

REALISTIC RENDERING OF A MULTI-LAYERED HUMAN BODY MODEL

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Mehmet Şahin YEŞİL

August 2003

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Uğur Güdükbay (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Bülent Özgüç

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Özgür Ulusoy

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Science

ABSTRACT

REALISTIC RENDERING OF A MULTI-LAYERED HUMAN BODY MODEL

Mehmet Şahin YEŞİL

M.S. in Computer Engineering

Supervisor: Assist. Prof. Dr. Uğur Güdükbay

August 2003

In this thesis study, a framework is proposed and implemented for the realistic rendering of a multi-layered human body model while it is moving. The proposed human body model is composed of three layers: a skeleton layer, a muscle layer, and a skin layer. The skeleton layer, represented by a set of joints and bones, controls the animation of the human body model using inverse kinematics. Muscles are represented by action lines, which are defined by a set of control points. The action line expresses the force produced by a muscle on the bones and on the skin mesh. The skin layer is modeled in a 3D modeler and deformed during animation by binding the skin layer to both the skeleton layer and the muscle layer. The skin is deformed by a two-step algorithm according to the current state of the skeleton and muscle layers. In the first step, the skin is deformed by a variant of the skinning algorithm, which deforms the skin based on the motion of the skeleton. In the second step, the skin is deformed by the underlying muscular layer. Visual results produced by the implementation is also presented. Performance experiments show that it is possible to obtain real-time frame rates for a moderately complex human model containing approximately 33,000 triangles on the skin layer.

Keywords: Human body modeling and animation, multi-layered modeling, articulated figure, kinematics, inverse kinematics, action line, skinning, deformation.

ÖZET

ÇOK KATMANLI BİR İNSAN MODELİNİN GERÇEKÇİ GÖRÜNTÜLENMESİ

Mehmet Şahin YEŞİL

Bilgisayar Mühendisliği Bölümü, Yüksek Lisans

Tez Yöneticisi: Yrd. Doç. Dr. Uğur Güdükbay

Ağustos 2003

Bu tez çalışmasında, insan vücudunun hareket ederken gerçekçi olarak görüntülenmesi (boyanması) için çok katmanlı bir insan modeli önerilmiş ve gerçekleştirilmiştir. Önerilen insan modeli, iskelet katmanı, kas katmanı ve deri katmanı olmak üzere üç katmandan oluşmaktadır. Bir grup eklem ve kemikle temsil edilen iskelet katmanı, insan modelinin ters kinematik yöntemler kullanılarak hareket ettirilmesini sağlamaktadır. Kaslar, bir grup kontrol noktasıyla tanımlanan etki doğruları ile modellenmiştir. Bu etki doğruları, kas tarafından kemik ve deri yüzeyi üzerine uygulanan kuvvetleri temsil etmektedir. Deri katmanı üç boyutlu bir modelleyici kullanılarak modellenmiştir ve bu katmanın iskelet katmanı ile kas katmanına bağlanması suretiyle canlandırma süresince derinin deforme edilmesi sağlanmıştır. Deri yüzeyi, iskelet ve kas katmanının mevcut durumlarına bağlı olarak iki aşamalı bir algoritma kullanılarak deforme edilmiştir. Birinci aşamada, deriyi iskeletin hareketine göre deforme eden bir deri kaplama algoritması kullanılmıştır. İkinci aşamada ise, deri yüzeyi, alttaki kas katmanı vasıtasıyla deforme edilmiştir. Gerçekleştirilen sistem ile üretilen görsel sonuçlar da tezde sunulmuştur. Başarım sonuçları sistemin orta karmaşıklıkta

(deri katmanında 33,000 üçgen içeren) bir model için gerçek zamanlı hızlarda çalıştığını göstermiştir.

Anahtar kelimeler: İnsan vücudunun modellenmesi ve canlandırılması, çok katmanlı modelleme, eklemlili vücut, kinematik, ters kinematik, etki doğrusu, deri kaplama, deformasyon.

ACKNOWLEDGMENTS

I gratefully thank my supervisor Assist. Prof. Dr. Uğur Güdükbay for his supervision, guidance, and suggestions throughout the development of this thesis.

I am also gratefully thankful to Prof. Dr. Bülent Özgüç for his support, guidance and great help throughout my master's study. I would also like to give special thanks to my thesis committee member Prof. Dr. Özgür Ulusoy for his valuable comments.

Besides, I would like to thank to Aydemir Memişoğlu for implementing the testbed for my research and for his appreciative support and Mustafa Akbaş for his comments and support on improving this thesis.

And very special thanks to my friend, Özgür Tüfekçi, for his patience and support.

Contents

1	INTRODUCTION	1
1.1	Human Body Modeling in General	2
1.2	Our Approach	3
1.3	Organization	5
2	RELATED WORK	6
2.1	Stick Figure Models	7
2.2	Surface Models	7
2.2.1	Rigid Body Animation	8
2.2.2	Stitching	8
2.2.3	Skinning	10
2.2.4	Deformation Based on Keyshapes	12
2.3	Volume Models	13
2.3.1	Implicit Surfaces	13
2.3.2	Collision Models	14

2.4	Physically-Based Models	15
2.5	Multi-Layered Models	16
2.5.1	Skeleton	16
2.5.2	Muscle	17
2.5.3	Skin	18
2.6	Summary	19
3	MODELING AND ANIMATION OF THE SKELETON	21
3.1	Modeling of the Skeleton Layer	21
3.1.1	Articulated Structure Based on the H-Anim Specification .	22
3.1.2	The Articulated Figure Model	24
3.2	Animation of the Skeleton Layer	25
3.2.1	Motion Control	25
3.2.2	Motion of the Skeleton	28
4	MODELING AND ANIMATION OF THE MUSCLES	30
4.1	Modeling of the Muscle Layer	30
4.1.1	Action Lines	32
4.2	Animation of the Muscle Layer	33
5	MODELING AND ANIMATION OF THE SKIN	36
5.1	Modeling of the Skin Layer	36

5.1.1	Skin Attaching	37
5.2	Animation of the Skin Layer	42
5.2.1	Skeletal Deformation	43
5.2.2	Muscular Deformation	46
6	EXPERIMENTAL RESULTS	47
6.1	Visual Results	47
6.2	Performance Analysis	51
7	CONCLUSION	54
	APPENDICES	56
A	HUMAN MODEL IN XML FORMAT	56
A.1	Document Type Definition (DTD) of Human Model	56
A.2	XML Formatted Data of Human Model	57
B	THE SYSTEM AT WORK	60
B.1	Overview	60
B.2	The Main Menu	62
B.3	The Motion Control and The Skinning Toolbox	62
B.4	The Keyframe Editor	64
B.5	The Viewing Area	65

List of Figures

2.1	Stitching components: a) bone structure and b) attached mesh structure (reproduced from [24])	9
2.2	Skinned arm mesh bent to 90 degrees and 120 degrees (reproduced from [24])	11
2.3	The elbow collapses when the forearm is twisted (reproduced from [24])	12
2.4	The elbow collapses upon flexion (reproduced from [24])	12
2.5	The modeling of volume-preserving ellipsoid (reproduced from [41])	18
3.1	The H-Anim Specification 1.1 hierarchy (from [47])	23
3.2	A portion of humanoid data in XML format.	25
3.3	The front and left view of the skeleton	26
3.4	The articulated figure and the Cardinal spline curves	28
4.1	Action line abstraction of a muscle	32
4.2	The structure of an action line: control points and forces on these points	33

4.3	The deformation of an action line and force field changes: a) rest position and initial force fields of an action line, b) action line shortens and forces increases due to muscle contraction, and c) action line lengthens and forces decreases due to muscle elongation .	35
5.1	The shaded point and solid view of the skin model	37
5.2	First step of the attaching algorithm: the right upper arm part is bound with the right shoulder joint and the left thigh part is bound with the left hip joint.	39
5.3	Second step of the attaching algorithm: the front and left view of some critical joints and their influence areas. The darken rectangles denotes the vertices that are attached with the corresponding joint.	40
5.4	The holes at the backside of the model while walking due to unattached vertices	41
5.5	Binding skin vertices with action lines. Point j is attached to its nearest three control points, v_1 , v_2 and v_3	43
5.6	The skeletal animation: a) reference position and b) after rotation	44
5.7	The skinning algorithm	45
5.8	The skinning method used in our implementation: a) vertex assignment and b) after rotation	46
6.1	The front view of the human walking behavior.	48
6.2	The back view of the human walking behavior.	49
6.3	The multi-dimensional motion of the arm of the human model. . .	50

6.4	Showing muscular deformation on the skin mesh.	51
6.5	Different resolution valued skin meshes: a) high resolution (original model), b) middle resolution, and c) low resolution.	53
B.1	Top level user interface of the system.	61
B.2	The motion control and the skinning toolbox	63
B.3	The keyframe editor	65
B.4	The viewing area	66

List of Tables

6.1	Performance evaluation results	52
-----	--	----

To my family...

Chapter 1

INTRODUCTION

For the Computer Graphics society, human body modeling and animation has long been an important and challenging area. The reason why human body modeling still remains one of the greatest challenges is two-fold. First, the human body is so complex that no current model comes even close to its true nature. The second reason is that we, as human beings, are all experts when it comes to human shapes. The sensitivity of our eyes to human figures is such that we can easily identify the unrealistic body shapes (or body motions).

Nowadays, virtual humans are more necessary for our lives. There are lots of fields in which 3D virtual characters are in action: video games, films, television, virtual reality, ergonomics, medicine, biomechanics, etc. We can classify all these applications in the following three categories: film production, real-time applications and simulations involving humans.

The first class, the film industry, has widely used the computer-generated images. Films like Jurassic Park, Titanic and Final Fantasy featured virtual humans in a very realistic manner. In addition, due to the decreasing cost of

computer-generated imagery, virtual characters are also used increasingly in commercials. For example, safety instructions on most long-distance flights are now demonstrated by digital actors.

The applications that require real-time interaction form the second category. Video games and virtual environments are fundamental examples. Most of the virtual reality applications need representation of the user and the interaction with the environment in real-time (for example virtual surgery). Military simulation is another example of real-time applications. That means virtual reality applications are mostly used for training people for specific tasks.

Lastly, virtual humans are necessary for most of the simulation applications. From computer aided ergonomics to automotive industry, virtual actors play important role. Medicine and sports also benefit from the same technology in order to get more accurate results.

1.1 Human Body Modeling in General

These necessities also shaped the computer graphics area, and the modeling and animation of human figure has been an important research problem in this area. In the seventies, Badler and Smoliar [3] and Herbison-Evans [18] firstly worked for representing 3D models by volumetric primitives. Since then many new methods and models were introduced by the researchers. The problems of these researches could be divided into two key parts: realistic modeling of the human figure, and the motion of this model. Human body modeling consists of the basic structure modeling; i.e., the definition of the joints and the segments, their positions and orientations and the hierarchy between these components. Then, the body volume, that is compound of muscles and fat, and the skin part of the model should be defined. The second part of the problem, simulating

human motion, is also a complex task. It is very difficult to take into account all the interactions with the environment involved in a simple movement.

In the early stages, we see that humans have been represented as simple articulated bodies made of segments and joints (stick figures). These articulated bodies were simulated using kinematics based methods. More recently, *dynamic methods* have been used to improve the reality of the movement. However, since the human body is a collection of complex rigid and non-rigid components that are very difficult to model, these dynamic and kinematics models did not meet the needs. Consequently, researchers began to use human anatomy to produce human models with more realistic behaviors.

1.2 Our Approach

This study proposes an anatomically based, layered structure of the human body. The structure is composed of three layers: a skeleton layer, a muscle layer and a skin layer. Our model is similar to the multi-layered anatomically based models of Wilhelms and Van Gelder [60] and Scheepers et al. [41].

The *skeleton layer* is composed of joints and bones and controls the motion of the body by manipulating the angles of the joints. The basis for the articulated structure is relied on the *Humanoid Animation (H-Anim)* standard, which is developed by Humanoid Animation Working Group of Web3D Consortium. In our model, by conforming the hierarchy and naming conventions of H-Anim standards, we used *Extensible Markup Language (XML)* data format to represent the skeleton. In the system, the motion of the skeleton is controlled by inverse kinematics methods. Spline-driven animation techniques are used, at the low-level control, to generate position and velocity curves of the joints [30]. Furthermore, arclength deficiency of these splines is handled by arclength parameterization. Inverse kinematics problem is solved by means of a package,

named *Inverse Kinematics using Analytical Methods (IKAN)*, that provides the required functionality for controlling the motion of the body parts like arms and legs.

The *muscle layer* represents the muscular structure of the human body. We control the deformation of the muscle through the joint angles of the skeleton layer under the following constraint: the insertion/origin points of the muscle on the bones are attached during animation and form the action line. In our system, the muscle layer is represented only by action lines and the muscle shape is not considered. An action line denotes the imaginary line along which the force exerted onto the bone is produced. This line also includes the forces exerted onto the skin layer that create the skin deformation due to muscular motion during the animation. It is a polyline that includes some control points, which are responsible for determining the forces mentioned above. We represented some fusiform muscles in the upper and lower parts of the legs and arms.

The *skin layer* is created through a 3D modeler and contains 53 different parts. The skin layer is deformed by the positions of the joints in the skeleton layer and the forces applied by the muscle layer. To deform the skin layer realistically during animation, we bind the vertices on the skin layer to the joints of the skeleton layer and to the action lines of the muscle layer simultaneously. This binding operation is performed in three steps. In the first step, for each object of the skin mesh, a particular joint of the skeleton is determined and the vertices of the object are attached to this joint. The second step deals with binding some vertices, especially the ones closer to the joints, with more than one joint. In the last step, some problematic vertices are handled manually. After attaching skin vertices to the skeleton, we also bind the required vertices with the muscles. At the end, the skin mesh become available for representing both skeletal and muscular deformations.

In this study, we aimed to choose the appropriate human model by studying human joints and motion capabilities. In addition to this, we tried to design an efficient method that leads us realistically and automatically deform the skin of the model based on the transformation of the inner layers. This method should also be computationally efficient.

1.3 Organization

The organization of the thesis is as follows: In Chapter 2, related work on human body modeling is given. Chapter 3 deals with the modeling and animation of the skeletal layer, which is the first layer of our multi-layered body model. Chapter 4 focuses on the modeling and animation of the second layer, muscles. In Chapter 5, the modeling and animation of the skin mesh is presented. Chapter 6 presents the visual results and the results of the performance experiments. Chapter 7 concludes and gives the future research areas. In Appendix A, the human model in XML format is presented. In Appendix B, the implementation details are given and the functionality of the different parts of the system are described.

Chapter 2

RELATED WORK

Representation and deformation of human body shape is a well studied subject in computer graphics. The models proposed up to now can be divided into four categories:

- stick figure models,
- surface models,
- volume models,
- and multi-layered models.

Due to technological limitations and the demand for rapid feedback, most of the time, curves, stick figures, and simple geometric primitives are used to represent human body at the cost of losing the realism of the animation. In order to increase realism, surface and volume models have been proposed. Finally, layered approaches that combine the skeleton layer with different layers such as muscle, skin or clothing layer have been proposed [9, 39, 44, 58]. These additional layers are used to improve the realism of the model.

2.1 Stick Figure Models

The first studies on human animation were made in 1970s [3]. Technological limitations of those days allow only stick figures with only a few joints and segments and geometric primitives. These models are built by using hierarchical set of rigid segments connected at joints. Such models are called *articulated figures* and their complexity depends on the number of joints and segments used. They provide easy control of movement at the expense of losing from realism of the animation. Studies on directed motions of articulated figures by Korein et al. [22] and stick figure model by Thalmanns [52] are good examples for this category.

2.2 Surface Models

Surface models were proposed as an improvement to the stick models. A new layer, which represents human skin, was introduced in addition to the skeleton layer [4]. Therefore, this model is based on two layers: a skeleton, which is the backbone of the character animation and a skin, which is a geometric envelop of the skeleton layer. The deformations in the skin layer are governed by the motion of the skeleton layer. This model can be examined under three topics:

- points and lines (the simplest surface model),
- polygons (used in Rendezvous a Montreal [54]),
- and curved surface patches (Bezier, Hermite, bi-cubic, B-spline and Beta-spline).

2.2.1 Rigid Body Animation

The skin layer can be modeled as a collection of polygonal meshes laying on top of the skeleton. Motion of the skin layer is achieved by attaching each mesh vertex to a specific joint. By this way, the motion of the skin layer fits the motion of the articulated structure. However, muscle behaviors is not modeled in this approach and body parts may disconnect from each other during some motions. Contrary to these deficiencies, these models are still very common in Web applications [47].

A solution to the deformation problem of this model is to use a continuous deformation function with respect to the joint points. First, Komatsu [21] used this method in order to deform the control points of biquartic Bézier patches. Then, the concept of Joint-dependent Local Deformation (JLD) was introduced by Thalmanns [53]. Both in Komatsu’s approach and in JLDs, the skin is deformed algorithmically. First of all, the skin vertices are mapped to the skeleton segments in order to limit the influence of the joint connecting corresponding segments. Next, a function of the joint angles is used to deform the vertices. These studies showed that using specialized algorithms may achieve more realistic skin deformations. However, there are two main limitations: first, mathematical functions are too limited to accurately describe the complexity of a real human body, and second, a graphic designer cannot easily control the deformations because they are specified via an algorithm.

