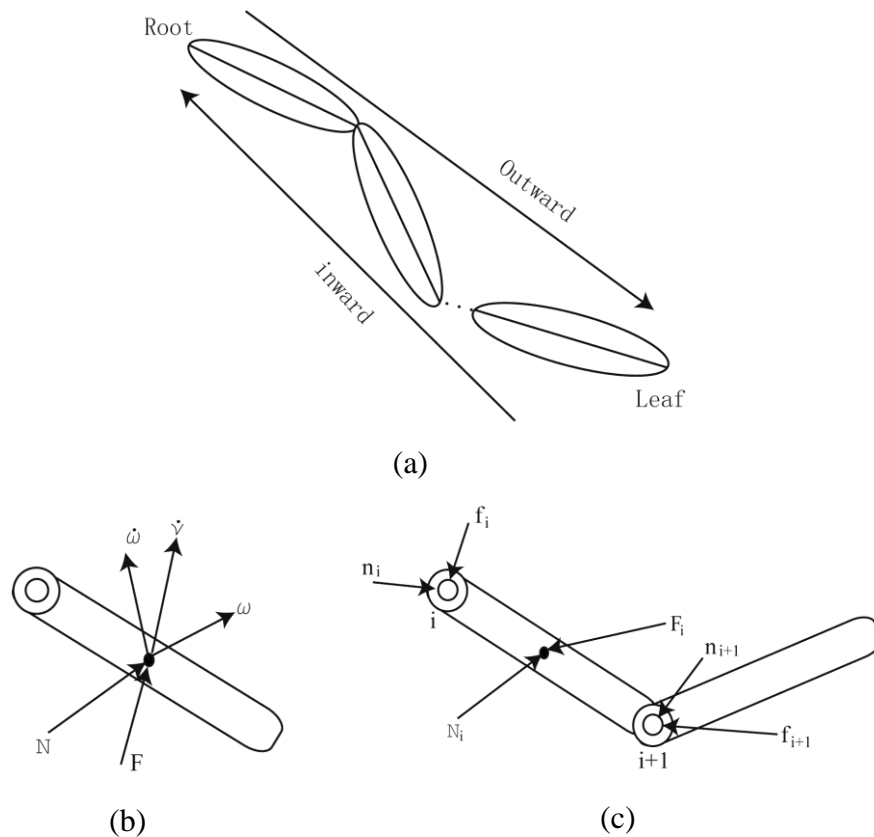


Figure A (a) The procedure of iterative Newton-Euler dynamic formulation, outward and inward process; (b) A single rigid link rotate with velocity ω and acceleration $\dot{\omega}$ due to the moment N acting on it and linear acceleration \dot{v} due to force F ; (c) A single rigid body in force balance due to different sources acting on it.

Summary of Notation

o_i Origin of the i^{th} link;

c_i Centre of mass of the i^{th} link;

ω_i^i Angular velocity of the i^{th} link;

$\dot{\omega}_i^i$ Angular acceleration of the i^{th} link;

z_i^i Unit vector of joint axis i of the i^{th} link;

P_j^i Distance vector from o_i to o_j in the i^{th} link frame;

R_i^{i-1} 3×3 transformation matrix relating the i^{th} link frame to the $(i-1)^{th}$ link frame;

-
- $I_i^{c_i}$ Inertia tensor of the i^{th} link in its frame;
 $F_{c_i}^i$ Force vector acting on c_i of i^{th} link;
 $N_{c_i}^i$ Moment vector about c_i in the i^{th} link;
 f_i^{i-1} Force vector exerted on i^{th} link by $(i-1)^{th}$ link;
 τ_i Torque at joint i ;
 g Gravity;
 m_i Mass of the i^{th} link;

Outward Iterative Step (start from link $i+1$)

The rotational movement of joint link propagate from root joint to leaf joint.

The rotational velocity is given by

$$\omega_{i+1}^{i+1} = R_i^{i+1} \omega_i^i + \dot{q}_{i+1} z_{i+1}^{i+1} \quad (\text{A.1})$$

By differentiate equation (A.1), the angular acceleration between each link can be derived by

$$\dot{\omega}_{i+1}^{i+1} = R_i^{i+1} \dot{\omega}_i^i + R_i^{i+1} \omega_i^i \times \dot{q}_{i+1} z_{i+1}^{i+1} + \ddot{q}_{i+1} z_{i+1}^{i+1} \quad (\text{A.2})$$

The linear acceleration of each joint origin is obtained by

$$\dot{v}_{i+1}^{i+1} = R_i^{i+1} (\dot{\omega}_i^i \times P_{i+1}^i + \omega_i^i \times (\omega_i^i \times P_{i+1}^i) + \dot{v}_i^i) \quad (\text{A.3})$$

According to the linear acceleration of joint origin, the linear acceleration of the centre of mass of each link can be obtained as below,

$$\dot{v}_{c_{i+1}}^{i+1} = \dot{\omega}_{i+1}^{i+1} \times P_{c_{i+1}}^{i+1} + \omega_{i+1}^{i+1} \times (\omega_{i+1}^{i+1} \times P_{c_{i+1}}^{i+1}) + \dot{v}_{i+1}^{i+1} \quad (\text{A.4})$$