2.2.2 Stitching

Stitching is defined as the operation on a continuous mesh attached to a bone structure as seen in Figure 2.1. In rigid body animation, polygons are attached to bones. Transformation of those polygons is accomplished by changing the matrix representing the corresponding bone. In stitching, each vertex of a polygon can

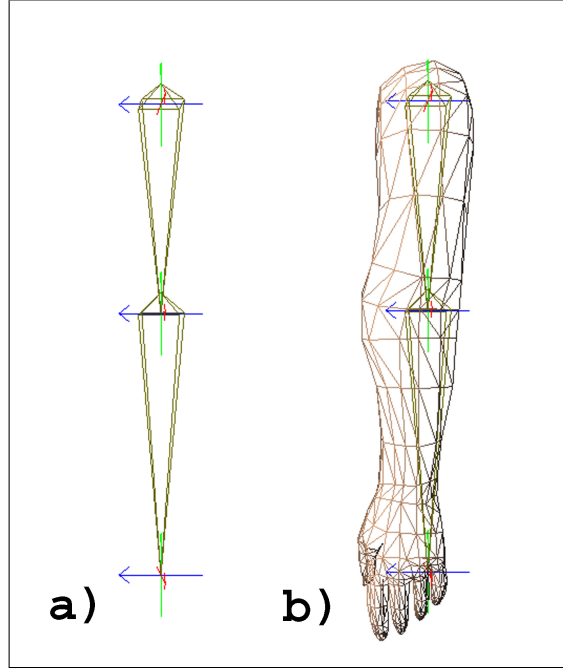


Figure 2.1: Stitching components: a) bone structure and b) attached mesh structure (reproduced from [24])

be attached to a different bone. Therefore, each vertex can be transformed by a different matrix representing the bone that the vertex is attached. This type of attachment enables us to create a single polygon that “stitch” multiple bones at the same time by attaching different vertices to different bones. A requirement for a polygon is that it should fill the gap formed as a result of a manipulation of the bones [12].

Rigid body animations and stitching differ in the data topology used for representing a character. In rigid body animations, a pointer to the bone structure that is to be animated is essential. The matrix representing the corresponding bone transforms the geometry. On the other hand, in stitching, one must store the data that lists the vertices and the corresponding bones. Once this data is available, each vertex is transformed by the orientation and the animation matrices of corresponding bones successively. Breaking up a continuous skin so that

the vertices exist in the local space of the bone that they are attached simplifies the implementation of stitching.

2.2.3 Skinning

Although stitching is a valid technique, it is not perfect and it has some problems. Unnatural geometries appear during extreme joint rotations. For example, rotating a forearm by 120 degrees using stitching technique results in a shear effect at the elbow. A solution proposed to solve this problem is to allow a vertex to be effected by more than one bone. This solution is called *full skinning* and compatible with the behavior of the human body, because, for example, the skin on one's elbow is not affected by only a single bone but both the upper and the lower arm bones affect it.

In order to implement this type of a technique, each vertex in a skinned mesh must contain a list of bones affecting it. In addition to listing the bones affecting the vertex, the weight for the bone that specifies the amount of the effect must be included. The position of each vertex is calculated using Equation 2.1 [12].

$$new\ vertex = \sum_{i=0}^{N-1} weight_i \times matrix_i \times vertex \quad (2.1)$$

where, in the case of linear skinning, the sum of all weights is 1.0.

The technique outlined above produces the skinned forearm given in Figure 2.2. It should be noted that even in the extreme 120-degree case, the continuity of the elbow is still maintained.

Skinning is a very simple technique that has a very low computational cost. For this reason, it is used in video games. Along with the recent graphics cards with matrix blending support, the computational cost is further reduced. Skinning is applicable not only to human skin deformation but also to cloth deformation. Current research focuses on the improvement of the skinning procedure

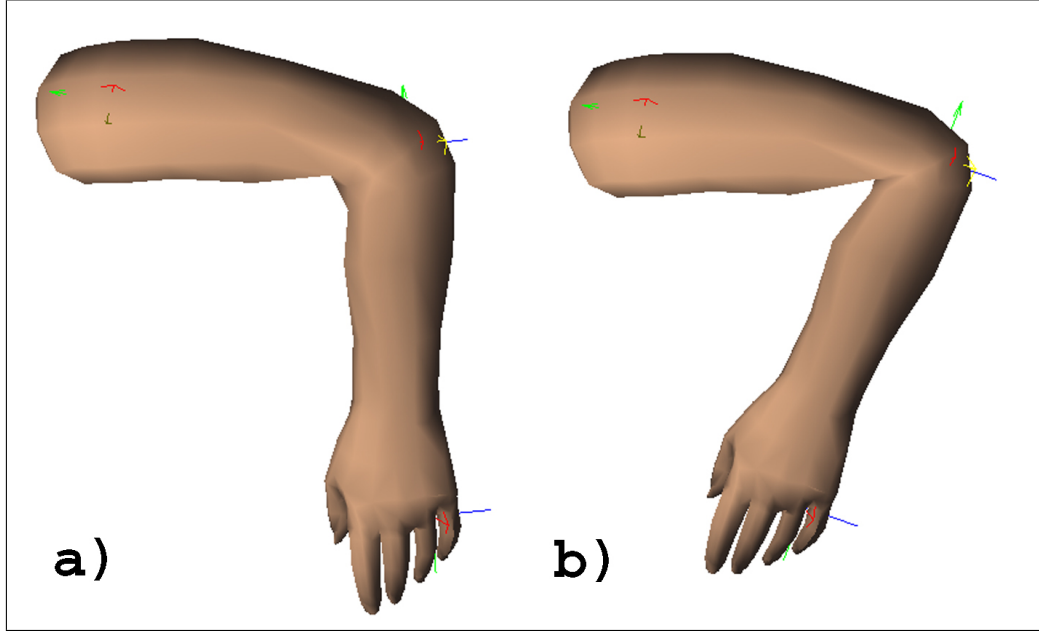


Figure 2.2: Skinned arm mesh bent to 90 degrees and 120 degrees (reproduced from [24])

by increasing the the speed of computations. Sun et al. [49] use the concept of normal-volume, i.e., they restrict the number of computations by mapping a high-resolution mesh onto a lower-resolution control mesh. Singh and Kokkevis choose surface-based FFDs to deform skins [45]. A very useful property of the surface-oriented control structures is that they bear a strong resemblance to the geometry they deform. In addition, they can be constructed from the deformable geometry automatically. An advantage of the method is that a high-resolution object can be deformed by skinning the low-resolution surface control structure.

Assigning weights is a semi-automatic process that requires huge amount of human intervention, which is a big limitation for the skinning technique. In addition, a combination of weights for highly mobile portions of the skeleton may be very appropriate for one position of the skeleton but the same weights may not be acceptable for another position. Therefore, there is no single combination of weights that provides an acceptable performance for all parts of the

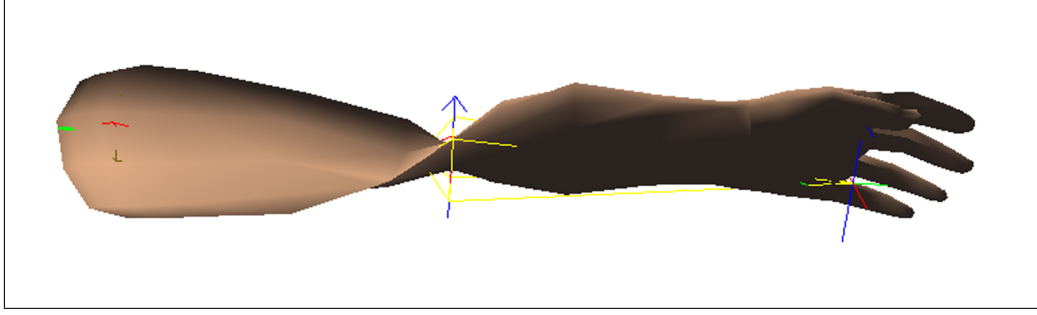


Figure 2.3: The elbow collapses when the forearm is twisted (reproduced from [24])

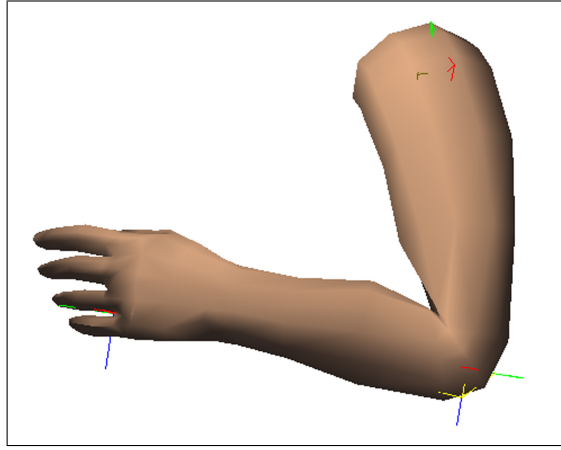


Figure 2.4: The elbow collapses upon flexion (reproduced from [24])

body. Two examples are shown in Figure 2.3 and 2.4. Contrary to these deficiencies, the skinning method is still one of the most acceptable techniques for skin deformation due to its simplicity.

2.2.4 Deformation Based on Keyshapes

Using predefined keyshapes have recently been proposed as a new technique for skin deformation [24, 46]. *Keyshapes* are some triangular meshes in some skeletal positions. They are obtained via a digitization procedure [50]. The idea behind this technique is that new shapes are created by interpolation or extrapolation.

Deformation-by-keyshapes technique differs from 3D morphing algorithms in that it is limited to smooth interpolation problems.

Deficiencies of the skinning techniques are not present in this approach. Moreover, it performs better than multi-layer deformation models [46]. There is no limitation on the number of keyshapes, the designer can add as many keyshapes as he wants. This makes the technique quite flexible.

2.3 Volume Models

Controlling the surface deformation across joints is the major problem with the surface models. In *volume models*, simple volumetric primitives like ellipsoids, spheres and cylinders are used to construct the shape of the body [61]. A good example to volume models is metaballs. They can do better than surface models. However, it is really a hard job to control a large number of primitives during the animation. In the very early studies, volumetric models were built from geometric primitives such as ellipsoids and spheres to approximate the shape of the body. These models were proposed when the computer graphics were in its early stages. Therefore, they were constrained by the limited computer hardware. Along with the advances in computer hardware technology, implicit surfaces present an interesting generalization of these early models. Today, volumetric models often allow to handle collisions.

2.3.1 Implicit Surfaces

An *implicit surface* is also known as an iso-surface. It is defined with the help of a function that assigns a scalar value to each 3D point in space. Then, an iso-surface is extracted from the level set of points that are mapped to a same scalar value. Skeletons, which are constructed from points and lines, are the source of

the scalar field. Each skeleton produces a potential field whose distribution is determined by the field function. A common choice for the field function is to use a high-order, generally more than four, polynomial of distance to the skeleton. This approach is known as the metaball formulation.

Due to their natural smoothness, implicit surfaces are often used in the representation of organic forms. One of the first examples of this method is a “blobby man” created by Blinn [5]. It is generated from an implicit surface that is constructed using an exponentially decreasing field function. In one of the later studies, Yoshimoto showed that a complete realistic-looking virtual human body can be created with metaballs at a lower storage cost [61]. A more complicated implicit formulation introduced by Bloomenthal [6, 7].

Implicit surfaces have many properties that give rise to successful body modeling. The most useful property of such surfaces is their continuity, which is the main requirement for obtaining realistic shapes. There are two advantages worth to mention: first, due to the compact formulation of the field functions, little memory is required and second, they are simple to edit since they are defined by point or polygon skeletons. However, undesired blending may be observed during animations. Hybrid techniques that are mixtures of surface deformation models and implicit surfaces are proposed as a solution to this problem [27, 55].

2.3.2 Collision Models

Handling collisions between different models or different parts of a same model and generating deformed surfaces in parallel is well achieved by some volumetric models. Elastic properties are included into the formulation of distance-based implicit surfaces by Cani-Gascuel [16]. In this work, a correspondence between the radial deformation and the reaction force is established. A non-linear finite element model of a human leg derived from the Visible Human Database [36] was

recently proposed by Hirota et al. [19]. It also achieves a high level of realism in the deformation.

2.4 Physically-Based Models

Simulation of the physical properties of object models is a new and attractive area of computer graphics. Shape deformation and motion of such models are simulated by obeying the physical rules that govern the real motion or deformation [2]. Therefore, animations become more realistic. Physical quantities such as forces, velocities, and collisions can also be included in this model. Barr, Terzopoulos, Platt, Fleischer [51], and Haumann [17] have applied substance properties of the object to model flexible elastic behavior. The discrete molecular components of the object can be considered as point masses interconnected by springs with stiffness and damping attributes. Values of spring properties are calculated from the physical properties of the object. Flexible materials and elastic surfaces can be successfully modeled using mass-spring models. Nedel simulated muscles using a mass-spring system composed of angular springs [34].

Physically based models are very good at animating “inanimate” or “not consciously moving” objects. Some fields of biomechanics tries to model the physical properties of body tissues. The major difficulties arising in modeling the dynamics of soft body parts are the structural inhomogeneity and the inelastic, time-dependent behaviors of tissues. Different types of tissues behaves different. Thus, complex models such as those presented by Fung [14] are required for accurate physical simulations of these mixed structures.

2.5 Multi-Layered Models

Muscle layer was first used by Chadwick et al. [9]. They coated their character with additional muscle layer. After that study, layered approach to character modeling became popular. Lasseter emphasized that computers provide the advantage of working an animation in layers [26]. Complex motions are easily created using the advantage of additively building an animation in different layers. The animator just specifies different kinds of constraint relationships between different layers. Then, he/she can control the global motion from a higher level.

Layered approach has been accepted both in construction and animation of computer generated characters. There are two major groups of models. The first group relies on a combination of ordinary computer graphics techniques like skinning and implicit surfaces, and tends to produce a single layer from several anatomical layers. The other group is inspired by the actual biology of the human body and tries to represent and deform every major anatomical layer, and to model their dynamic interplay.

2.5.1 Skeleton

The *skeleton layer* is an articulated structure that provides the foundation for controlling the motion [55]. Sometimes, the articulated structure is covered by material bones approximated by simple geometric primitives [19, 37, 40]. Studies on the skeleton layer is focused on the accurate characterization of the range of motion of specific joints. Different models of joint limits have been suggested in the literature: spherical polygons as boundaries for the spherical joints like the shoulder are used by Korein [23] whereas joint sinus cones for the shoulder and scapula joints were used by Maurel and Thalmann [29].

2.5.2 Muscle

The foundations for the muscles and deformations used in a muscle layer are based on *free form deformations (FFD)* [42]. Muscles construct a relationship between the control points of the FFDs and the joint angles. The possibility that the FFD box does not approximate the muscle shape very well and the FFD control points have no physical meaning can be listed as the limitations of the method.

Moccozet modeled the behavior of the hand muscles using Dirichlet free form deformation (DFFD), which is a generalization of FFD that provides a more local control of the deformation and removes the strong limitation imposed on the shape of the control box [33]. It seems that the most appropriate method for modeling organic forms is implicit surfaces. Therefore, they are widely used to model the muscle layer. In [56], implicit primitives like spheres and superquadrics are used to approximate muscles. On the other hand, in [55], the gross behavior of bones, muscles, and fat are approximated by grouped ellipsoidal metaballs with a simplified quadratic field function. This technique does not produce realistic results for highly mobile parts of the human body, in which each ellipsoidal primitive is simultaneously influenced by several joints.

Since an ellipsoid approximates quite well the appearance of a fusiform muscle, muscle models tend to use the ellipsoid as the basic building block when the deformation is purely geometric. Moreover, the analytic formulation of an ellipsoid provides a scaling ability. When the primitive is scaled along one of its axes, the volume of the primitive can easily be preserved by adequately adjusting the lengths of the two remaining axes (see Figure 2.5). Following a similar way, the ratio of the height to width can be kept constant. This is the reason why a volume-preserving ellipsoid for representing a fusiform muscle is used in [41, 59]. Scheepers' studies are unique in the sense that in addition to isotonic contractions, isometric contractions of the muscles can also occur [41].

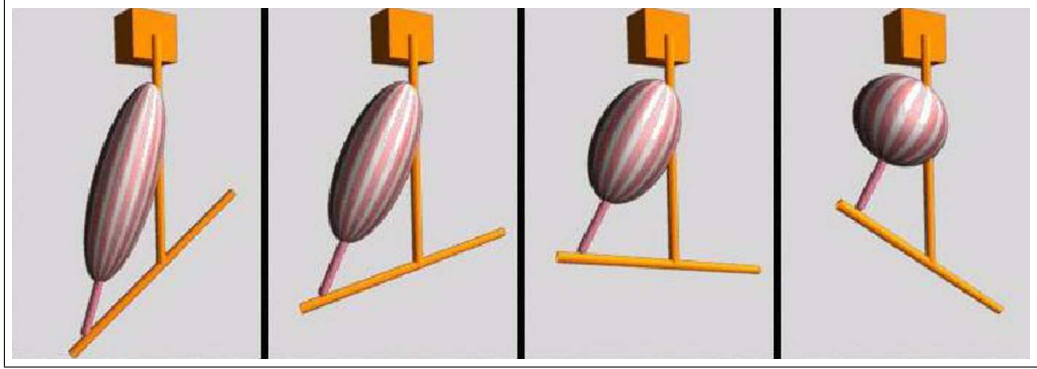


Figure 2.5: The modeling of volume-preserving ellipsoid (reproduced from [41])

A polyline called “action line” is introduced by Nedel and Thalmann [34]. The action line is used for abstracting muscles. It represents the force produced by the muscle on the bones, and a surface mesh deformed by an equivalent mass-spring network. A noteworthy feature of this mass-spring system is the introduction of angular springs. These angular springs are very useful to smooth out the surface and to control the volume variation of the muscle.

B-spline solids can also be used for modeling muscles, as described by Ng-Thow-Hing [35]. The advantage of using B-spline solids for modeling muscles is that B-spline solids can capture the multiple muscle shapes (fusiform, triangular, etc.) and can also render various size of attachments.

2.5.3 Skin

Polygonal [9, 19, 60], parametric [15, 41, 55], subdivision [13, 27], and implicit [7, 8] surfaces have been used for modeling the skin.

Polygonal surfaces are preferred due to the fact that they are processed by the graphics unit. Therefore, when speed or interactivity are needed, polygon meshes are most suitable. However, some surface discontinuities, which are required to be smoothed out, may arise when polygonal surfaces are used.

Parametric surfaces yield very smooth shapes. This makes them very attractive for modeling the skin. Shen and Thalmann [55] derive a lower degree polynomial field function for the inner layers.

As mentioned previously, implicit surfaces are very appropriate for representing organic forms. The main limitation of their usage is that it is difficult or even impossible to apply texture maps. Therefore, they are very seldom used for directly extracting a skin and used frequently for invisible anatomical layers.

Subdivision surfaces can be considered as the most proper representation for the skin layer. They have several advantages such as: i) smoothness can be guaranteed by recursively subdividing the surface, ii) a polygonal version suitable for rendering is automatically derived without further computations, and iii) interpolating schemes can be used [13].

There are three ways of deforming the skin in multi-layered models [1]:

- First, surface deformation models are applied to the skin. Then, the skin is projected back onto the inner anatomical layers.
- A mechanical model is used to deform the skin while keeping the skin a certain distance away from the material beneath.
- The skin is defined as the surface of a volume finite element (mass-spring model of the body).

2.6 Summary

Thalmanns and Badler were the first who generated realistic human models. Marilyn and Humphrey models created by Thalmanns were implemented with sampled data from sculptures [54]. Jack, whose shoulder was accurately modeled as a clavicle and shoulder pair, was introduced by Badler [4]. Wilhelms

used ellipsoids to model bones, muscles and fatty tissue in animal modeling [58]. The polygonal skin surface is attached to the inner layers, such as muscles and bones. The skin is deformed and transformed automatically when the body moves. Chadwick and his coworkers developed a structured layered model based on anatomy [9]. In his model, each layer is assigned a specific purpose; motion control is the responsibility of the skeleton layer, model shape is controlled by the skin layer which is deformed using free-form deformations (FFD). Singh et al. [43], Thalmann et al. [55] and Turner [57] were also developed multi-layer models. Scheepers's model was an improvement of multi-layered models. The improvement was the introduction of muscular deformation. In this model, the skeleton layer controls the motion of the human as before; the difference is that in addition to changes in the skeleton layer, changes in the muscle layer also affects the deformation of the skin layer [31].

Chapter 3

MODELING AND ANIMATION OF THE SKELETON

3.1 Modeling of the Skeleton Layer

The skeleton, which is defined by a set of joints and bones, is the major layer that determines the general shape of a body. A tree structure is used to represent the relationships between the joints. The root node of the tree corresponds to the center of the body whereas leaf nodes correspond to the limbs, for example, hands or feet.

In an adult body, there exist 206 bones. In addition to supporting the body weight and enabling movement, the skeleton protects the internal organs and store vital nutrients. There are two main parts of the skeleton: i) The central bones of skull, ribs, and vertebral column, and ii) bones of the arms and legs, along with the scapula, clavicle, and pelvis.

A joint or articulation is defined as the location where two body parts meet. Human joints are categorized into three groups (immovable, slightly movable, or freely movable) according to their range of motion. It is also possible to classify the joints according to their structure. Immovable joints are placed in the skull. A typical place where slightly movable joints is found is the vertebral column. The surfaces of bones in a freely movable joint are completely separated and bound by ligaments (strong fibrous bands).

3.1.1 Articulated Structure Based on the H-Anim Specification

A hierarchical articulated structure entirely controls the motion of a character. In this study, the *H-Anim* standard is accepted as the basis for the articulated structure.

The H-Anim specification [47] is the product of a joint study to standardize the representation and animation of virtual humans (see Figure 3.1). The specification is based on the *Virtual Reality Modeling Language (VRML)*. However, if the VRML file containing the human body definition can be parsed and correctly interpreted, the standard can be used by any software for human body modeling and animation. The ability to reuse any animation produced for H-Anim standard is one of the attractive benefits of using an H-Anim compliant skeleton.

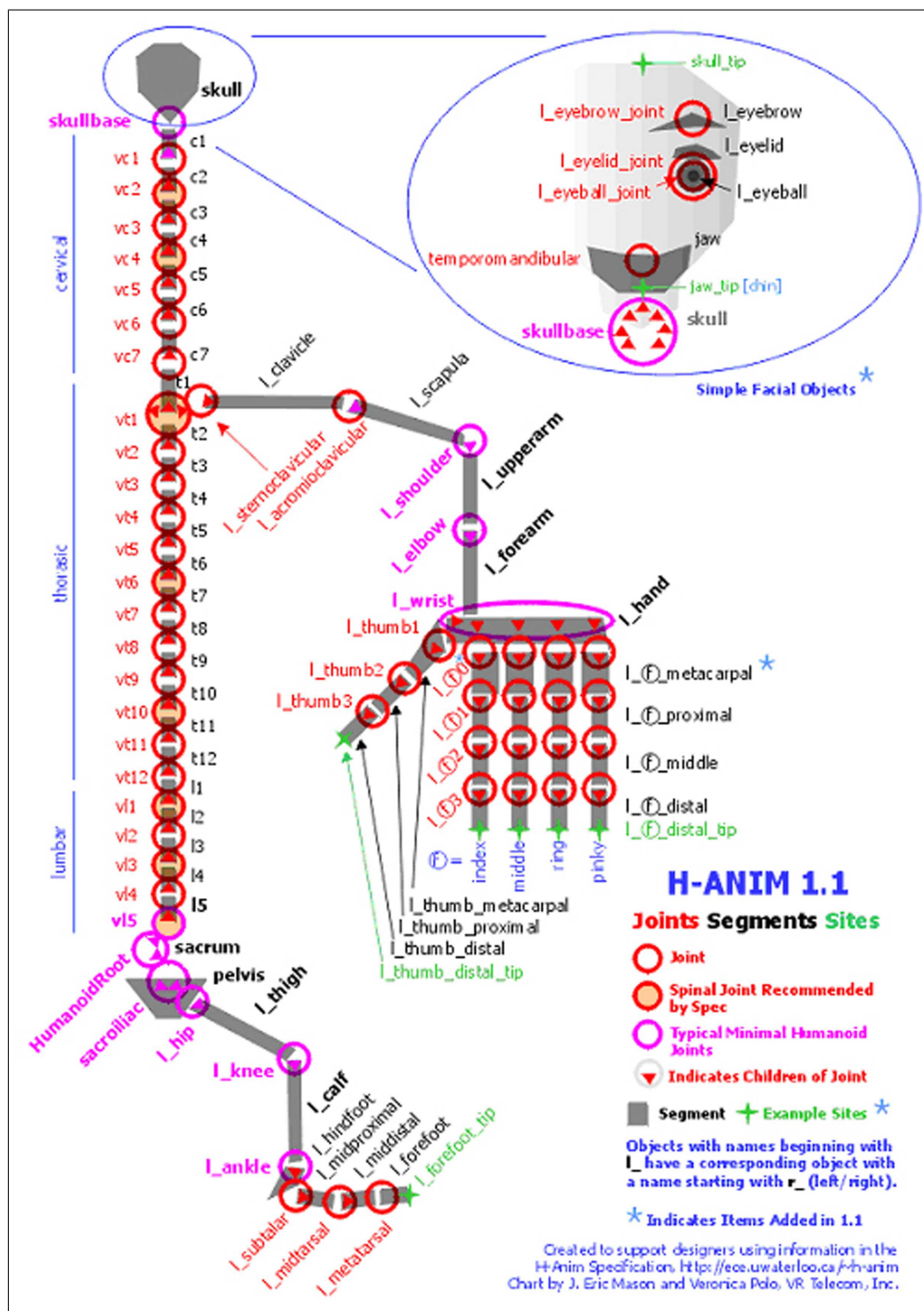


Figure 3.1: The H-Anim Specification 1.1 hierarchy (from [47])

The joints in an H-Anim file is listed in a hierarchical manner. Besides, a list of segments gathering all the information concerning the representation of body parts are arranged hierarchically, and a list of specific landmarks on the body are available. In order to support the needs of a broad range of applications, four different levels of animation are suggested. Finally, new joints (only as leaves) can be added to the hierarchical structure.

3.1.2 The Articulated Figure Model

In our studies, the human body model that is constructed by Memisoglu [30] is used. It is based on the H-Anim 1.1 Specification. The model is constructed by conforming the hierarchy of joints and segments of human body and their naming conventions given in the H-Anim 1.1 Specification. XML data format is used to represent the skeleton. XML provides assistance for describing and defining the skeleton data due to its structured, and self-descriptive format. A portion of the human model data in XML format is given in Figure 3.2 and the complete human model data in XML format can be found in Appendix A.2.

The main elements in H-Anim 1.1 Specification, namely **Humanoid**, **Joint**, and **Segment**, are used to arrange a strict hierarchy to construct the skeleton. Each element has the following attributes. **center** attribute is used for positioning the associated element. **name** attribute specifies the name of associated element. **ulimit** and **llimit** attributes are used for setting the upper and lower limits for the joint angles, respectively.

The complete hierarchy is too complex for most of the applications. Therefore, we do not need completely define all the joints in our application. However, it should be noted that the data format stated above can be used for representing the full skeleton specified in the H-Anim 1.1 Specification. A front and left


```

<Joint name="HumanoidRoot" center="0.1296 0.0000 -3.8563">
  <Joint name="sacroiliac" center="0.0075 0.0000 -4.2920">
    <Joint name="l_hip" center="-0.0005 -0.4446 -4.2920"
      limitx="-1.00 2.00" limity="-1.40 2.90" limitz="-1.40 1.30">
      <Joint name="l_knee" center="0.1441 -0.4446 -2.2778"
        limity="-3.00 0.00">
        <Joint name="l_ankle" center="-0.3444 -0.4446
          -0.3070">
          <Joint name="l_midtarsal" center="0.1722 -0.4446
            -0.0005" />
        </Joint>
      </Joint>
    </Joint>
  <Joint name="r_hip" center="0.0136 0.4446 -4.2920"
    limitx="-1.00 2.00" limity="-1.40 2.90" limitz="-1.40 1.30">
    <Joint name="r_knee" center="0.1488 0.4446 -2.2993"
      limity="-3.00 0.00">
      <Joint name="r_ankle" center="-0.3585 0.4446
        -0.3332">
        <Joint name="r_midtarsal" center="0.1722 0.4446
          0.0000" />
      </Joint>
    </Joint>
  </Joint>
</Joint>

```

Figure 3.2: A portion of humanoid data in XML format.

view of the skeleton corresponding to the XML based model in Appendix A.2 are shown in Figure 3.3.

3.2 Animation of the Skeleton Layer

3.2.1 Motion Control

The system designed by [30] aims to establish a method for the low-level motion control using inverse kinematic methods. Positions, joint angles, forces, torques and other motion parameters are specified manually. *Spline-driven animation techniques* are chosen to specify the characteristics of the motion. The paths for



Figure 3.3: The front and left view of the skeleton

pelvis, ankle and wrist motions are specified by Cardinal splines. Cardinal splines are a class of cubic splines that are very powerful and have easily controllable input control points fitting mechanism with a tension parameter. Cubic splines are mostly preferred due to less computation cost and low memory requirement. More detailed information about splines and the description of a high-level walking animation mechanism that makes use of these low-level techniques can be found in [30].

In addition to position curves generated for pelvis, ankle and wrist, there exists a velocity curve specified independently for each body part. Thus, characteristics of the motion can be changed just by modifying the velocity curve. This method is called as *double interpolant* method [48].

However, arclength problem arises when moving an object along a given position spline due to the parametric nature of cubic splines. The arclength problem appears when we have a path specified by a spline and we are looking for a set of points along the spline such that the distance travelled along the curve between consecutive frames is constant. The *arclength parameterization* of spline curve is a remedy to this problem.

Spline curves provide useful functionality for motion control. They can be constructed with a few set of control points and can be manipulated easily. Figure 3.4 indicates an example to these motion control issues. Following this approach, the system provides a low-level animation system to the user in which one can obtain any kind of motion by specifying a set of spline curves for position, distance and joint angles over time.

Furthermore, a high-level motion mechanism for walking that uses these low-level techniques is provided. In the system, high level control of walking is achieved by allowing the user to specify a few number of locomotion parameters. Specifying the straight travelling path on flat ground without obstacles, and the speed of locomotion, the system generates the walking automatically by computing the 3D path information and the low-level kinematics parameters of the body elements. Furthermore, some extra parameters such as the size of the walking step, the time elapsed during double-support, rotation, tilt, and lateral displacement of pelvis can be adjusted by the user to have a different walking motion. The characteristics of the walking motion of the system is explained in detail in [30].

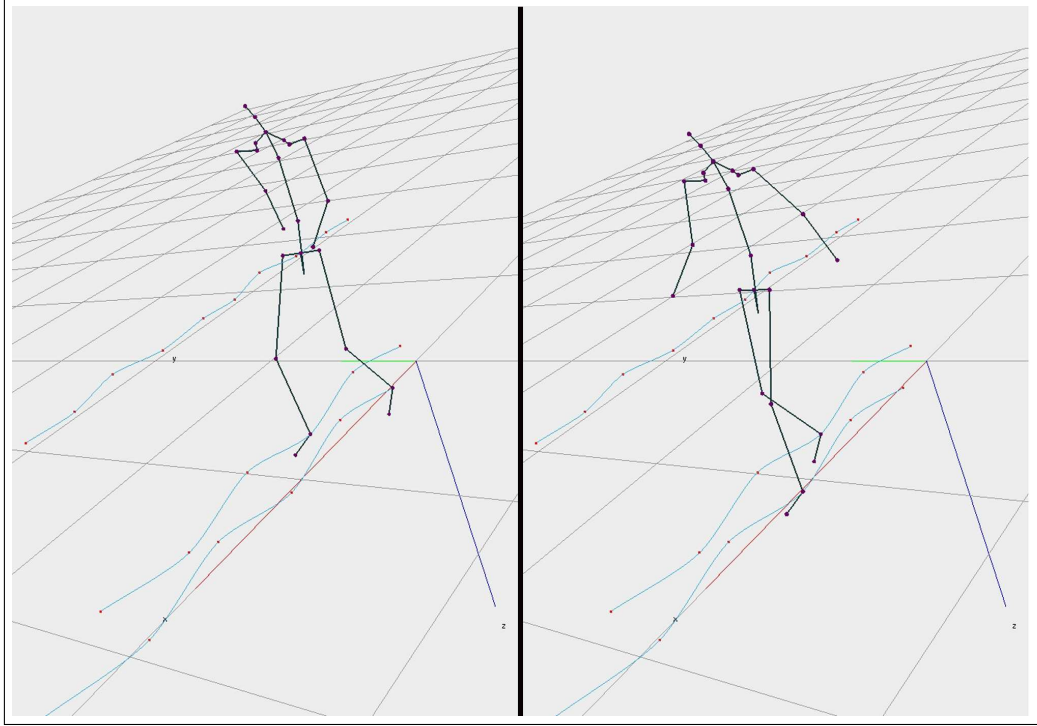


Figure 3.4: The articulated figure and the Cardinal spline curves

3.2.2 Motion of the Skeleton

In this study, *kinematics* methods are used for the motion of the skeleton. Kinematics methods involve the study of motion specification independent of underlying forces that produced the motion. *Forward* and *inverse kinematics* are two different approaches for the animation of a figure. In forward kinematics, the animator explicitly defines the position, rotation angles, and etc. whereas in inverse kinematic methods, the goal positions are specified and the position and rotation of joints are computed accordingly. Moving a hand to grab an object or placing the foot to a desired position requires the usage of inverse kinematics methods. Thus, our system uses inverse kinematics for motion control.

Memişoğlu [30] uses an inverse kinematics package named IKAN software, which was developed at the University of Pennsylvania, to implement his design

[10]. IKAN is a complete set of inverse kinematics algorithms for an anthropomorphic arm or leg. IKAN provides functionality to control the arm and leg of my human model. For the arm, with the goal of putting the wrist in the desired location, IKAN computed the joint angles for the shoulder and elbow. In the case of leg, rotation angles for hip and knee is calculated. The details of how IKAN is incorporated with our implementation is explained in [30].

Chapter 4

MODELING AND ANIMATION OF THE MUSCLES

4.1 Modeling of the Muscle Layer

While the skeleton creates the general structure of the body, muscles determine the general shape of the surface mesh. As a biological fact, human body muscles account for nearly half of the total mass of the body and fill the gap between the skeleton and the skin [38].

Human body movements require the muscles to perform different tasks, that is why our body includes three types of muscles; cardiac, smooth and skeletal muscles [28]. *Cardiac muscles*, which are used only in the heart, perform the pumping of the blood throughout the body. The second class of muscles, *smooth muscles*, is a part of the internal organs and found in the stomach, bladder, and blood vessels. Both of these two muscles are called involuntary muscles, because they cannot be consciously controlled. On the other hand, *Skeletal muscles* put

into practice the voluntary movements. They are attached to bones by tendons and provide us the capability to perform various actions by simply contracting and pulling the bones they are attached to towards each other. In this study, only skeletal muscles are modeled because we aimed to represent only the external appearance of the human body.