Having computed the linear and angular accelerations of the centre of mass of each link, it is able to apply the Newton-Euler equations (6.11) to compute the inertial force and torque acting at the centre of mass of each link. As shown in Figure A (b), then the following equations can be obtained:

$$F_{i+1}^{i+1} = m_{i+1} \dot{v}_{c_{i+1}}^{i+1} \quad (\text{A.5})$$

$$N_{i+1}^{i+1} = I_{i+1}^{c_{i+1}} \dot{\omega}_{i+1}^{i+1} + \omega_{i+1}^{i+1} \times I_{i+1}^{c_{i+1}} \omega_{i+1}^{i+1} \quad (\text{A.6})$$

[Lau and Kuffner 2005] Lau, M. and Kuffner, J.J., 2005. Behavior Planning for Character Animation. *Proceedings of 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 271-280. ISBN: 1-7695-2270-X, DOI: 10.1145/1073368.1073408.

[Lee et al. 2002] Lee, J., Chai, J., Reitsma, P., Hodgins, J.K. and Pollard, N.S., 2002. Interactive Control of Avatars Animated with Human Motion Data. *ACM Transactions on Graphics*, 21(3), pp 491-500. ISSN: 0730-0301, DOI: 10.1145/566654.566607.

[Lee and Lee 2004] Lee, J. and Lee, K.H., 2004. Precomputing Avatar Behavior from Human Motion Data. *Proceedings of 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 79-87. ISBN: 3-905673-14-2, ISSN: 1712-5288, DOI: 10.1145/1028523.1028535.

[Lee and shin 1999] Lee, J. and Shin, S.Y., 1999. A Hierarchical Approach to Interactive Motion Editing for Human-like Figure. *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp 39-48. ISBN: 0-201-48560-5, DOI: 10.1145/311535.311539.

[LeVeau 1992] LeVeau, B.F., 1992. Williams & Lissner's Biomechanics of Human Motion. 3rd ed, W.B. Saunders Company, ISBN: 0-7216-5743-5.

[Lim and Thalmann 2002] Lim, I.S. and Thalmann, D., 2002. Construction of Animation Models out of Captured Data. In Proc. IEEE Conf. Multimedia and Expo, pp 829-832. ISBN: 0-7803-7304-9, DOI: 10.1109/ICME.2002.1035910.

[Liu et al. 1994] Liu, Z., Gortler, S.J. and Cohen, M.F., 1994. Hierarchical Spacetime Control. *SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pp 35-42. ISBN: 0-89791-667-0, DOI: 10.1145/192161.192169.

[Liu and Popović 2002] Liu, C.K. and Popović, Z., 2002. Synthesis of Complex Dynamic Character Motion from Simple Animations. *ACM Transactions on Graphics*, 21(3), pp 408-416. ISBN: 1-58113-521-1, ISSN: 0730-0301, DOI: 10.1145/566654.566596.

[Mathworks 2002] Mathworks, 2002.
Available from: <http://www.mathworks.com/> [Accessed October 2002].

REFERENCES

[Abe et al. 2004] Abe, Y., Liu, C.K. and Popović, Z., 2004. Momentum-based Parameterization of Dynamic Character Motion. *In Proceedings of ACM SIGGRAPH/Eurographics symposium on Computer animation 2004*, pp 173-182. ISSN: 1727-5288, ISBN: 3-905673-14-2, DOI: 10.1145/1028523.1028546.

[Abe et al. 2007] Abe, Y., da Silva, M. and Popović, J., 2007. Multiobjective control with frictional contacts. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 249-258. ISBN: 978-1-59593-624-4.

[AccessMocap 2005] AccessMocap, 2005.
Available from: <http://www.accessmocap.co.uk/> [Accessed May 2005]

[Alexa and Müller 2000] Alexa, M. and Müller, W., 2000. Representing Animations by Principal Components. *Computer Graphics Forum, (Eurographics 2000) 19(3)*, pp 411-418. ISSN: 0167-7055, DOI: 10.1111/1467-8659.00433.

[Arikan and Forsyth 2002] Arikan, O. and Forsyth, D.A., 2002. Interactive Motion Generation from Examples. *SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, ACM Transactions on Graphics*, 21(3), pp 483-490. ISSN: 0730-0301, DOI: 10.1145/566654.566606.

[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), pp 402-408. ISSN: 0730-0301, DOI: 10.1145/882262.882284.

[Arikan et al. 2005] Arikan, O., Forsyth, D.A. and O'Brien J.F., 2005. Pushing People Around. *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 59-66. ISBN: 1-7695-2270-X, DOI: 10.1145/1073368.1073376.

[Armstrong and Green 1985] Armstrong, W.W. and Green, M., 1985. The dynamics of Articulated Rigid Bodies for Purposes of Animation. *The Visual Computer*, 1(4), pp 231-240. ISSN: 0178-2789, DOI: 10.1007/BF02021812.