Skeletal muscles are located on top of the bones and other muscles, and they are structured side by side and in layers. There are approximately 600 skeletal muscles in the human body and they make up 40% to 45% of the total body weight. A typical skeletal muscle is an elastic, contractile material that originates at fixed origin locations on one or more bones and inserts on fixed insertion locations on one or more other bones [38]. The relative positions of these origins and insertions determine the diameter and shape of the muscle. In a real human, muscle contraction causes joint motion but in our articulated figure, muscles deform due to the joint motion, in order to result realistic skin deformations during animation.

There are two types of contraction: isometric (same length) and isotonic (same tonicity) contraction. In *isotonic contraction*, the muscle belly changes shape that causes the total length of the muscle shortens. As a result, the bones to which the muscle is attached are pulled towards each other. On the other hand, due to *isometric contraction*, the changes in the shape of the muscle belly because of the tension in the muscle does not change the length of the muscle, so no skeletal motion is produced [1]. Although most body movements requires both isometric and isotonic contraction, in our application only isotonic contraction is considered. Because, isometric contractions do not have much influence on the appearance of the body model during the animation.

In most muscle models, a muscle is represented using two levels: the action line and the muscle shape. However, we represented the muscle layer only with action lines and we did not consider the muscle shape as seen in Figure 4.1.

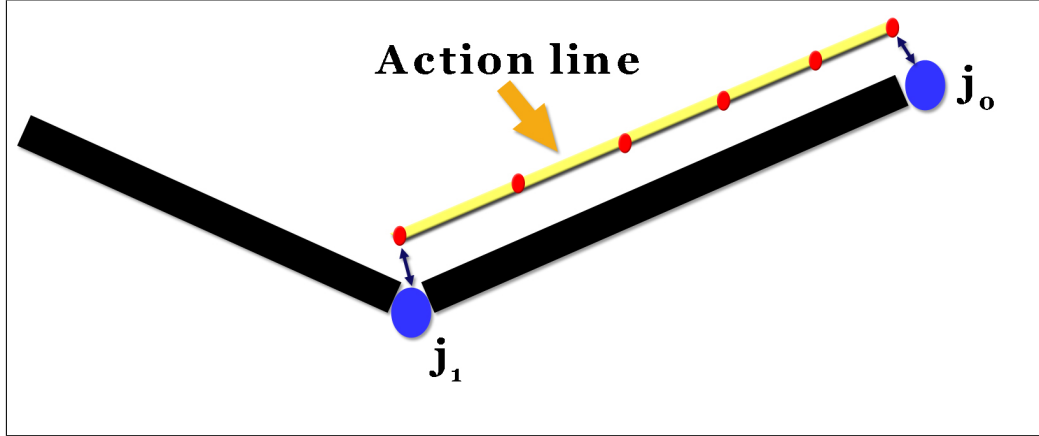


Figure 4.1: Action line abstraction of a muscle

We modeled muscles only in the upper arm, upper leg and lower leg parts of the human body. The key idea of our approach is that the deformations of the skin mesh are driven by only the underlying action lines and no muscle shape is needed. This muscle structure will be explained in the next section.

4.1.1 Action Lines

An *action line* denotes the imaginary line along which the force applied onto the bone is produced. Nevertheless, the definition of this line is not so clear [1]. Many specialists assume the action line as a straight line, but the most common definition receives the action line as a series of line segments (or polyline in computer graphics terminology) [11]. These segments and their number are determined through the anatomy of the muscle. As a matter of fact, action line is the representative of the muscle force at a cross-section.

In this study, muscles are represented by an action line that simulates the muscle forces and is basically defined by an origin and an insertion point. However, as it is stated above, we have decided to represent the actions of the muscles by using polylines, as described in [11]. For this purpose, we used some control points, which are responsible for guiding the line and including the forces exerted

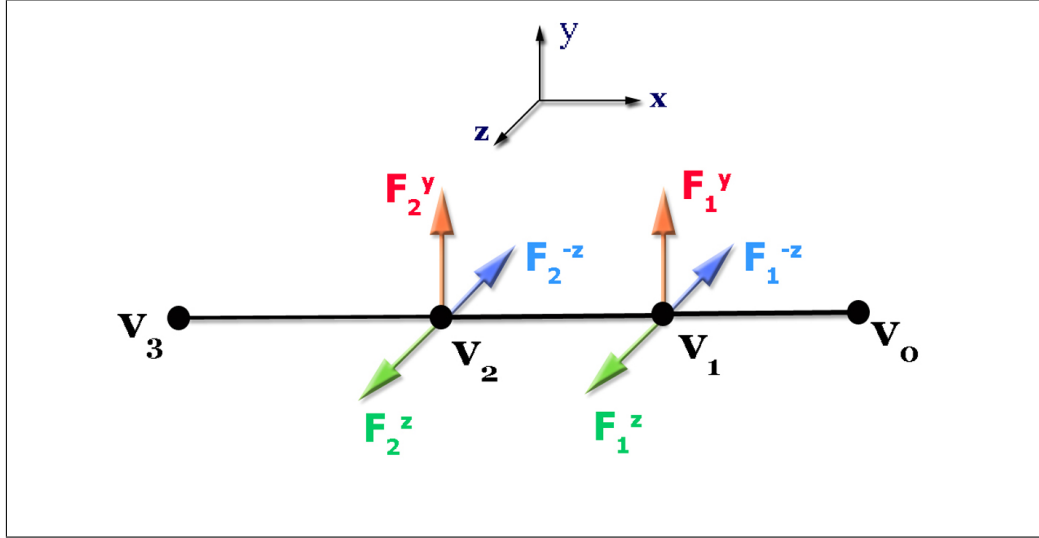


Figure 4.2: The structure of an action line: control points and forces on these points

onto the skin mesh. These force fields are inversely proportional to the length of the corresponding action line segment. An example of this kind of action line is shown in Figure 4.2.

4.2 Animation of the Muscle Layer

Animation of the muscles is a more difficult task than the modeling. It is a very complicated process to derive the position and deformation of a muscle in any possible frame. As in the most graphics applications, we used the method that the deformations of the muscles are inferred from the the motion of the skeleton, which is contrary to the real life.

In our approach, the deformations of the skin layer are driven by the underlying bones and action lines. This makes the three-dimensional nature of the deformation problem to be reduced to one dimension. As it is stated, each action line is modeled by a polyline and these polylines include some number of control points. The control points that correspond to the insertion and origin of the

muscle are attached to the skeleton joints so that their motion is dictated by the skeleton. The positions of all the remaining control points are obtained through a linear interpolation formulation for each animation frame.

First of all, since the positions of the control points of the action line provide information as to how the surface mesh will expand or shrink over time, we need to determine the local frame of each action line control point.

After the local frames are constructed, action line becomes ready to animate in correspondence with the underlying skeleton. Since the insertion and origin points of the action line are fixed on the skeleton, when the skeleton layer moves, this movement is reflected on the action line as decrease or increase in length. Parallel to this change in action line length, the lengths of each action line segment also change. Since the force fields on each control point are inversely proportional with the segment length, this variation in length also causes a change in the force fields as demonstrated in Figure 4.3. The next step in animation is the deformation in the skin mesh due to the changes on the forces that are exerted on skin vertices by the action line control points. This deformation is automatically propagated on the skin layer via the anchors between skin vertices and action line. Detailed information about these anchors are presented in Subsection 5.1.1. If the segment length shortens, the force fields increase and make the skin mesh to bump. Similarly, the elongation in segment length results in decrease in force fields and relaxation of skin mesh.

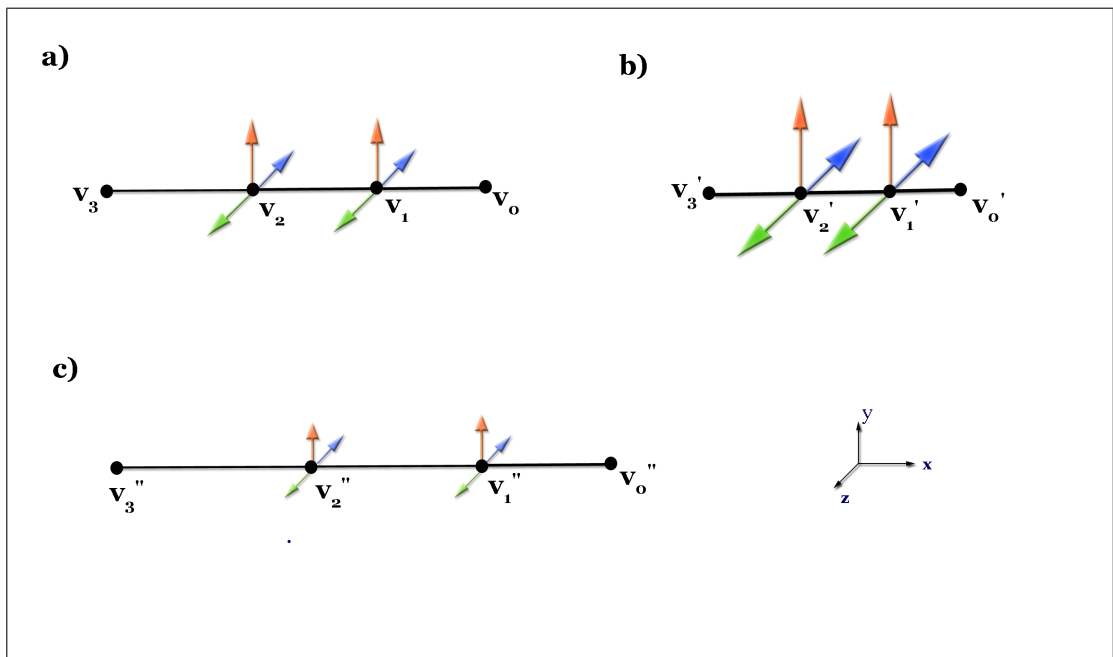


Figure 4.3: The deformation of an action line and force field changes: a) rest position and initial force fields of an action line, b) action line shortens and forces increases due to muscle contraction, and c) action line lengthens and forces decreases due to muscle elongation

Chapter 5

MODELING AND ANIMATION OF THE SKIN

5.1 Modeling of the Skin Layer

The *skin* is a continuous external sheet that covers the body. The skin accounts for about 16% of the body weight and has surface area from 1.5 to 2.0 m^2 in adults [25]. Its thickness varies depending on the location.

In computer graphics, there are basically three ways to model a skin layer. Designing from scratch or modifying an existing mesh in a 3D modeler is the first method. Another way is to laser scan a real person that produces a dense mesh that truly represents a human figure. The last method for modeling the skin layer is extracting it from underlying components if exist.

Our skin model is modeled in 3D modeler, Poser [32]. We used P3 Nude Man model, which is one of the standard characters of the Poser software. The shaded points and the solid view of the model are seen in Figure 5.1. The model contains 17,953 vertices and 33,234 faces. The whole body is composed of 53

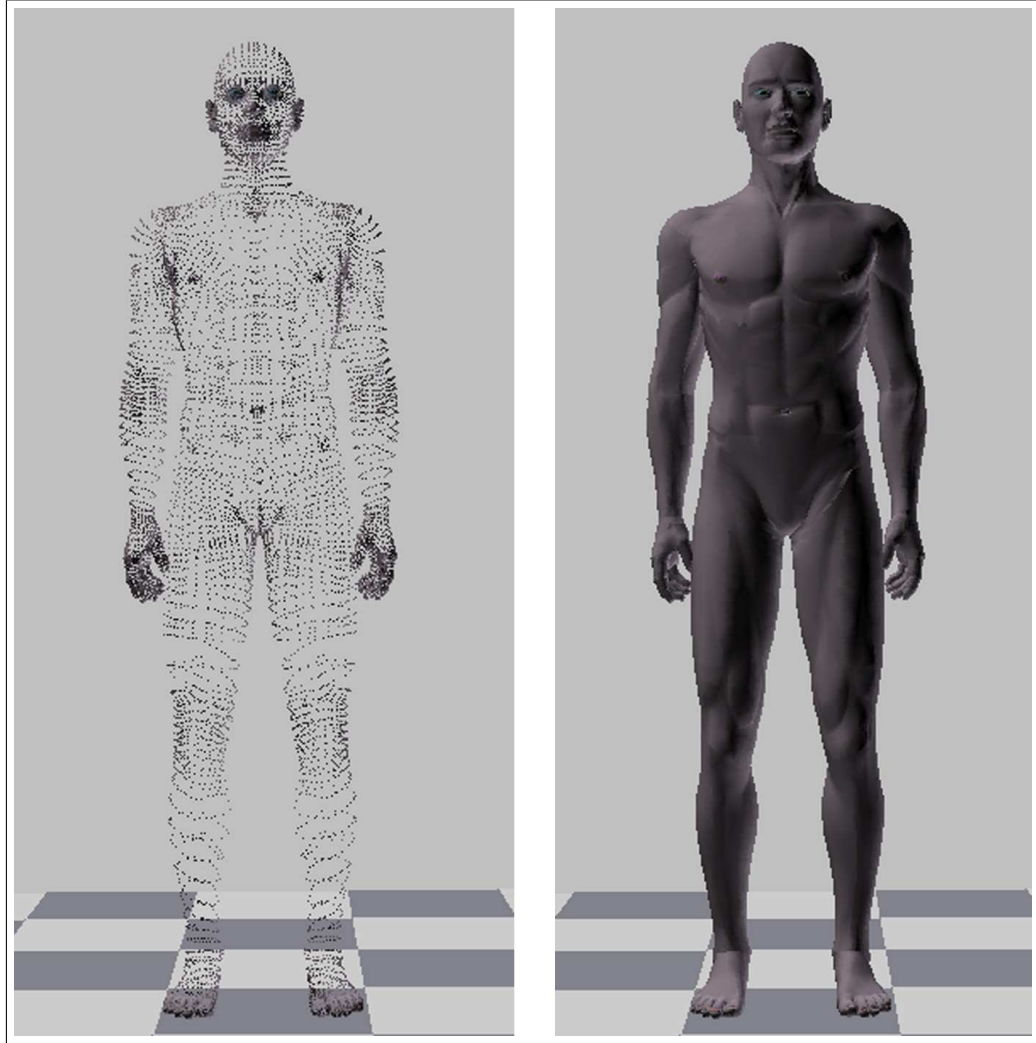


Figure 5.1: The shaded point and solid view of the skin model

parts like hip, abdomen, head, right leg, left hand, etc. This structure provides us the required functionality for binding vertices to the inner layers.

5.1.1 Skin Attaching

After surface creation, in a second stage called *attaching*, each vertex in the skin is associated with the closest underlying body components (muscle and bone). Basically, the attachment of a particular skin vertex is the nearest point on its underlying component. This provides the functionality that the shape

changes in the underlying component are propagated through these anchors to the corresponding skin vertices. The attaching procedure will be examined in more detail in the sequel.

In our system, skin vertices are first bound with the joints of the skeleton layer in a multi-step algorithm. Basically, to attach the skin to the joints, skin vertices are transformed into the joint coordinate system. As it is stated in Section 5.1, the skin model is decomposed into parts. *Parts* are groups of vertices of the skin mesh correspond to some part of the body. In the first step of the attaching algorithm, for each part of the skin, we determined a particular joint and attached the vertices of the object to this joint, i.e., right arm object is bound with right shoulder joint and left thigh object is anchored to left hip joint as seen in Figure 5.2. The mathematical details of this and proceeding attachment procedures are explained in Section 5.2. This binding procedure is a virtual process and dictates the position of the body parts when the underlying joint orientation changes.

However, the first step is not sufficient for realistic deformation of the skin layer. In order to perform full skinning method, it is required for some vertices to be bound with more than one joint. Especially, the vertices near the joints are need to be attached to two adjacent joints. The second step of the attaching algorithm is focused on binding the required vertices with two joints. For this purpose, we determine a distance value as a threshold for the joints; left ankle, right ankle, left knee and right knee (see Figure 5.3). For the vertices of each part that are closer to these joints, if the distance between the vertex and the joint is smaller than the threshold value then this vertex is bound with the joint with a weight although the vertex is belonging to another object. The weight of the joints are determined with inversely proportional to the distance between the vertex and the joint.