[Ascension 2003] Ascension Technology, 2003. Available from: <http://www.ascension-tech.com/> [Accessed December 2003]

[Autodesk 2002] Autodesk Maya, 2002. Original owned by Aliaswavefront. Available from: <http://www.aliaswavefront.com/> [Accessed May 2002]

[Autodesk 2007] Autodesk MotionBulider, 2007. Available from: <http://www.autodesk.com/> [Accessed May 2007]

[Aydin and Nakajima 1999] Aydin, Y. and Nakajima, M., 1999. Balanced Control and Mass Center Adjustment of Articulated Figures in Interactive Environments. *The Visual Computer*, 15(3), pp 113-123. ISSN: 0178-2789, DOI: 10.1007/S003710050166.

[Baraff 1989] Baraff, D., 1989. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. *SIGGRAPH Computer Graphics*, 23(3), pp 223-232. ISBN: 0-89791-312-4, DOI: 10.1145/74333.74356.

[Barbič et al. 2004] Barbič, J., Safonova, A., Pan, J., Faloutsos, C., Hodgins, J.K. and Pollard, N.S., 2004. Segmenting Motion Capture Data into Distinct Behaviours. *GI '04: Proceedings of Graphics Interface 2004*, pp 185-194. ISSN: 0713-5424, ISBN: 1-56881-227-2.

[Barclay et al. 1997] Barclay, A., Gill, P.E. and Rose, J.B., 1997. *SQP methods and their application to numerical optimal control*. Report NA 97-3, Department of Mathematics, University of California, San Diego.

[Bertalmio et al. 2000] Bertalmio, M., Sapiro, G., Caselles, V. and Ballester, C., 2000. Image Inpainting. *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp 417-424. ISBN: 1-58113-208-5, DOI: 10.1145/34779.344972.

-
- [Bodenheimer et al. 1997] Bodenheimer, B., Rose, C., Rosenthal, S. and Pella, J., 1997. The Process of Motion Capture: Dealing with the Data. *Computer Animation and Simulation '97, Eurographics Animation Workshop, Springer-Verlag*, pp 3-18. ISBN: 978-3-211-83257-8
- [Boulic et al 2003] Boulic, R., Callennec, B.L., Herren, M. and Bay, H., 2003. Experimenting Prioritized IK for Motion Editing. *Proceedings of Eurographics 2003*. ISSN: 0946-2767.
- [Boulic et al. 1995] Boulic, R., Mas, R. and Thalmann, D., 1995. Position Control of the Center of Mass for Articulated Figures in Multiple Support. In *6th Eurographics Workshop on Animation and Simulation*, pp 130-143.
- [Boulic and Thalmann 1992] Boulic, R. and Thalmann, D. 1992. Combined Direct and Inverse Kinematic Control for Articulated Figure Motion Editing. *Computer Graphics Forum*, 11(4), pp 189-202. ISSN: 0167-7055, DOI: 10.1111/1467-8659.1140189.
- [Brand and Hertzmann 2000] Brand, M. and Hertzmann, A., 2000. Style Machines. *SIGGRAPH '00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp 183-192. ISBN: 1-58113-208-5, DOI: 10.1145/344779.344865.
- [Brotman and Netravali 1988] Brotman, L.S. and Netravali, A.N., 1988. Motion Interpolation by Optimal Control. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, pp 309-315. ISBN: 0-89791-275-6, DOI: 10.1145/54852.378531.
- [Bruderlin and Calvert 1989] Bruderlin, A. and Calvert, T.W., 1989. Goal-directed, Dynamic Animation of Human Walking. In *Proceedings of the 16th Annual conference on Computer Graphics and Interactive Techniques*, pp 233-242. ISBN: 0-89791-312-4, DOI: 10.1145/74333.74357.
- [Bruderlin and Williams 1995] Bruderlin, A. and Williams, L., 1995. Motion Signal Processing. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp 97-104. ISBN: 0-89791-701-4, DOI: 10.1145/218380.218421.
- [Chai and Hodgins 2005] Chai, J. and Hodgins, J.K., 2005. Performance Animation from Low-dimensional Control Signals. *ACM Transactions on Graphics*, 24(3), pp 686-696. ISSN: 0730-0301, DOI: 10.1145/1186822.1073248.