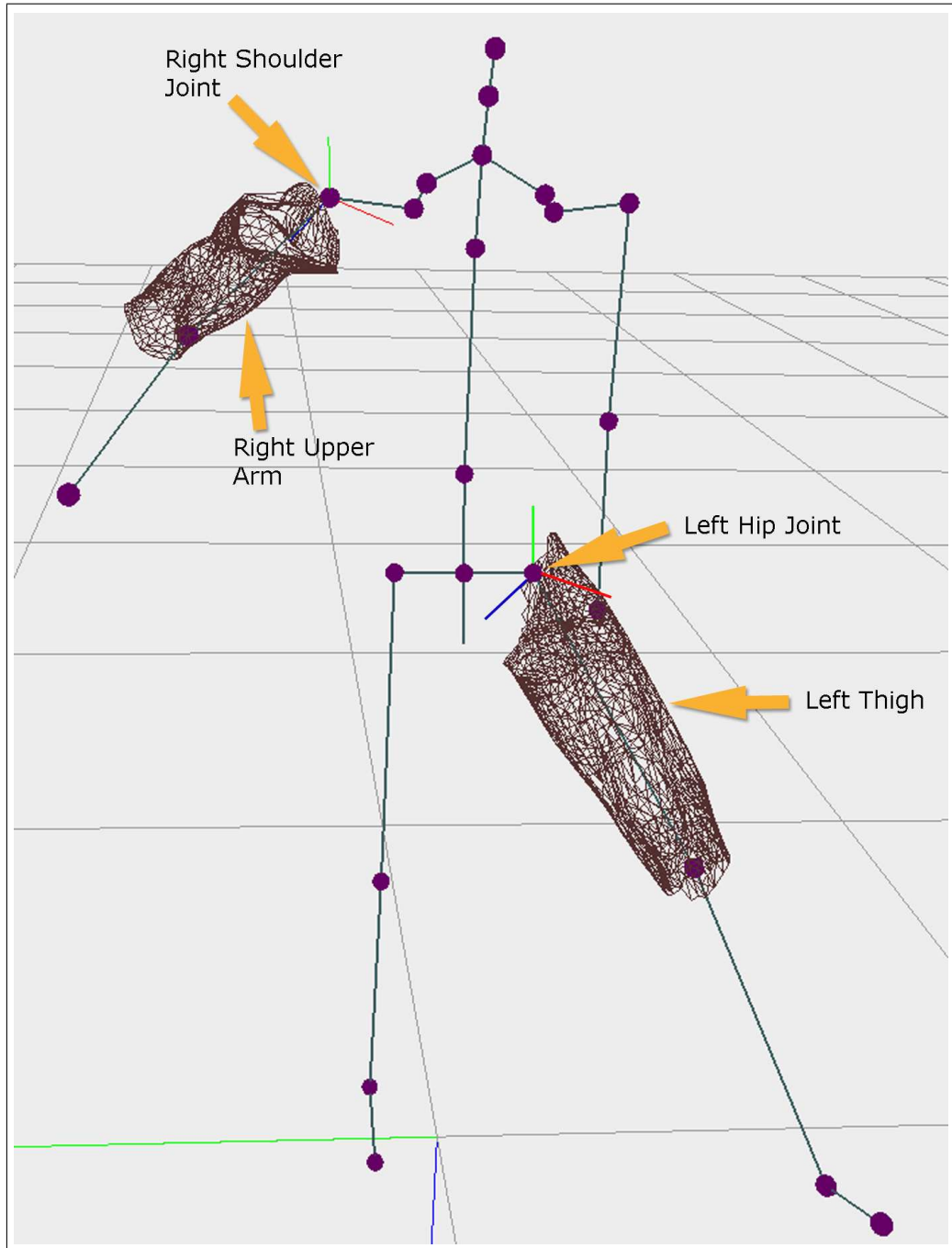


Figure 5.2: First step of the attaching algorithm: the right upper arm part is bound with the right shoulder joint and the left thigh part is bound with the left hip joint.

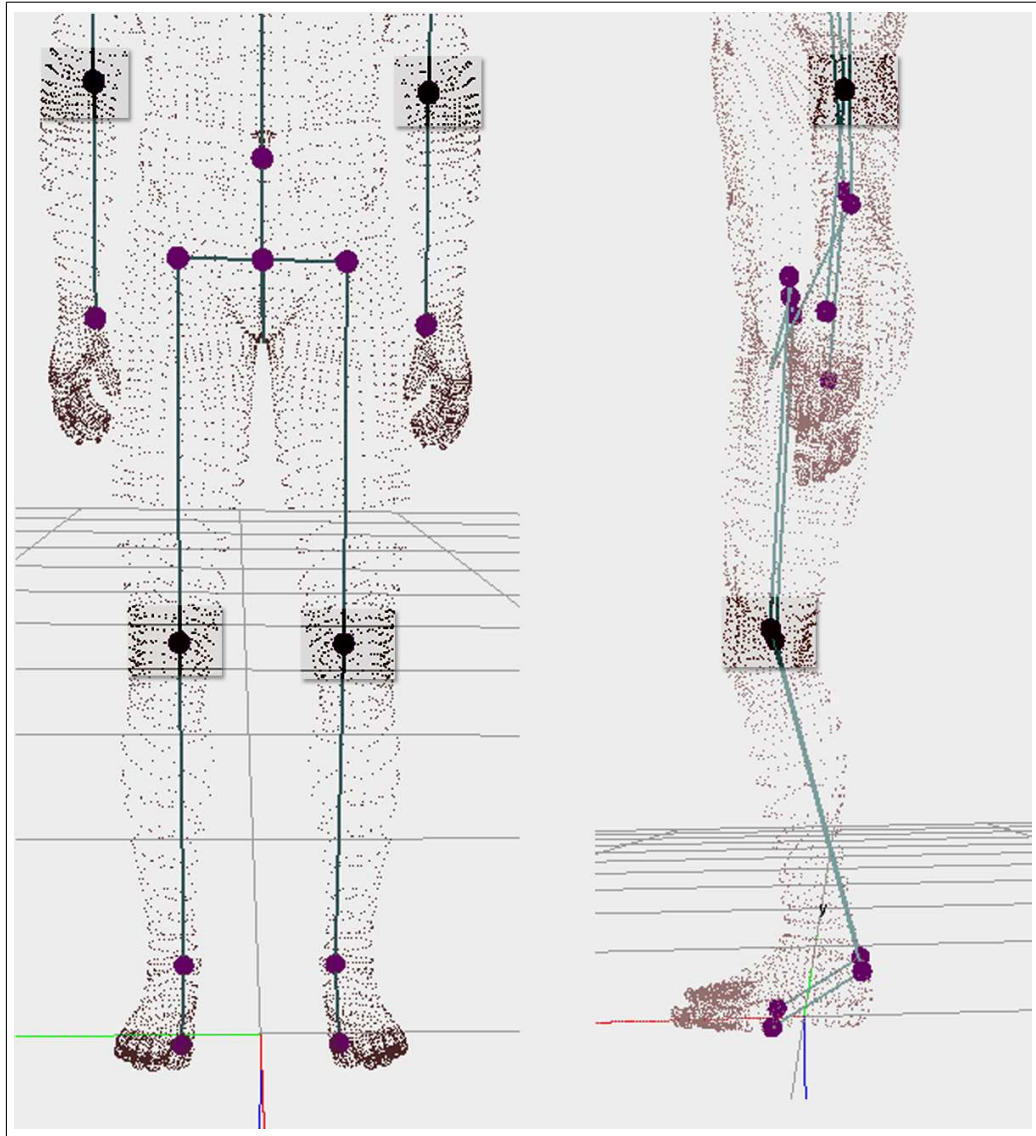


Figure 5.3: Second step of the attaching algorithm: the front and left view of some critical joints and their influence areas. The darkened rectangles denotes the vertices that are attached with the corresponding joint.

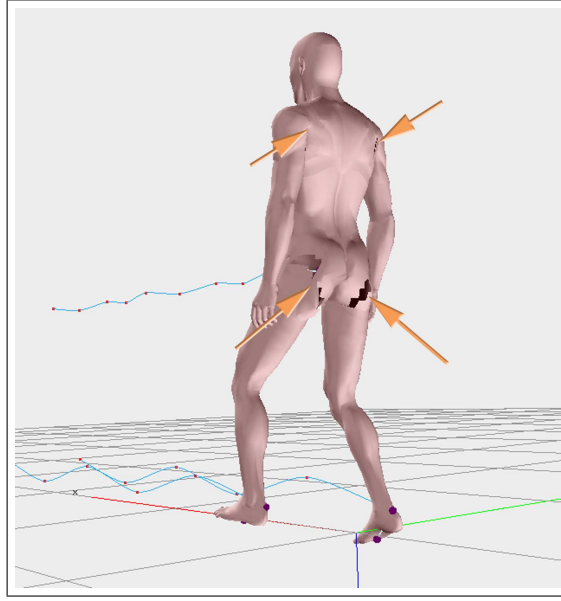


Figure 5.4: The holes at the backside of the model while walking due to unattached vertices

The second step of the algorithm also has some problems. We aimed to bind the near vertices to the joints but we could not determine the appropriate distance value for the binding operation. If we use a smaller threshold value, some of the vertices that should be attached are not attached with the joints. For example, some vertices of the back part of the hip need to be attached to the left hip or right hip joints in order to generate a realistic walking motion. However, since the distance of these vertices to the corresponding joints are bigger than the threshold value, undesired renderings were generated during animation (see Figure 5.4). The unattached vertices caused holes at the backside of the model, also some in the shoulders. As a solution we tried to increase the threshold value. Unfortunately, this attempt caused unnatural results for the other parts of the body during movement.

Therefore, contrary to the first two steps, which are fully automatic, we implemented a manual step to overcome the deficiencies. This last step includes selecting some of the unattached vertices manually by using mouse and then binding them with the appropriate joints. For this purpose, we implemented

the capability for picking skin vertices on the screen by clicking on them with mouse. When all the necessary vertices are selected, they are attached to the related joints.

Anchoring points to action lines of muscles can be achieved by a similar process, which is described next. An action line can be considered as a skeleton; control points denote joints and line segments correspond to bones. This allows us to reuse algorithms developed for mapping skin vertices to skeleton joints with some extensions. In the skin-skeleton mapping algorithm, each point of an object are bound with a particular joint. In skin-action line mapping algorithm, each vertices of an object are again attached to a particular action line, but since an action line is composed of a set of control points and each control point has a different force field on the skin mesh (see Figure 4.2), we have to bind each vertex to a number of control points of the action line. As it is seen from Figure 5.5, we decided to select the nearest three action line control points of a vertex. Then, the vertex is influenced by the sum of the effects of these three points. The details of the effects of control points on skin vertices are given in Subsection 5.2.2.

5.2 Animation of the Skin Layer

Due to the deformation of the underlying skeletal and muscular layers, we deform the skin mesh according to the following steps: vertex positioning on the surface due to skeletal movement and displacement along the normal to the surface so as to simulate the action of the muscular system. The first step simulates the adherence of the skin to the inner layers. In the second stage, the skin is shaped by the underlying muscles.

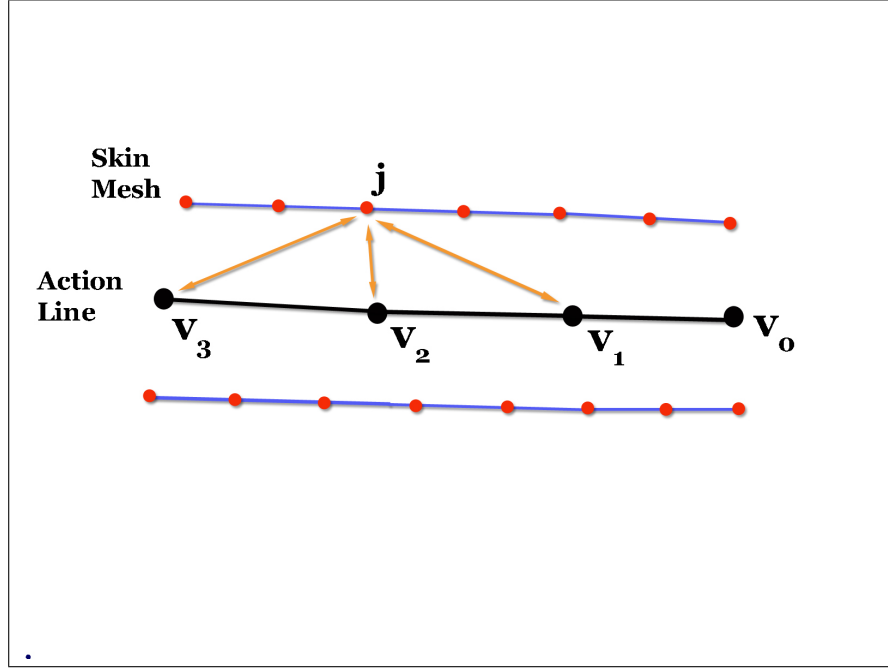


Figure 5.5: Binding skin vertices with action lines. Point j is attached to its nearest three control points, v_1 , v_2 and v_3 .

5.2.1 Skeletal Deformation

Before applying muscle-induced deformations to the skin layer, we displace each skin vertex based on the skeletal position. In this step we aimed two goals: first generating a smooth skin appearance and second, simulating the adherence of the skin layer to the skeleton layer. We now introduce the skinning based surface deformation model, which is based on [20]

Skeleton and skin layers are designed in some reference position. Here, skeletal animation aimed to move only the skeleton and propagate this movement to the skin. For this purpose, assume that the parent of joint j is denoted by $p(j)$, the root joint has index zero, and $R(j)$ is the 4×4 matrix describing the transformation from joint $p(j)$ to joint j . Then, the transformation from the world coordinate system to the local frame of joint j is as follows:

$$\mathbf{A}(j) = \mathbf{R}(0) \dots \mathbf{R}(p(j)) \mathbf{R}(j) \quad (5.1)$$

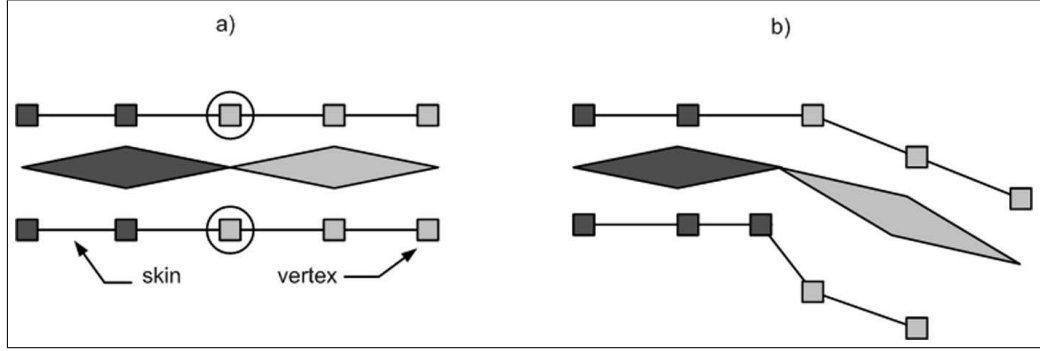


Figure 5.6: The skeletal animation: a) reference position and b) after rotation

where $R(j)$ denotes the coordinate transformation from $p(j)$ to j .

Besides, we use transformation matrix $T(j)$ that expresses the rotation of the bone starting at joint j to move a character. So, the final transformation of the joint with the movement applied to it and all its parents is expressed by:

$$\mathbf{F}(j) = \mathbf{F}(p(j))\mathbf{R}(j)\mathbf{T}(j) \quad (5.2)$$

Now the actual posture of the vertex v can be determined when it is transformed to vertex v' by Equation 5.3, assuming that v is attached to joint j .

$$\mathbf{v}' = \mathbf{F}(j)\mathbf{A}(j)^{-1}\mathbf{v} \quad (5.3)$$

However, the basic skeletal animation method does not generate a realistic shape near the joints due to sudden changes in attached bones as seen in Figure 5.6. Since we aimed to model smooth skin deformation using a segmented skeleton, we used the skinning method as we explained briefly in Subsection 5.1.1. To be more precise, allowing attachment of one vertex to more than one bone and averaging the results produces smooth skin appearance near the joints. In this scheme, if vertex v is attached to joints j_1, \dots, j_n then the formula for the resulting vertex position is:

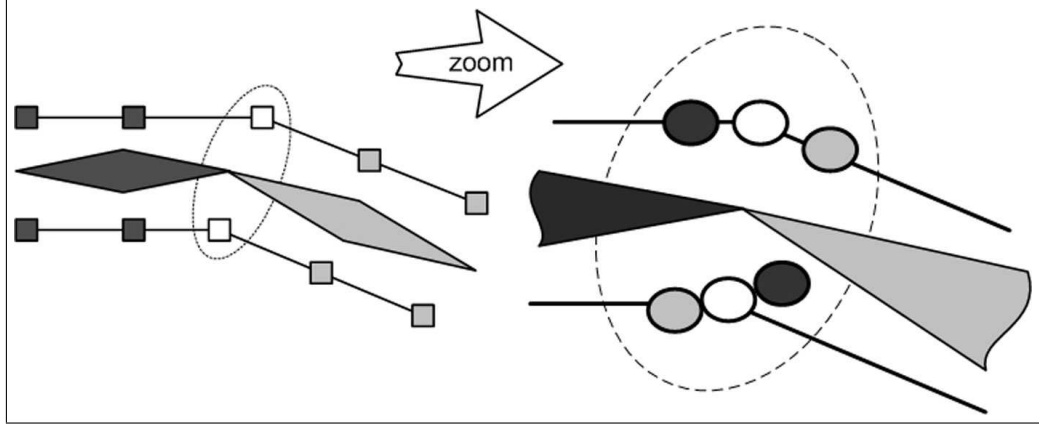


Figure 5.7: The skinning algorithm

$$\mathbf{v}' = \sum_{i=0}^n w(i) \mathbf{F}(j_i) \mathbf{A}(j_i)^{-1} \mathbf{v} \quad (5.4)$$

where $w(1), \dots, w(n)$ are weights of the attachment to particular joint where total weight is equal to 1 and n is 2 for our case. Skin deformation with this technique is illustrated in Figure 5.7. In 5.7, the vertices that are near the joints were assigned to both bones with equal weights 0.5 and 0.5. The white vertices represent the results of averaging.

In our system, we implemented a more complicated skinning algorithm. The vertices that are closer to a joint more than a threshold value are attached to two joints and deformed by the movements of both bones. The process is demonstrated in Figure 5.8.