-
- [Choi and Ko 1999] Choi, K. and Ko, H., 1999. On-line Motion Retargetting. *The Journal of Visualization and Computer Animation*, 11(5), pp 223-235. John Wiley & Sons, Ltd., ISSN: 1099-1778, DOI: 10.1002/1099-1778(200012)11:5<223::AID-VIS236>3.0.CO;2-5.
- [Choi et al. 2003] Choi, M.G., Lee, J. and Shin S.Y., 2003. Planning Biped Locomotion using Motion Capture Data and Probabilistic Roadmaps. *ACM Transactions on Graphics*, 22(2), pp 182-203. ISSN: 0730-0301, DOI: 10.1145/636886.636889.
- [Cohen 1992] Cohen, M.F., 1992. Interactive Spacetime Control for Animation. *SIGGRAPH Computer Graphics*, 26(2), pp 293-302. ISSN: 0097-8930, DOI: 10.1145/133994.134083.
- [Craig 1989] Craig, J.J., 1989. *Introduction to Robotics, Mechanics and Control*. 2nd ed, Addison Wesley, ISBN: 0-201-09528-9.
- [Dasgupta and Nakamura 1999] Dasgupta, A. and Nakamura, Y., 1999. Making Feasible Walking Motion of Humanoid Robots from Human Motion Capture Data. In *Proceedings of IEEE International Conference on Robotics and Automation*, vol.2, pp 1044-1049. ISBN: 0-7803-5180-0, DOI: 10.1109/ROBOT.1999.772454.
- [Faloutsos et al. 2001] Faloutsos, P., Van De Panne, M. and Terzopoulos, D., 2001. Composable Controllers for Physics-based Character Animation. *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp 251-260. ISBN: 1-58113-374-X, DOI: 10.1145/383259.383287.
- [Fang and Pollard 2003] Fang, A.C. and Pollard, N.S., 2003. Efficient Synthesis of Physically Valid Human Motion. *ACM Transactions on Graphics*, 22(3), pp 417-426. ISSN: 0730-0301, DOI: 10.1145/882262.882286.
- [Featherstone 1987] Featherstone, R., 1987. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, ISBN: 0-89838-230-0.
- [Fletcher 2000] Fletcher, R. 2000. *Practical Methods of Optimization*. 2nd ed, Wiley, ISBN: 0-471-49463-1.
- [Girard and Maciejewski 1985] Girard, M. and Maciejewski, A.A., 1985. Computational Modeling for the Computer Animation of Legged Figure. *ACM SIGGRAPH Computer Graphics*, 19(3), pp 263-270. ISSN: 0097-8930, DOI: 10.1145/325334.325244.

[Glardon et al. 2004] Glardon, P., Boulic, R. and Thalmann, D., 2004. PCA-Based Walking Engine Using Motion Capture Data. *CGI '04: Proceedings of the Computer Graphics International*, pp 292-298. ISBN: 0-7695-217-1, ISSN: 1530-1052, DOI: 10.1109/CGI.2004.73.

[Gleicher 1997] Gleicher, M., 1997. Motion Editing with Spacetime Constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pp 139-148. ISBN: 0-89791-884-3, DOI: 10.1145/253284.253321.

[Gleicher 1998] Gleicher, M., 1998. Retargetting Motion to New Characters. *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp 33-42. ISBN: 0-89791-999-8, DOI: 10.1145/280814.280820.

[Gleicher 2001] Gleicher, M., 2001. Motion Path Editing. *I3D '01: Proceedings of the 2001 Symposium on Interactive 3D Graphics*, pp 195-202. ISBN: 1-58113-292-1, DOI: 10.1145/364338.364400.

[Heath 1997] Heath, M.T., 1997. *Scientific Computing: An Introduction Survey*. International editions, McGraw-Hill, ISBN: 0-07-027684-6.

[Hodgins et al. 1995] Hodgins, J.K., Wooten, W.L., Brogan, D.C. and O'Brien, J.F., 1995. Animating Human Athletics. *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp 71-78. ISBN: 0-89791-701-4, DOI: 10.1145/218380.218414.

[Hodgins and Pollard 1997] Hodgins, J.K. and Pollard, N.S., 1997. Adapting Simulated Behaviors for New Characters. *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp 153-162. ISBN: 0-89791-896-7, DOI: 10.1145/258734.258822.

[Honda 2003] Honda Motor Co. Ltd., 2003.
Available from: <http://www.honda.co.jp/english/technology/robot/>.
[Accessed March 2003]

[Hsu et al. 2007] Hsu, E., da Silva, M. and Popović, J., 2007. Guided Time Warping for Motion Editing. *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 42-52. ISBN: 978-1-59593-624-4.

-
- [Hsu et al. 2004a] Hsu, E., Gentry, S. and Popović, J., 2004. Example-Based Control of Human Motion. *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp 69-77. ISBN: 3-905673-14-2, ISSN: 1727-5288, DOI: 10.1145/1028523.1028534.
- [Hsu et al. 2005] Hsu, E., Pulli, K. and Popović, J., 2005. Style Translation for Human Motion. *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pp 1082-1089. DOI: 10.1145/1186822.1073315.
- [Huang 1997] Huang, Z.Y., 1997. *Motion control for Human Animation*. Ph.D. thesis, Swiss Federal Institute of Technology.
- [Inman et al. 1981] Inman, V.T., Ralston, H.J. and Todd, F., 1981. *Human Walking*. Williams & Witkins, Baltimore, ISBN: 0-683-04348-X.
- [Isbweb 2003] International Society of Biomechanics, 2003. Available from: <http://www.isbweb.org/> [Accessed March 2003].
- [Issacs and Cohen 1987] Issacs, P.M. and Cohen, M.F., 1987. Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics. *SIGGRAPH Computer Graphics*, 21(4), pp 215-224. ISBN: 0-89791-227-6, DOI: 10.1145/37402.37428.
- [Jenkins and Mataric 2002] Jenkins, O.C. and Mataric, M.J., 2002. Deriving Action and Behavior Primitives from Human Motion Data. *In Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 2551-2556.
- [Jolliffe 2002] Jolliffe, I.T., 2002. *Principle Component Analysis*. 2nd ed. Springer series in statistics. Springer-Verlag, ISBN: 0-387-95442-2.
- [Kirk et al. 2005] Kirk, A.G., O'Brien, J.F. and Forsyth, D.A., 2005. Skeletal Parameter Estimation from Optical Motion Capture Data. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2005*, pp 782-788. ISBN: 0-7695-2372-2, ISSN: 1063-6919, DOI: 10.1109/CVPR.2005.326.
- [Ko 1994] Ko, H., 1994. *Kinematic and Dynamic techniques for Analyzing, Predicting, and animating Human Locomotion*. Ph.D. thesis, University of Pennsylvania.

[Komura et al. 2004] Komura, T., Leung, H. and Kuffner, J., 2004. Animating Reactive Motions for Biped Locomotion. *VRST '04: Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp 32-40. ISBN: 1-58113-907-1, DOI: 10.1145/1077534.1077542.

[Komura et al. 2000] Komura, T., Shinagawa, Y. and Kunii, T.L., 2000. Creating and Retargetting Motion by the Musculoskeletal Human Body Model. *The Visual Computer*, 16(5), pp 254-270. ISSN: 0178-2789, DOI: 10.1007/S003719900065.

[Kovar 2004] Kovar, L., 2004. Automated Methods for Data-Driven Synthesis of Realistic and Controlable Human Motion. Ph.D. thesis, University of Wisconsin-Madison.

[Kovar and Gleicher 2003] Kovar, L. and Gleicher, M., 2003. Flexible Automatic Motion Blending with Registration Curves. *In Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 214-224. ISBN: 1-58113-659-5, ISSN: 1727-5288.

[Kovar and Gleicher 2004] Kovar, L. and Gleicher, M., 2004. Automated Extraction and Parameterization of Motions in Large Data Sets. *ACM Transactions on Graphics*, 23(3), pp 559-568. ISSN: 0730-0301, DOI: 10.1145/1015706.1015760.

[Kovar et al. 2002a] Kovar, L., Gleicher, M. and Pighin, F., 2002. Motion Graphs. *ACM Transactions on Graphics*, 21(3), pp 473-482. ISSN: 0730-0301, DOI: 10.1145/566654.566605

[Kovar et al. 2002b] Kovar, L., Schreiner, J. and Gleicher, M., 2002. Footskate Cleanup for Motion Capture Editing. *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 97-104. ISBN: 1-58113-573-4, DOI: 10.1145/545261.545277.

[Kwon and Shin 2005] Kwon, T. and Shin, S.Y., 2005. Motion Modeling for On-Line Locomotion Synthesis. *Proceedings of 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 29-38. ISBN: 1-7695-2270-X, DOI: 10.1145/1073368.1073373.

[Laszlo et al. 1996] Laszlo, J., Van De Panne, M. and Fiume, E., 1996. Limit Cycle Control and its Application to the Animation of Balancing and Walking. *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp 155-162. ISBN: 0-89791-746-4, DOI: 10.1145/237170.237231.

[Lau and Kuffner 2005] Lau, M. and Kuffner, J.J., 2005. Behavior Planning for Character Animation. *Proceedings of 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 271-280. ISBN: 1-7695-2270-X, DOI: 10.1145/1073368.1073408.

[Lee et al. 2002] Lee, J., Chai, J., Reitsma, P., Hodgins, J.K. and Pollard, N.S., 2002. Interactive Control of Avatars Animated with Human Motion Data. *ACM Transactions on Graphics*, 21(3), pp 491-500. ISSN: 0730-0301, DOI: 10.1145/566654.566607.

[Lee and Lee 2004] Lee, J. and Lee, K.H., 2004. Precomputing Avatar Behavior from Human Motion Data. *Proceedings of 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 79-87. ISBN: 3-905673-14-2, ISSN: 1712-5288, DOI: 10.1145/1028523.1028535.

[Lee and shin 1999] Lee, J. and Shin, S.Y., 1999. A Hierarchical Approach to Interactive Motion Editing for Human-like Figure. *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp 39-48. ISBN: 0-201-48560-5, DOI: 10.1145/311535.311539.

[LeVeau 1992] LeVeau, B.F., 1992. Williams & Lissner's Biomechanics of Human Motion. 3rd ed, W.B. Saunders Company, ISBN: 0-7216-5743-5.

[Lim and Thalmann 2002] Lim, I.S. and Thalmann, D., 2002. Construction of Animation Models out of Captured Data. In Proc. IEEE Conf. Multimedia and Expo, pp 829-832. ISBN: 0-7803-7304-9, DOI: 10.1109/ICME.2002.1035910.

[Liu et al. 1994] Liu, Z., Gortler, S.J. and Cohen, M.F., 1994. Hierarchical Spacetime Control. *SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pp 35-42. ISBN: 0-89791-667-0, DOI: 10.1145/192161.192169.

[Liu and Popović 2002] Liu, C.K. and Popović, Z., 2002. Synthesis of Complex Dynamic Character Motion from Simple Animations. *ACM Transactions on Graphics*, 21(3), pp 408-416. ISBN: 1-58113-521-1, ISSN: 0730-0301, DOI: 10.1145/566654.566596.

[Mathworks 2002] Mathworks, 2002.
Available from: <http://www.mathworks.com/> [Accessed October 2002].