The interpretation of Figure 5.8 is that, first, we select the vertices that are in some neighborhood of a particular joint. These vertices can be either in one of the adjacent parts of the joint. If we can not select all the necessary vertices automatically by means of distance measurement, we select the remaining vertices manually by picking up vertices and binding them to the corresponding joint interactively. Secondly, a linear weighting algorithm, which is inversely proportional to the distance between the vertex and the joint, is applied to the selected

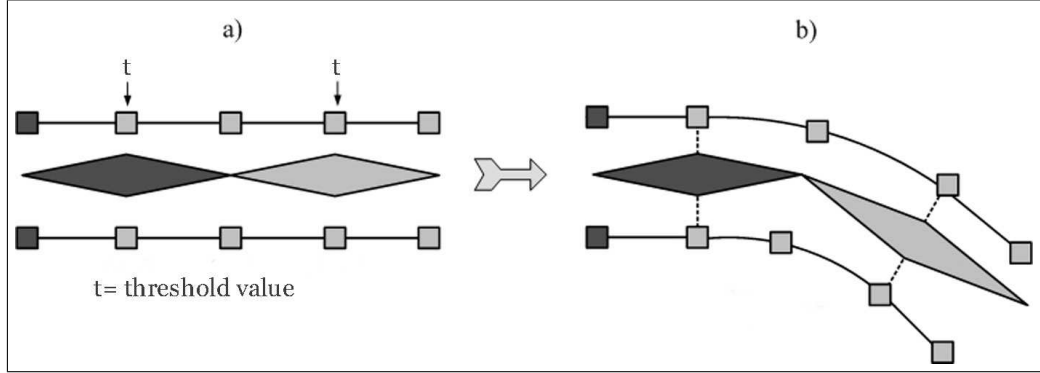


Figure 5.8: The skinning method used in our implementation: a) vertex assignment and b) after rotation

vertices. As a result, these selected vertices are influenced by both of the nearby joints and the parent of these joints.

5.2.2 Muscular Deformation

The muscular deformation is based on the control points of the action lines. Due to the skeletal movements, the length of the action line and the lengths of each segment of the action line change. There exists an inverse proportionality between the length of the segment and the force fields of the control points. The forces that are exerted on the skin mesh by the action line may increase or decrease, due to length changes. These changes are propagated through the skin layer via the anchors between the skin vertices and action line control points. As a result of this propagation, the skin mesh is deformed in order to create a realistic appearance during animation. The visual results of the muscular deformation are presented in Section 6.1.

Chapter 6

EXPERIMENTAL RESULTS

6.1 Visual Results

The front and back views of the walking behavior produced by our system is illustrated in Figures 6.1 and 6.2.

The visual results in Figures 6.1 and 6.2 show that our system is capable of generating realistic rendering of the human walking behavior. There is no holes or wrinkles on the skin mesh, even near the highly moveable joints like shoulders, hips and knees.

The postures of the arm of the human model generated in a multi-dimensional motion are presented in Figure 6.3. The deformation of the skin mesh can be seen more clear in these figures.

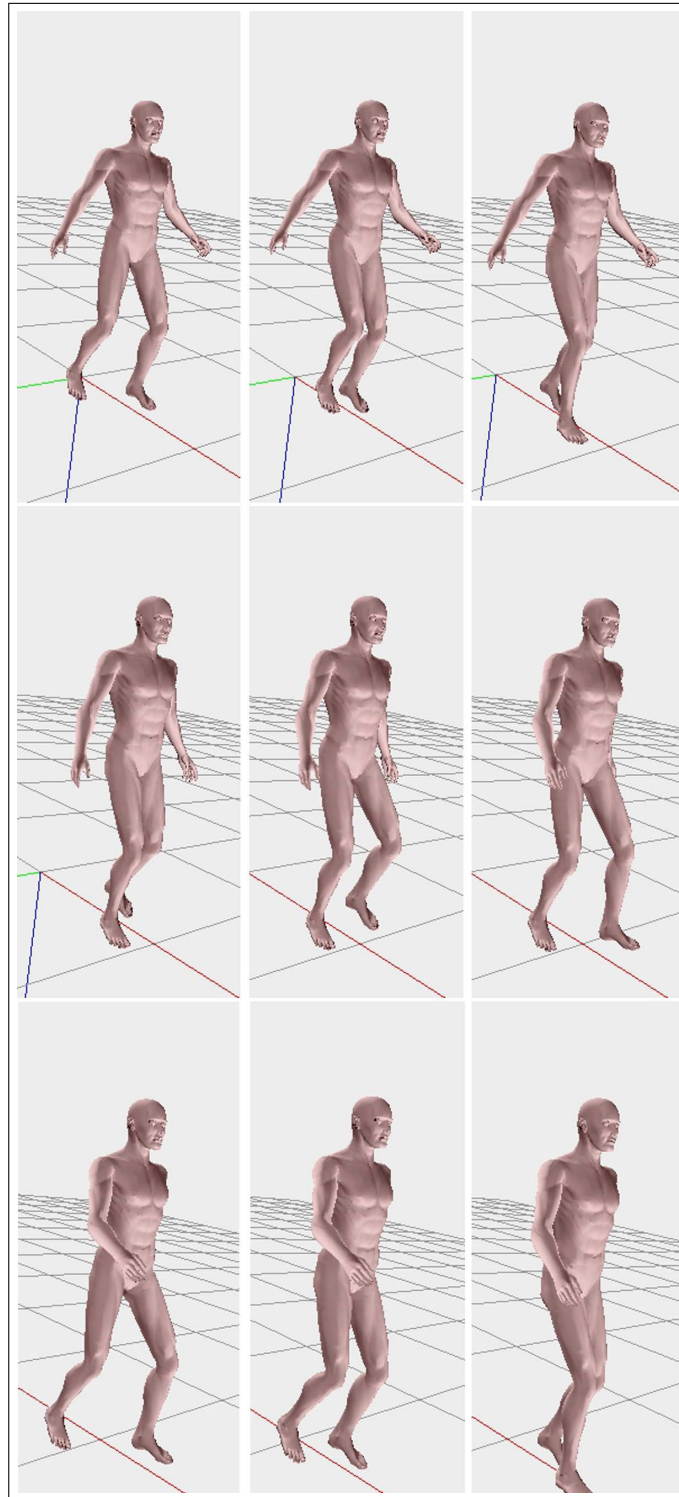


Figure 6.1: The front view of the human walking behavior.

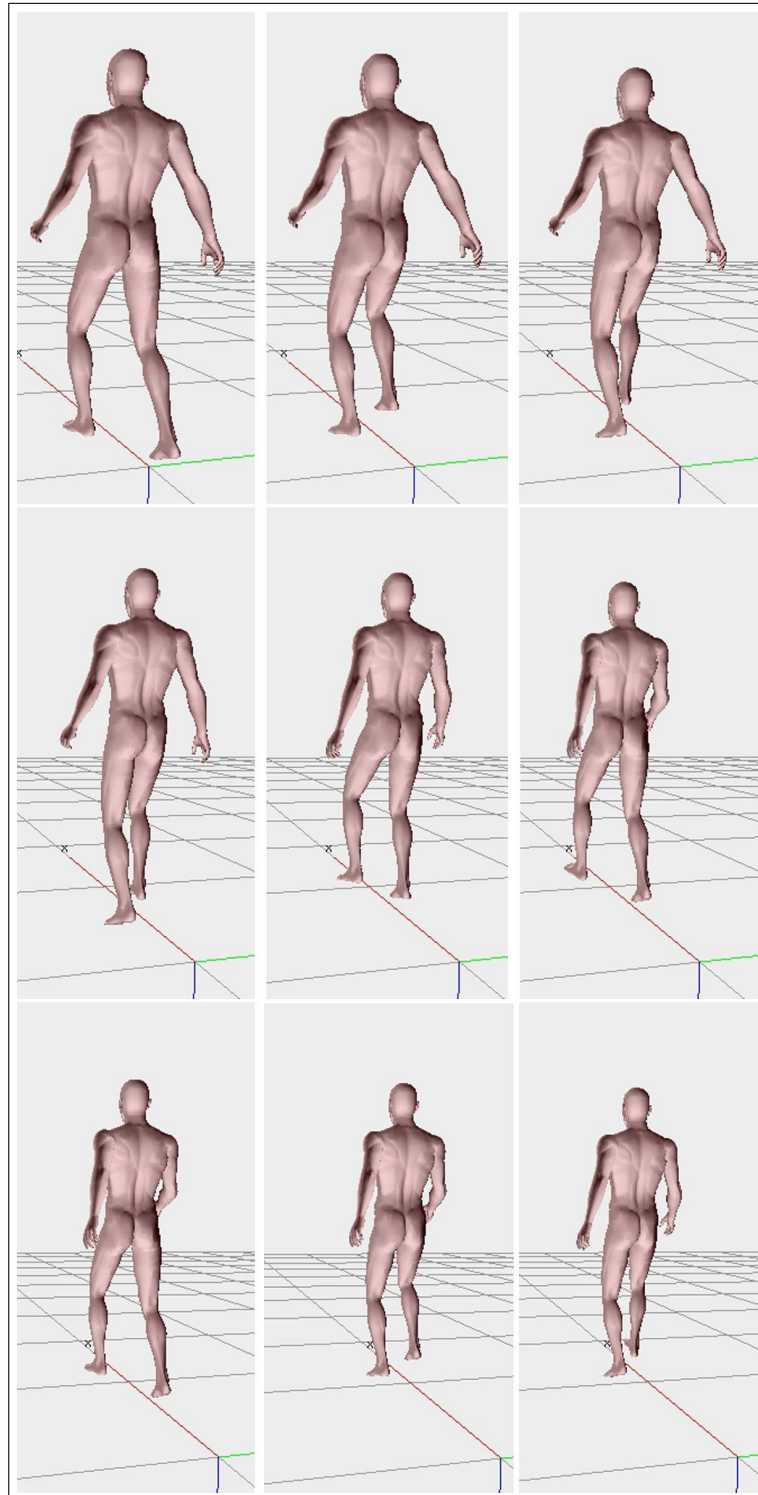


Figure 6.2: The back view of the human walking behavior.

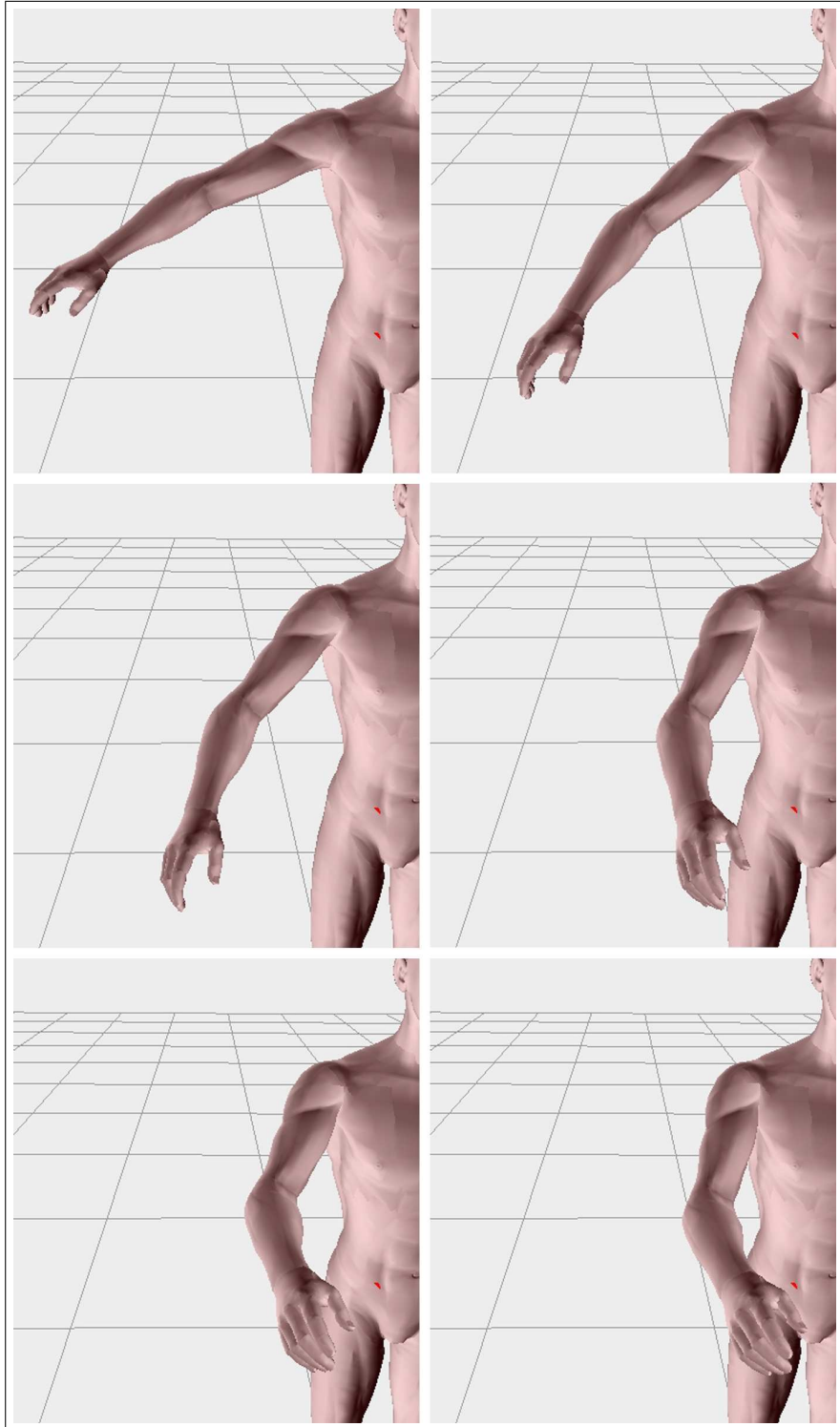


Figure 6.3: The multi-dimensional motion of the arm of the human model.

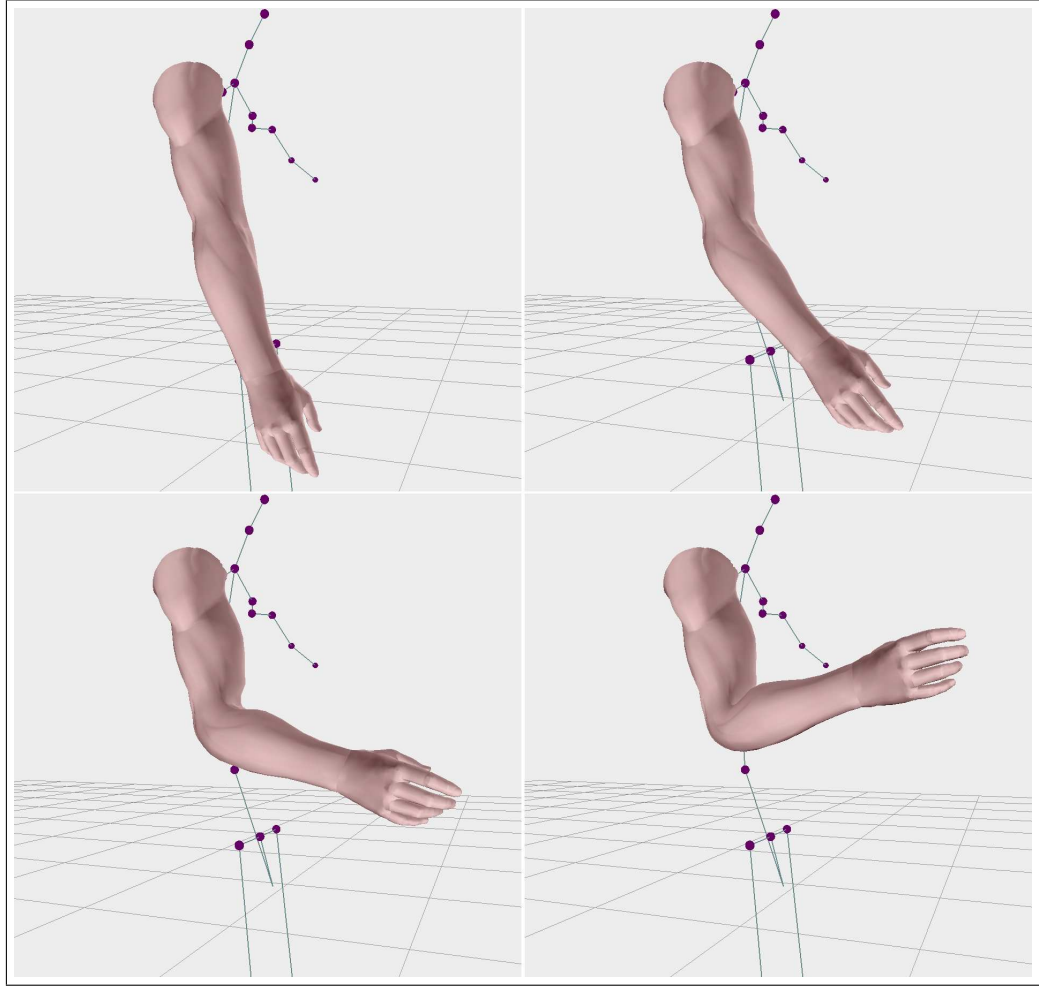


Figure 6.4: Showing muscular deformation on the skin mesh.

In order to demonstrate muscular and skeletal deformations, we show a series of stills taken from animations rendered in our system. In the animation seen in the Figure 6.4, the right arm of the character is raised forward and then flexed in order to demonstrate muscular deformation on the skin mesh.

6.2 Performance Analysis

We implemented the proposed system in a PC platform with Pentium IV 2.4 GHz CPU, 512 MB Memory and 128 MB GeForce4 Ti 4200 graphics card. The software platform was Microsoft Visual C++ 6.0 with OpenGL libraries. Original

	# of Vertices	# of Triangles	# of Vertices Bound with One Joint	# of Vertices Bound with Two Joint	Frame Rate (fps)
High Resolution	17,953	33,234	15,305	2,648	19
Middle Resolution	8,923	15,203	7,623	1,270	48
Low Resolution	5,133	7,625	4,189	944	92

Table 6.1: Performance evaluation results

human model is composed of a skin mesh that contains 17,953 vertices and 33,234 faces, a muscle layer, and a skeleton structure with 25 joints and 23 bones. Since bones are simulated with lines, they do not have an significant effect on total performance. The most tedious and time consuming operation is the binding process of the skin layer with the underlying layers. However, this does not effect the frame rate since it is done as a preprocessing step before the animations. We tested the system with different skin meshes ranging from low to high resolutions (see Figure 6.5). The average frame rates given in terms of frames per second (fps) for different meshes are given in Table 6.1.