[Menache 1999] Menache, A., 1999. *Understanding Motion Capture for Computer Animation and Video Games*. 1st ed, Morgan Kaufmann, ISBN: 0-12-490630-3.

[Ménardais et al. 2004] Ménardais, S., Kulpa, R., Multon, F. and Arnaldi, B., 2004. Synchronization for Dynamic Blending of Motion. *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp 325-335. ISBN: 3-905673-14-2, ISSN: 1727-5288, DOI: 10.1145/1028523.1028567.

[McCann and Pollard 2007] McCann, J. and Pollard, N., 2007. Responsive Characters from Motion Fragments. *ACM Transactions on Graphics*, 26(3), Article No. 6. ISSN: 0730-0301, DOI: 10.1145/1275808.1276385.

[Mckenna and Zeltzer 1990] Mckenna, M. and Zeltzer, D., 1990. Dynamic Simulation of Autonomous Legged Locomotion. *SIGGRAPH '90: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, pp 29-38. ISBN: 0-89791-344-2, DOI: 10.1145/97879.97882.

[Monzani et al. 2000] Monzani, J.S., Baerlocher, P., Boulic, R. and Thalmann, D., 2000. Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargeting. *In Proceedings of Eurographics 2000, Computer Graphics Forum*, 19(3). ISSN: 0167-7055, DOI: 10.1111/1467-8659.00393.

[Mukai and Kuriyama 2005] Mukai, T. and Kuriyama, S., 2005. Geostatistical Motion Interpolation. *ACM Transactions on Graphics*, 24(3), pp 1062-1070. ISSN: 0730-0301, DOI: 10.1145/1186822.1073313.

[Müller et al. 2005] Müller, M., Röder, T. and Clausen, M., 2005. Efficient Content-based Retrieval of Motion Capture Data. *ACM Transactions on Graphics*, 24(3), pp 677-685. ISSN: 0730-0301, DOI: 10.1145/1186822.1073247.

[Nocedal and Wright 1999] Nocedal, J. and Wright, S.J., 1999. *Numerical Optimization (Springer Series in Operations Research)*. Springer-Verlag, ISBN: 0-387-98793-2.

[O'Brien et al. 2000] O'Brien, J.F., Bodenheimer, R.E., Brostow, G.J. and Hodgins, J.K., 2000. Automatic Joint Parameter Estimation from Magnetic Motion Capture Data. *Proceedings of Graphics Interface 2000*, pp 53-66.

[Oore et al. 2002] Oore, S., Terzopoulos, D. and Hinton, G., 2002. Local Physical Models for Interactive Character Animation. *In Proceedings of Eurographics 2002, Computer Graphics Forum*, 21(3). ISSN: 0167-7055, DOI: 10.1111/1467-8659.00593.

[Oshita and Makinouchi 2001] Oshita, M. and Makinouchi, A., 2001. A Dynamic Motion Control Technique for Human-like Articulated Figures. *In Proceedings of Eurographics 2001, Computer Graphics Forum*, 20(3). ISSN: 0167-7055, DOI: 10.1111/1467-8659.00512.

[Parent 2002] Parent, R., 2002. *Computer Animation: Algorithms and Techniques*. 1st ed, Morgan Kaufmann, ISBN: 1-55860-579-7.

[Park et al. 2002] Park, S.I., Shin, H.J. and Shin, S.Y., 2002. On-line Locomotion Generation Based on Motion Blending. *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 105-111. ISBN: 1-58113-573-4, DOI: 10.1145/545261.545279.

[Perlin 1995] Perlin, K., 1995. Real Time Responsive Animation with Personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1), pp 5-15. ISSN: 1077-2626, DOI: 10.1109/2945.468392.

[Phillips and Badler 1991] Phillips, C.B. and Badler, N.I., 1991. Interactive Behaviors for Bipedal Articulated Figures. *SIGGRAPH Computer Graphics*, 25(4), pp 349-358. ISSN: 0097-8930, DOI: 10.1145/127719.122756.

[Pollard et al. 2002] Pollard, N.S., Hodgins, J.K., Riley, M.J. and Atkeson, C.G., 2002. Adapting Human Motion for the Control of a Humanoid Robot. *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp 1390-1397.

[Pollard and Behmaram-Mosavat 2000] Pollard, N.S. and Behmaram-Mosavat, F., 2000. Force-based Motion Editing for Locomotion Tasks. *In Proceedings of IEEE International Conference on Robotics and Automation*, pp 663-669. ISBN: 0-7803-5889-9.

[Popović 1999] Popović, Z., 1999. *Motion Transformation by Physically Based Spacetime Optimisation*. Ph.D thesis, Carnegie Mellon University.

[Popović and Witkin 1999] Popović, Z. and Witkin, A., 1999. Physically Based Motion Transformation. *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp 11-20. ISBN: 0-201-48560-5, DOI: 10.1145/311535.311536.