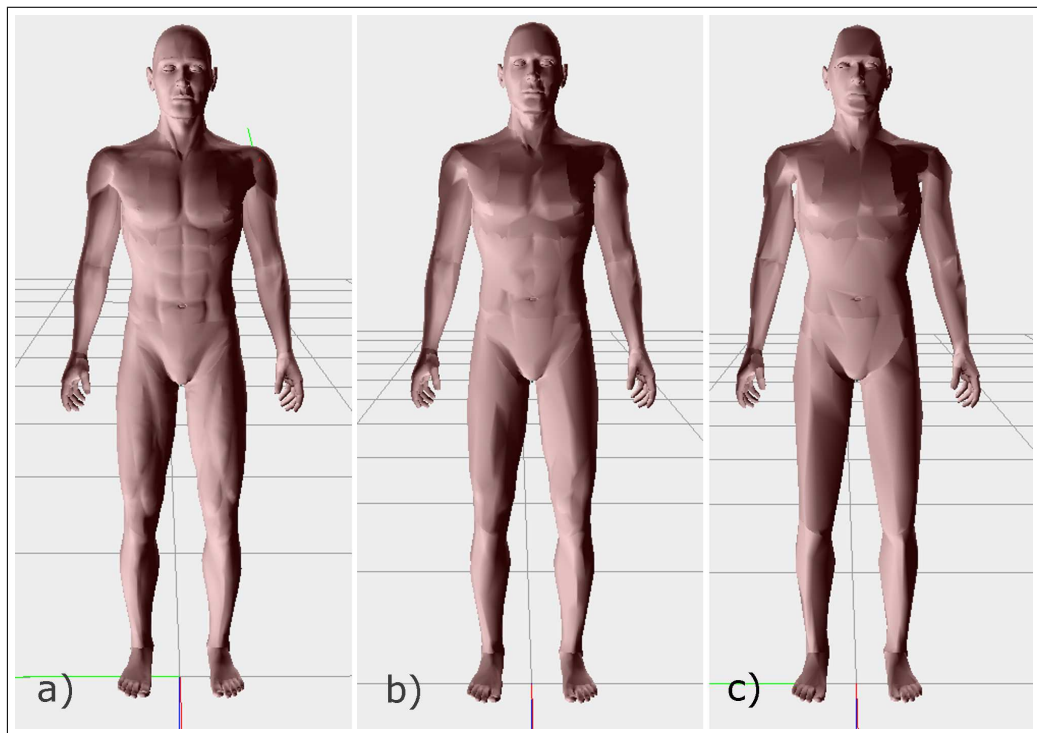


Figure 6.5: Different resolution valued skin meshes: a) high resolution (original model), b) middle resolution, and c) low resolution.

Chapter 7

CONCLUSION

In this study, we proposed an animation system for a human body model. The system is based on an anatomically based approach that builds layers of the human body: the skeleton layer, the muscle layer and the skin layer. The skeleton layer consist of an articulated figure that is modeled based on the H-Anim 1.1 Specification. By using the skeleton layer, we control the motion of the human body, and with the muscle layer, we can deform the skin layer according to the motion. We introduced a simple muscle model that approximates the human anatomy. The deformation of the skin is mostly driven by the muscle action lines. Our skin layer is modeled in an 3D modeler. The vertices of the skin mesh is attached with the inner layers in such a way that, the attachment of a particular skin vertex is the nearest point on its underlying component. This provides the functionality that if the shape of the underlying component changes, these changes are propagated through these anchors to the skin layer. Performance evaluations showed that the multi-layered model produces realistic skin deformation in real-time.

We suggest a few directions for future research. We did not deal with the problem of collision detection and response between different body parts, which is

called body awareness. Besides, other muscle forms and more number of muscles could be simulated.

Appendix A

HUMAN MODEL IN XML FORMAT

In this appendix, the document type definition (DTD) of the human model used in our implementation and XML formatted data of the human model are given.

A.1 Document Type Definition (DTD) of Human Model

```
<!ELEMENT Humanoid (author, name, center, Joint*)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT center (#PCDATA)>
<!ATTLIST center value CDATA #REQUIRED>
<!ELEMENT Joint (Joint*, Segment*, Site*)>
<!ATTLIST Joint name CDATA #REQUIRED>
<!ATTLIST Joint center CDATA #REQUIRED>
<!ATTLIST Joint limitx CDATA #IMPLIED>
```



```

<!ATTLIST Joint limity CDATA #IMPLIED>
<!ATTLIST Joint limitz CDATA #IMPLIED>
<!ELEMENT Segment EMPTY>
<!ATTLIST Segment name CDATA #REQUIRED>
<!ATTLIST Segment mass CDATA #IMPLIED>
<!ELEMENT Site EMPTY>
<!ATTLIST Site center CDATA #REQUIRED>
<!ATTLIST Site name CDATA #REQUIRED>

```

A.2 XML Formatted Data of Human Model

```

<?xml version="1.0" ?> <Humanoid>
  <author>Aydemir Memisoglu</author>
  <name>Little</name>
  <center value="0.0000 0.0000 0.0000" />
  <Joint name="HumanoidRoot" center="0.1296 0.0000 -3.8563">
    <Joint name="sacroiliac" center="0.0075 0.0000 -4.2920">
      <Joint name="l_hip" center="-0.0005 -0.4446 -4.2920"
        limitx="-1.00 2.00" limity="-1.40 2.90"
        limitz="-1.40 1.30">
        <Joint name="l_knee" center="0.1441 -0.4446 -2.2778"
          limity="-3.00 0.00">
          <Joint name="l_ankle" center="-0.3444 -0.4446 -0.3070">
            <Joint name="l_midtarsal" center="0.1722 -0.4446 -0.0005"/>
          </Joint>
        </Joint>
      </Joint>
    </Joint>
  <Joint name="r_hip" center="0.0136 0.4446 -4.2920"
    limitx="-1.00 2.00" limity="-1.40 2.90"

```

```

limitz="-1.40 1.30">
  <Joint name="r_knee" center="0.1488 0.4446 -2.2993"
    limity="-3.00 0.00">
    <Joint name="r_ankle" center="-0.3585 0.4446 -0.3332">
      <Joint name="r_midtarsal" center="0.1722 0.4446
        0.0000" />
    </Joint>
  </Joint>
</Joint>
</Joint>
<Joint name="vl5" center="-0.3632 -0.0000 -4.8458">
  <Joint name="vt6" center="-0.3744 -0.0000 -6.2893">
    <Joint name="vt1" center="-0.2811 -0.0000 -6.7971">
      <Joint name="vc4" center="-0.1393 -0.0000 -7.1298">
        <Joint name="skullbase" center="0.0104 -0.0000
          -7.3858" />
      </Joint>
    <Joint name="r_sternoclavicular" center="-0.2544 0.3248
      -6.6328">
      <Joint name="r_acromioclavicular" center="-0.2877 0.3912
        -6.4830">
        <Joint name="r_shoulder" center="-0.2521 0.8925 -6.5425"
          limitx="-1.00 3.00" limity="-1.70 3.10"
          limitz="-1.00 1.30">
          <Joint name="r_elbow" center="-0.5892 1.9601 -6.3165"
            limity="0.00 3.00">
            <Joint name="r_wrist" center="-0.2611 2.9274
              -6.1421"/>
          </Joint>
        </Joint>
      </Joint>
    </Joint>
  </Joint>
</Joint>

```

```

        </Joint>
    </Joint>
</Joint>
<Joint name="l_sternoclavicular" center="-0.2652 -0.3838
-6.5804">
    <Joint name="l_acromioclavicular" center="-0.2984 -0.4502
-6.4779">
        <Joint name="l_shoulder" center="-0.2811 -0.9496 -6.5280"
        limitx="-1.00 3.00" limity="-1.70 3.10"
        limitz="-1.00 1.30">
            <Joint name="l_elbow" center="-0.4192 -0.9426 -5.1151"
            limity="0.00 3.00">
                <Joint name="l_wrist" center="-0.3728 -0.9285 -3.8543"/>
            </Joint>
        </Joint>
    </Joint>
</Joint>
</Joint>
</Joint>
</Joint>
</Joint>
</Joint>
</Joint>
</Humanoid>

```

Appendix B

THE SYSTEM AT WORK

In this part we introduce the user interface and the functionality of different components of the system. The system is implemented by Aydemir Memişoğlu and Mehmet Şahin Yeşil. Motion control part is implemented by Aydemir Memişoğlu using inverse kinematics based on analytical methods. Realistic rendering using a multi-layered human model that consists of skeleton, muscles and skin is implemented by Mehmet Şahin Yeşil.

B.1 Overview

Our application is Single Document Interface (SDI) application implemented using Visual C++ 6.0 and Microsoft Foundation Classes (MFC). The graphics display API OpenGL is used. The top level user interface of the system is seen in Figure B.1. The elements on the interface can be mainly divided into four parts:

- *The Main Menu:* This consists of menu bar and toolbar. It basically allows the user to control the application.

- *The Motion Control and Skinning Toolbox:* This toolbox allows user to control the skeleton, generate new motions and also model the skin mesh. It consists of three panels: Skeleton Panel, Motion Panel and Skinning Panel.
- *The Keyframe Editor:* This editor allows user to generate curves for distance, joint angles in order to characterize the motion of the articulated figure.
- *The Viewing Area:* The viewing area has quad view layout with the front, top, side, and perspective views of the 3D environment.

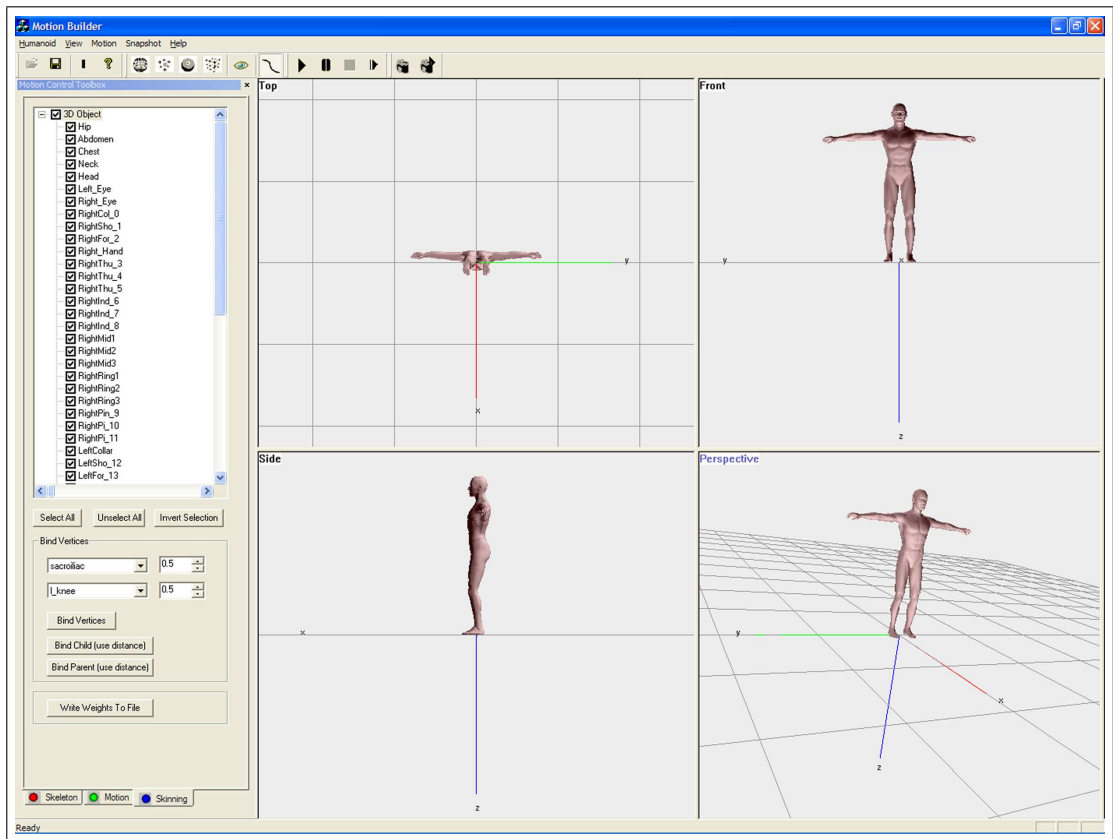


Figure B.1: Top level user interface of the system.

B.2 The Main Menu

The main menu of the program consists of the menu bar and the other toolbars. The menu bar includes “File”, “View”, “Motion”, “Snapshot”, and “Help” subitems and provides the general functionalities like creating and opening an object model, changing the user interface options, and creating a new motion or loading an existing one. The user also can start, stop, pause and play a loaded animation step by step. From the toolbar, it is possible to change the view of the model. Points view, wireframed and smooth shaded views are available in the program. The “Snapshot” menu item and the toolbar gives user the opportunity to take the snapshots of the animation.

B.3 The Motion Control and The Skinning Toolbox

The motion control and skinning toolbox includes three panels (see Figure B.2). The first panel, called the skeleton panel, contains skeletal information of the articulated figure. On the upper part of the panel, the hierarchical structure of the skeleton is illustrated in a MFC tree control. The node part of the panel shows the position and orientation information of the selected joint. These values of the joint can also be modified by the user and the changes will be directly adopted to the figure on the viewing area.

On the other hand, the motion panel of the toolbox allows the user to make modifications on the position curves of the end-effector (ankles and wrists) and root joints of the structure. These modifications result with different types of motion. The upper part of the motion panel enables the user to select the desired joint, and types of control for a joint. The user can add and delete joint to control or update the types of control for the selected controlled joint

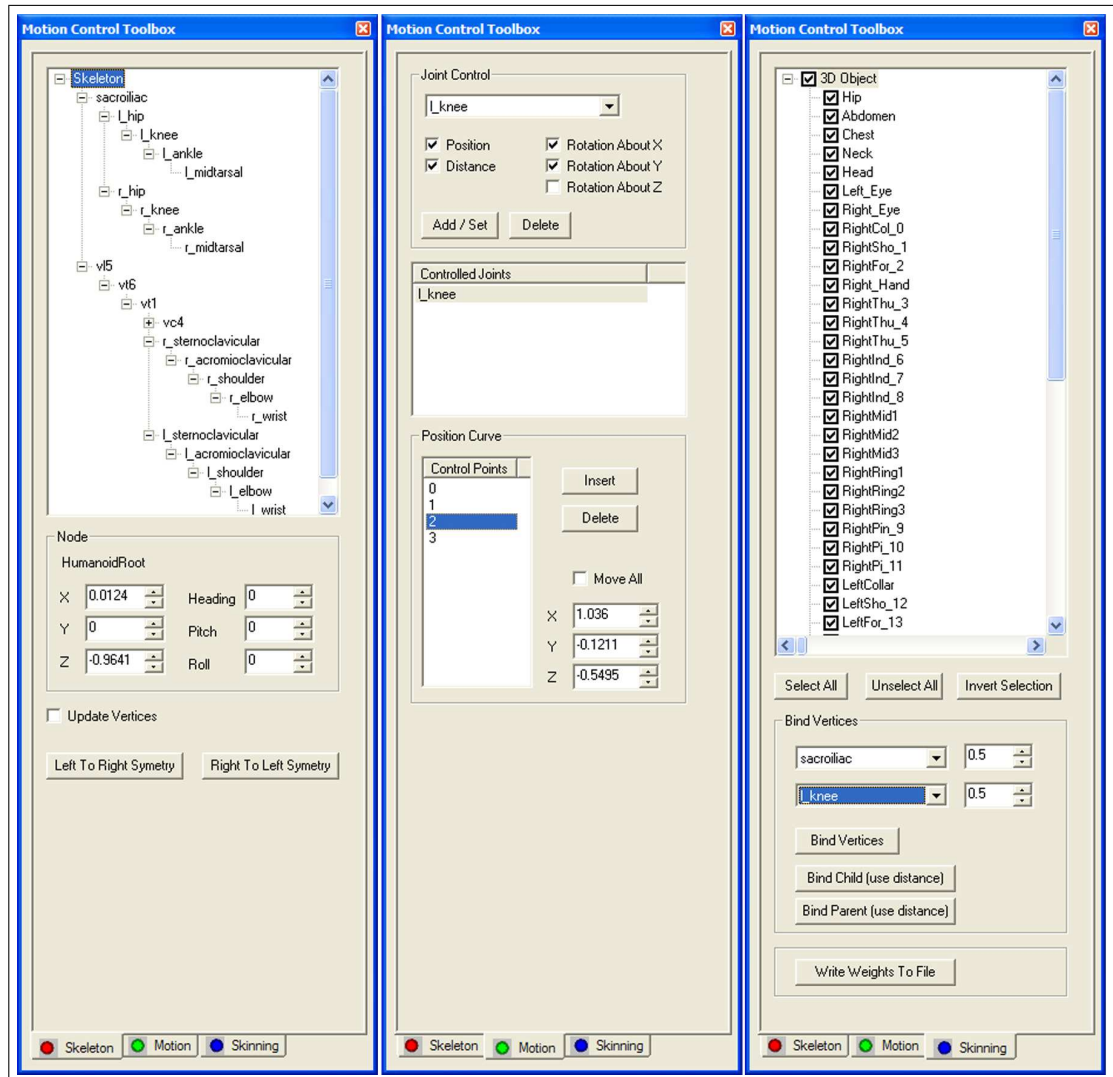


Figure B.2: The motion control and the skinning toolbox

by using ‘Add / Set’ and ‘Delete’ buttons. Selected joints will be listed on the ‘controlled joints’ listbox and the position curve of the joint will be also drawn on the viewing area. Properties of the position curve of the selected joint can be seen on the position curve part of the panel. These properties are mainly denoted by the control points. The user can either change the position of an existing control point or create a new one. Effects of these manipulations of the position curves are directly seen on the figure drawn in the viewing area.