-
- [Press et al. 1992] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., 1992. *Numerical Recipes in C: The Art of Scientific Computing*. 2nd ed. Cambridge University Press, ISBN: 0-521-43108-5.
- [Pullen and Bregler 2002] Pullen, K. and Bregler, C., 2002. Motion Capture Assisted Animation: Texturing and Synthesis. *ACM Transactions on Graphics*, 21(3), pp 501-508. ISBN: 1-58113-521-1, ISSN: 0730-0301, DOI: 10.1145/566570.566608.
- [Raibert 1986] Raibert, M.H., 1986. *Legged Robots that Balance*. MIT Press.
- [Raibert and Hodgins 1991] Raibert, M.H. and Hodgins, J.K., 1991. Animation of Dynamic Legged Locomotion. *SIGGRAPH Computer Graphics*, pp 349-358. ISSN: 0097-8930, DOI: 10.1145/127719.122755.
- [Ren et al. 2005] Ren, L., Patrick, A., Efros, A.A., Hodgins, J.K. and Rehg, J.M., 2005. A Data-driven Approach to Quantifying Natural Human Motion. *ACM Transactions on Graphics*, 24(3), pp 1090-1097. ISSN: 0730-0301, DOI: 10.1145/1073204.1073316.
- [Rogers and Adams 1990] Rogers, D.F. and Adams, J.A., 1990. *Mathematical Elements for Computer Graphics*. 2nd ed, McGraw-Hill, ISBN: 0-07-053529-9.
- [Rose et al. 1998] Rose, C., Cohen, M.F. and Bodenheimer, B., 1998. Verb and Advrbs: Multidimensional Motion Interpolation. *IEEE Computer Graphics and Applications*, 18(5), pp 32-41. ISSN: 0272-1716, DOI: 10.1109/38.708559.
- [Rose et al. 1996] Rose, C., Guenter, B., Bodenheimer, B. and Cohen, M., 1996. Efficient Generation of Motion Transitions Using Spacetime Constrains. *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp 147-154. ISBN: 0-89791-746-4, DOI: 10.1145/237170.237229.
- [Safonova 2006] Safonova, A., 2006. *Reducing the Search Space for Physically Realistic Human Motion Synthesis*, Ph.D. thesis, Carnegie Mellon University.
- [Safonova and Hodgins 2005] Safonova, A. and Hodgins, J.K., 2005. Analyzing the Physical Correctness of Interpolated Human Motion. *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 171-180. ISBN: 1-7695-2270-X, DOI: 10.1145/1073368.1073392.

-
- [Safonova and Hodgins 2007] Safonova, A. and Hodgins, J.K., 2007. Construction and Optimal Search of Interpolated Motion Graphs. *ACM Transactions on Graphics*, 26(3), Article No. 106. ISSN: 0730-0301, DOI: 10.1145/1275808.1276510.
- [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), pp 514-521. ISSN: 0730-0301, DOI: 10.1145/1015706.1015754.
- [Shin et al. 2003] Shin, H.J., Kovar, L. and Gleicher, M., 2003. Physical Touch-Up of Human Motions. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, IEEE Computer Society*. ISBN: 0-7695-2028-6. DOI: 10.1109/PCCGA.2003.1238261.
- [Shoemake 1985] Shoemake, K., 1985. Animating Rotation with Quaternion Curves. *SIGGRAPH Computer Graphics*, 19(3), pp 245-254. ISSN: 0097-8930, DOI: 10.1145/325334.325242.
- [Silaghi et al. 1998] Silaghi, M.C., Plänkers, R., Boulic, R., Fua, P. and Thalmann, D., 1998. Local and Global Skeleton Fitting Techniques for Optical Motion Capture. *CAPTECH '98: Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, pp 26-40, Springer-Verlag, ISBN: 3-540-65353-8.
- [Sok et al. 2007] Sok, K.W., Kim, M. and Lee, J., 2007. Simulating Biped Behaviors from Human Motion Data. *ACM Transactions on Graphics*, 26(3), Article 107. ISSN: 0730-0301, DOI: 10.1145/1276377.1276511.
- [Sun and Metaxas 2001] Sun, H.C. and Metaxas, D.M., 2001. Automating Gait Generation. *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp 261-270. ISBN: 1-58113-374-X, DOI: 10.1145/383259.383288.
- [Tak et al. 2000] Tak, S., Song, O.Y. and Ko, H.S., 2000. Motion Balance Filtering. In *Proceedings of Eurographics 2000, Computer Graphics Forum*, 19(3), pp 437-446. ISSN: 0167-7055, DOI: 10.1111/1467-8659.00436.
- [Troje 2002] Troje, N.F., 2002. Decomposing Biological Motion: A Framework for Analysis and Synthesis of Human Gait Patterns. *Journal of Vision*, 2(5):2, pp 371-387. ISSN: 1534-7362, DOI: 10.1167/2.5.2.