In addition, this toolbox provides the required functionality for modeling the skin mesh by means of the skinning panel. As it is seen on the rightmost part of Figure B.2, the list on the top represents the body parts of the skin mesh. By checking the checkboxes on this list you can select which body parts will be seen on top of the skeleton. You can also change your selection by using the buttons under the object list. This panel mainly designed for modeling the skin and attaching the skin vertices to the underlying layers. We give the opportunity to bind each vertex of any body part with one or two joints. For this purpose, after selecting the appropriate body part you can mark any vertex of this part by clicking on the vertex. The next step is choosing the necessary joints and giving corresponding weights for these joints from the “Bind Vertices” area. Then, you can bind the vertices with the selected joints and also save this configuration for future works.

B.4 The Keyframe Editor

The keyframe editor toolbox will be displayed for a selected joint when one of the controlled joints is double-clicked. While the motion panel includes the controls for manipulating the position curve of a joint, the keyframe editor provides the functionality for editing the distance and orientation curves of the joint. The distance curve specifies the distance travelled over time. The rotation curves

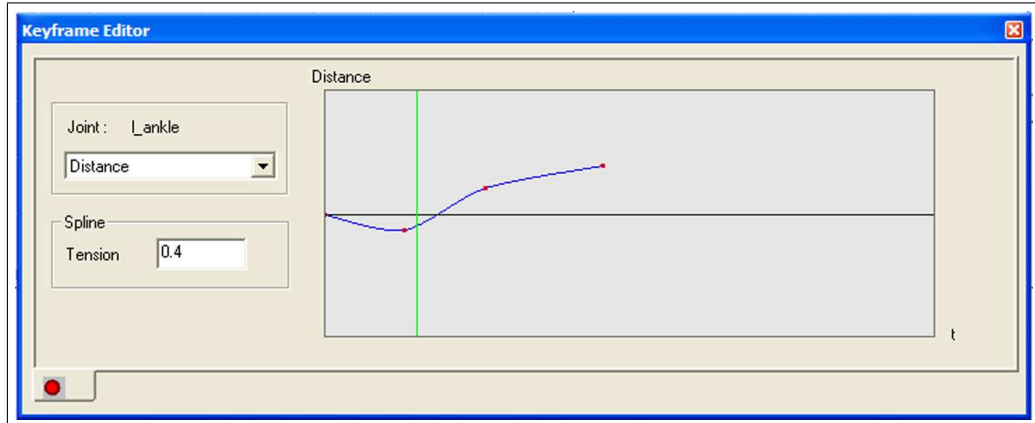


Figure B.3: The keyframe editor

specify the rotations about x, y, z axes applied over time. The editor will only include the curves that are selected in the motion panel. When the curve to be edited is selected, it will be drawn on the right frame of the editor (see Figure B.3). The curve can be edited at this frame by changing the positions of the control points and the changes can be followed on the viewing area. New control points can be added using the right mouse button and holding the ctrl key. Control points can be deleted using right mouse button and holding the ctrl key. A green line, which can be positioned along x-axis (axis representing time), enables the user to examine the pose of articulated figure at specified time. This makes the user to interactively manipulate the curve and see the effects at any time by positioning the green line on the x-axis using mouse. The user can also set the tension parameter of the curves which are Cardinal splines. All these functionalities provide the opportunity to create different motions easily.

B.5 The Viewing Area

The viewing area includes four view layout; top, front, side, and perspective (see Figure B.4). The user can either select and view one of the layouts or use all of them. Zooming and view point change operations can be performed in

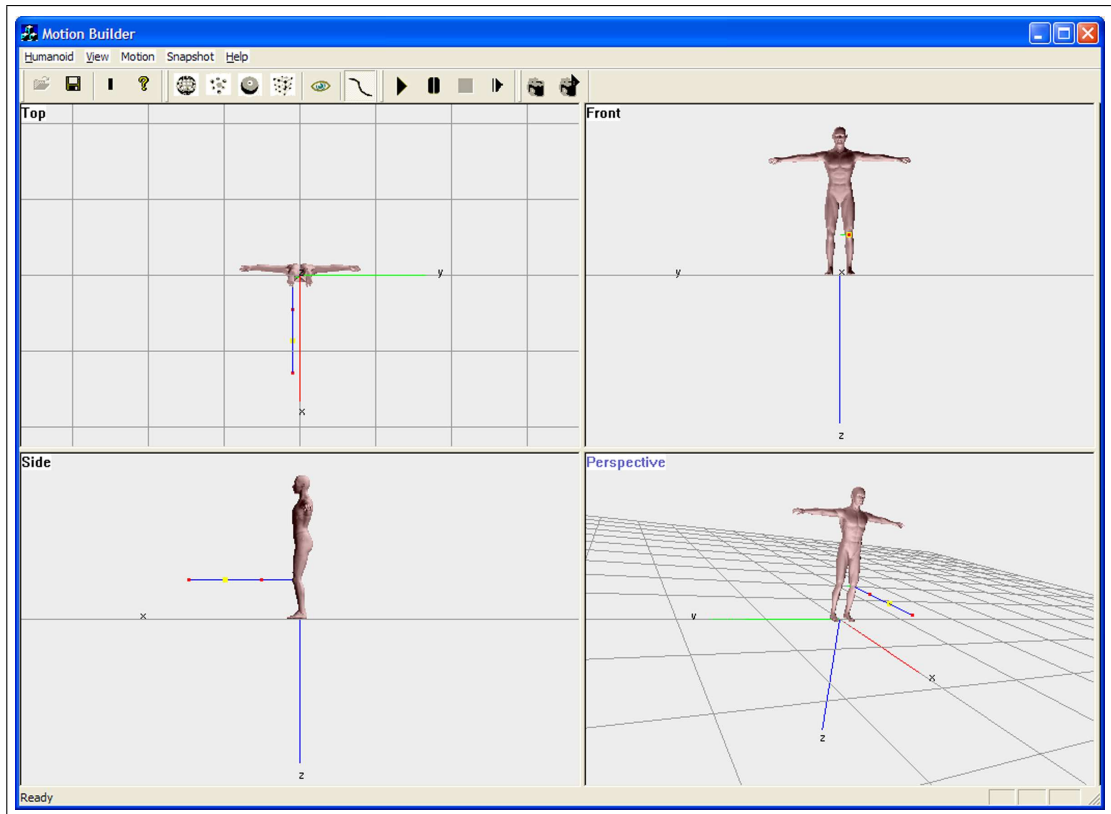


Figure B.4: The viewing area

each frame by using mouse and keyboard keys. Using left mouse button and holding the ctrl key, the user can rotate the figure in 3D space. The model can be translated using the right mouse button and holding the ctrl key. Zooming can be done using just the right mouse button.

Bibliography

- [1] Aubel, A., “Anatomically-based human body deformations”, PhD Thesis, No 2573, Ecole Polytechnique Federale de Lausanne, 2002.
- [2] Barr, A., “Dynamic constraints”, State of the Art in Image Synthesis Course Notes, ACM SIGGRAPH’86, 1986.
- [3] Badler, N. and Smoliar, S., “Digital representations of human movement”, ACM Computing Surveys, Vol. 11, No. 1, pp. 19-38, March 1979.
- [4] Badler, N., “Graphical behavior and animated agents”, Advanced Techniques in Human Modeling, Animation and Rendering, ACM SIGGRAPH’92 Course Notes #17, July 1992.
- [5] Blinn, J., “A generalization of algebraic surface drawing”, Transactions on Graphics, Vol. 1, pp. 235-256, 1982.
- [6] Bloomenthal, J. and Shoemake, K., “Convolution surfaces”, ACM Computer Graphics (Proceedings of SIGGRAPH’91), pp. 251-256, 1991.
- [7] Bloomenthal, J., “Hand crafted”, Modeling, Visualization and Animating Implicit Surfaces, ACM SIGGRAPH Course Notes #25, 1993.
- [8] Cani-Gascuel, M.P., “Layered deformable models with implicit surfaces”, Graphics Interface’98, Vancouver, Canada, 1998.

- [9] Chadwick, J., Haumann, D. and Parent, R., “Layered construction for deformable animated characters”, ACM Computer Graphics (Proceedings of SIGGRAPH’89), pp. 243-252, July 1989.
- [10] Deepak, T., Goswami, A., Badler, N., “Real-time inverse kinematics techniques for anthropomorphic limbs”, Graphical Models, Vol. 62, No. 5, pp. 353-388, September 2000.
- [11] Delp, S.L. and Loan, J.P., “A computational framework for simulating and analyzing human and animal movement”, IEEE Computing in Science & Engineering, Vol. 2, No. 5, pp. 46-55, Sept-Oct 2000.
- [12] DeLoura, M.A., *Game Programming Gems*, Charles River Media Inc., Rockland, Massachusetts, 2000.
- [13] DeRose, T., Kass, M., and Truong, T., “Subdivision surfaces in character animation”, ACM Computer Graphics (Proceedings of SIGGRAPH’98), pp. 85-94, 1998.
- [14] Fung, Y.C., *Biomechanics: Mechanical Properties of Living Tissues*, Springer-Verlag, New York, 1981.
- [15] Gascuel, M.P., Verroust, A., and Puech, C., “A modeling system for complex deformable bodies suited to animation and collision processing”, The Journal of Visualization and Computer Animation, Vol. 2, No. 3, pp. 82-91, 1991.
- [16] Gascuel, M.P., “An implicit formulation for precise contact modeling between flexible solids”, ACM Computer Graphics (Proceedings of SIGGRAPH’93), pp. 313-320, 1993.
- [17] Haumann, D.R. and Parent, R.E., “The behavioral testbed: Obtaining complex behavior from simple rules”, The Visual Computer, Special Issue on Mechanics, Control and Animation, Vol. 4, No. 6, December 1988.

- [18] Herbison-Evans, D., “Nudes 2: A numeric utility displaying ellipsoid solids”, SIGGRAPH’78 Conference Proceedings, pp. 354-356, August 1978.
- [19] Hirota, G., Fisher, S., State, A., Lee, C., and Fuchs, H., “An implicit finite element method for elastic solids in contact”, Proceedings of Computer Animation’01, Seoul, Korea, 2001.
- [20] Kavan, L. and Zara, J., “Real time skin deformation with bones blending”, WSCG’2003 Short Papers, University of West Bohemia, Pilsen, Czech Republic, pp. 69-74, 2003.
- [21] Komatsu, K., “Human skin model capable of natural shape variation”, The Visual Computer, Vol. 3, No. 5, pp. 265-271, 1988.
- [22] Korein, J. and Badler, N., “Techniques for generating the goal directed motions of articulated figures”, IEEE Computer Graphics and Applications, Vol. 2, No. 9, pp. 71-74, 1982.
- [23] Korein, J., *A Geometric Investigation of Reach*, The MIT Press, Cambridge, 1985.
- [24] Lander, K., “Skin them bones”, Game Developer, pp. 11-16, May 1998.
- [25] Lanir, Y., “Skin mechanics”. In *Handbook of Bioengineering*, Edited by R. Skalak, McGraw Hill, 1987.
- [26] Lasseter, J., “Principles of traditional animation applied to 3D computer animation”, ACM Computer Graphics (Proceedings of SIGGRAPH’87), pp. 35-44, 1987.
- [27] Leclercq, A., Akkouche, S., and Galin, E., “Mixing triangle meshes and implicit surfaces in character animation”, Proceedings of Animation and Simulation’01, pp. 37-47, Manchester, England, 2001.

- [28] Maurel, W., Wu, Y., Thalmann, M., and Thalmann, D., *Biomechanical Models for Soft Tissue Simulation*, Springer-Verlag, Berlin/Heidelberg, 1998.
- [29] Maurel, W. and Thalmann, D. “Human upper limb modeling including scapulo-thoracic constraint and joint sinus cones”, *Computers & Graphics*, Vol. 24, No. 2, pp. 203 - 218, 2000.
- [30] Memişoğlu, A., “Human motion control using inverse kinematics”, MS. Thesis, Department of Computer Engineering, Bilkent University, August 2003.
- [31] Min, K., Baek, S., Lee, G.A., Choi, H., and Park, C., “Anatomically-based modeling and animation of human upper limbs”, *Proceedings of International Conference on Human Modeling and Animation*, Seoul, Korea, June 2000.
- [32] “MetaCreations Software”, Poser 4,
<http://www.metacreations.com/products/poser>.
- [33] Mocozet, L., “Hand modeling and animation for virtual humans”, PhD Thesis, No 442, University of Geneva, 1996.
- [34] Nedel, L. and Thalmann, D., “Real time muscle deformations using mass-spring systems”, *Proceedings of Computer Graphics International (CGI’98)*, pp. 156-165, 1998.
- [35] Ng-Thow-Hing, V., “Anatomically-based models for physical and geometric reconstruction of humans and other animals”, Ph.D. Thesis, Department of Computer Science, University of Toronto, 2000.
- [36] The National Library of Medicine, “The visible human project”, available at http://www.nlm.nih.gov/research/visible/visible_human.html.

- [37] Porcher-Nedel, L. “Anatomic modeling of human bodies using physically-based muscle simulation”, Ph.D. Thesis, No 1831, Swiss Federal Institute of Technology, 1998.
- [38] Richer, P., “Trait danatomie artistique”, Bibliothque de limage, 1996.
- [39] Scheepers, F., “Anatomy based surface generation for articulated models of human figures”, PhD thesis, Department of Computer and Information Science, The Ohio State University, 1996.
- [40] Scheepers, F., Parent, R., May, F., and Carlson, W., “Procedural approach to modeling and animating the skeletal support of the upper limb”, Technical Report, OSU-ACCAD-1/96/TR1, The Ohio State University, 1996.
- [41] Scheepers, F., Parent, R., Carlson, W. and May, S., “Anatomy based modeling of the human musculature”, ACM Computer Graphics (Proceedings of SIGGRAPH’97), pp. 163-172, 1997.
- [42] Sederberg, T. and Parry, S., “Free-from deformation of solid geometric models”, ACM Computer Graphics (Proceedings of SIGGRAPH’86), pp. 151-160, 1986.
- [43] Singh, K., Ohya, J., Kishino, F. and Parent, R., “Human figure synthesis and animation for virtual space teleconferencing”, Virtual Reality Annual International Symposium’95, pp. 118-126, North Carolina, March 1995.
- [44] Singh, K., “Human figure synthesis and animation for VR applications”, PhD thesis, The Ohio State University, 1995.
- [45] Singh, K., Kokkevis, E., “Skinning characters using surface-oriented free form deformations”, Proceedings of Graphics Interface 2000, pp. 35-42, 2000.
- [46] Sloan, P.P., Rose, C.F., and Cohen, M., “Shape by example”, Proceedings of the Symposium on Interactive 3D Graphics, pp. 135-143, March 2001.

- [47] “Specification for a Standard Humanoid”, Version 1.1, <http://h-anim.org/Specifications/H-Anim1.1>.
- [48] Steketee, J. and Badler, N., “Parametric keyframe interpolation incorporating kinetic adjustment and phrasing control”, ACM Computer Graphics (Proceedings of SIGGRAPH’85), pp. 255-262, July 1985.
- [49] Sun, W., Hilton, A., Smith, R., and Illingworth, J., “Layered animation of captured data”, Proceedings of the 10th Eurographics Workshop on Animation and Simulation, Milano, Italy, 1999.
- [50] Talbot, D., “Accurate characterization of skin deformations using range data”, Master’s thesis, University of Toronto, 1998.
- [51] Terzopoulos, D., Platt, J.C., Barr H., and Fleischer, K., “Elastically deformable models”, ACM Computer Graphics (Proceedings of SIGGRAPH’87), July 1987.
- [52] Thalmann, M.N. and Thalmann, D., *Computer Animation: Theory and Practice*, Springer-Verlag, Berlin, 1985.
- [53] Thalmann, M.N. and Thalmann, D., “The direction of synthetic actors in the film Rendez-vous à Montréal”, IEEE Computer Graphics and Applications, Vol. 7, No. 12, pp. 9-19, 1987.
- [54] Thalmann, M.N. and Thalmann, D., *Synthetic Actors in 3D Computer-Generated Films*, Springer-Verlag, New York, 1990.
- [55] Thalmann, D., Shen, J., and Chauvineau, E., “Fast human body deformations for animation and VR applications”, Proceedings of Computer Graphics International’96, pp. 166-174. IEEE Computer Society Press, June 1996.
- [56] Turner, R. and Thalmann, D., “The elastic surface layer model for animated character construction”, Proceedings of the Computer Graphics International’93, Lausanne, Switzerland, Springer-Verlag, pp. 399-412.

- [57] Turner, R., “Leman: A system for construsting and animating layered elastic characters”, Computer Graphics-Developments in Virtual Environments, pp. 185-203, Academic Press, San Diego, CA, June 1995.
- [58] Wilhelms, J., “Modeling animals with bones, muscles and skin”, Technical Report UCSC-CRL-95-01, University of California, Santa Cruz, January 1995.
- [59] Wilhelms, J., “Animals with anatomy”, IEEE Computer Graphics And Applications, Vol. 17, No. 3, pp. 22-30, May 1997.
- [60] Wilhelms, J. and Van Gelder, A., “Anatomically based modeling”, ACM Computer Graphics (Proceedings of SIGGRAPH’97), pp. 173-180, 1997.
- [61] Yoshimoto, S., “Ballerinas generated by a personal computer”, The Journal of Visualization and Computer Animation, Vol. 3, No. 1, pp. 85-90, 1992.