-
- [Ude et al. 2000] Ude, A., Man, C., Riley, M. and Atkeson, C.G., 2000. Automatic Generation of Kinematic Models for the Conversion of Human Motion Capture Data into Humanoid Robot Motion. *Proceedings of IEEE Humanoids 2000*.
- [Unuma et al. 1995] Unuma, M., Anjyo, K. and Takeuchi, R., 1995. Fourier Principles for Emotion-based Human Figure Animation. *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp 91-96. ISBN: 0-89791-701-4, DOI: 10.1145/218380.218419.
- [Urtasun et al. 2004] Urtasun, R., Glardon, P., Boulic, R., Thalmann, D. and Fua, P., 2004. Style-Based Motion Synthesis. *Computer Graphics Forum*, 23(4), pp 799-812. ISSN: 0167-7055, DOI: 10.1111/j.1467-8659.2004.00809.x.
- [Van De Panne 1997] Van De Panne, M., 1997. From Footprints to Animation. *Computer Graphics forum*, 16(4), pp 211-223. ISSN: 0167-7055, DOI: 10.1111/1467-8659.00181.
- [Vince 1992] Vince, J., 1992. *3D Computer Animation*. Wokingham: Addison-Wesley, ISBN: 0-201-62756-6.
- [Vukobratovic et al. 1990] Vukobratovic, M., Borovac, B., Surla, D. and Stokic, D., 1990. *Biped Locomotion: Dynamics, stability, Control and Application*. 1st ed, New York, Berlin, Heidelberg: Springer-Verlag, ISBN: 0-387-17456-7.
- [Wang and Bodenheimer 2004] Wang, J. and Bodenheimer, B., 2004. Computing the Duration of Motion Transitions: An Empirical Approach. *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp 335-344. ISBN: 3-905673-14-2, ISSN: 1727-5288, DOI: 10.1145/1028523.1028568.
- [Winter 1990] Winter, D.A., 1990. *Biomechanics and Motor Control of Human Movement*. 2nd ed, New York: Wiley-Interscience, ISBN: 0-471-50908-6.
- [Witkin and Kass 1988] Witkin, A. and Kass, M., 1988. Spacetime Constraints. *SIGGRAPH '88: Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, 22(4), pp 159-168. ISBN: 0-89791-275-6, DOI:10.1145/378456.378507.

-
- [Witkin and Popović 1995] Witkin, A. and Popović, Z., 1995. Motion Warping. *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp 105-108. ISBN: 0-89791-701-4, DOI: 10.1145/218380.218422.
- [Xiao et al. 2008] Xiao, Z., Nait-Charif, H. and Zhang, J.J., 2008, Automatic Estimation of Skeleton Motion from Optical Motion Capture Data. *Lecture Notes in Computer Science. Motion in Games 2008*, pp 144 - 153. Springer: Berlin-Heidelberg, ISBN: 978-3-540-89219-9, ISSN: 0302-9743, DOI: 10.1007/978-3-540-89220-5.
- [Xiao et al. 2009a] Xiao, Z., Nait-Charif, H. and Zhang, J.J., 2009, Incomplete Motion Data Reconstruction for Character Animation. *In Proceeding of IADIS Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP 2009) Conference*, pp 89 – 98.
- [Xiao et al. 2009b] Xiao, Z., Nait-Charif, H. and Zhang, J. J., 2009, Real Time Automatic Skeleton and Motion Estimation for Character Animation. *Journal of Computer Animation and Virtual Worlds*, 20(5-6), pp 523 – 531. John Wiley & Sons, Ltd, ISSN: 1546-427X (online), ISSN: 1546-4261 (print), DOI: 10.1002/cav.277.
- [Yamane et al. 2004] Yamane, K., Kuffner, J.J. and Hodgins, J.K., 2004. Synthesizing Animations of Human Manipulation Tasks. *ACM Transactions on Graphics*, 23(3), pp 532-539. DOI: 10.1145/186562.1015756.
- [Yamane and Nakamura 2003] Yamane, K. and Nakamura, Y., 2003. Dynamics Filter - Concept and Implementation of on-line Motion Generator for Human Figures. *IEEE Transactions on Robotics and Automation*, 19(3), pp 421-432. ISBN: 0-7803-5886-4, DOI: 10.1109/ROBOT.2000.844132.
- [Yin et al. 2007] Yin, K., Loken, K. and Van De Panne, M., 2007. SIMBICON: Simple Biped Locomotion Control. *SIGGRAPH '07: ACM Transactions on Graphics*, 26(3). ISSN: 0730-0301, DOI: 10.1145/1276377.1276509.
- [Zhao and Badler 1994] Zhao, J. and Badler, N.I., 1994. Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures, *ACM Transactions on Graphics*, 13(4), pp 313-336. ISSN: 0730-0301, DOI: 10.1145/195826.195827.

[Zordan and Hodgins 2002] Zordan, V.B. and Hodgins, J.K., 2002. Motion Capture-driven Simulations that Hit and React. *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 89-96. ISBN: 1-58113-573-4, DOI: 10.1145/545261.545276.

[Zordan et al. 2005] Zordan, V.B., Majkowska, A., Chiu, B. and Fast, M., 2005. Dynamic Response for Motion Capture Animation. *ACM Transactions on Graphics*, 24(3), pp 697-701. ISSN: 0730-0301, DOI: 10.1145/1073204.1073249.

[Zordan and Van Der Horst 2003] Zordan, V.B. and Van Der Horst, N.C., 2003. Mapping Optical Motion Capture Data to Skeletal Motion Using a Physical Model. *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp 245-250. ISBN: 1-58113-695-5, ISSN: 1727-5288.