3D VOXEL-BASED MUSCLE VOLUME DEFORMATION BY FINITE ELEMENT METHOD

CSE 523/524 Master Project Final Report

By

Student: Qinghong Zhu

Advisor: Prof. Arie E. Kaufman

Department of Computer Science
SUNY at Stony Brook
NEW YORK, NY 11794-4400

January 12, 1998

With Special Thanks

With deep gratitude I thank Dr. Arie E. Kaufman, Leading Professor, for his supervision and his financial support for my graduate study at StonyBrook. Thanks also to Mr. Yan Chen, Ph.D candidate at Department of Computer Science, who also made contribution of writing this report.

Abstract

This report presents a voxel-based biomechanical model for muscle deformation using finite element method (FEM) and volume graphics. Hierarchical voxel meshes are reconstructed from filtered segmented muscle images followed by FEM simulation and volume rendering. Physiological muscle force is considered and linear elastic muscle models for both static and dynamic cases are simulated by FEM. Voxel-based wireframe, polygon surface rendering and volume rendering techniques are applied to show real-time muscle deformation processes as well as realistic animations. The simulations and renderings demonstrate an excellent performance of the new model. An algorithm is offered to employ volume graphics in medical applications.

Keywords: Keywords: voxel-based, biomechanically-based muscle modeling, real-time muscle simulation, finite element method, hierarchical representation, volume graphics, volumetric deformation.

Contents

List of Figures

Chapter 1

Project Summary

In medical applications of computer graphics – including surgical simulation, treatment planning, and outcome prediction – there is a need to model both rigid bodies and deformable, soft tissues. The voxel-based object models of Volume Graphics provide a way to represent the interior structure of heterogeneous, deformable tissues. In addition, it is possible to use 3D medical image data directly with this representation without requiring the approximation of edges and surfaces. This project investigates physically realistic manipulation of voxel-based objects.

Physically based modeling and manipulation of 3D graphical objects has important applications in computer graphics – particularly in animation and simulation. Current research in the area focuses on the simulation and visualization of object behavior, including movement of rigid bodies; behavior of natural objects such as clothes, hair or soil; and simulation of object interactions or deformation. Most existing models are surface based, where physical principles are applied to surface primitives or spline control points. However, these surface approaches are insufficient for simulating the effects of manipulation and modeling on object interiors. For example, analysis of the collision between two heterogeneous deformable objects would be impossible with surface techniques.

In this project, we investigate physically realistic manipulation of volumetric objects. In a volumetric approach, physically based manipulation is achieved by distributing the underlying physics of the specified region of interest across voxels forming the region. In particular, we are currently studying: techniques that specify physically based behavior of both object surfaces and interiors; efficient techniques and algorithms that enable manipulation and visualization on both rectangular and curvilinear grids (where the grid topology does not change); and the use of a multi-resolution hierarchy on both rectangular and curvilinear grids to accelerate the speed of manipulations

We have been applying the algorithms and techniques that we developed to two complimentary problems. First, we use deformable geometric objects, such as spheres (which we will use to test our methods against the physical solution) and high degree polynomials for computer aided-design applications. Second, we apply the deformation of volumetric objects to a specific medical application such as the simulation of the macroscopic dynamics of muscle.

Chapter 2

Background and Introduction

Human figure modeling and animation have been one of the primary areas of research in computer graphics since the early 1970's. The complexity of simulating the human body and its behavior is directly proportional to the complexity of the human body itself, and is compounded by its vast number of movements.

2.1 Biomechanically-based vs Anatomy-based

Recently there has been a trend to model the human figure in an approach similar to the one taken in artistic anatomy - by analyzing the relationship between exterior form and the underlying structures responsible for creating it. Scheepers, F. et al. [Sche97] use ellipsoids to represent muscle bellies. They also build models for Fusiform muscles and multi-belly muscles. Each muscle model allows the extremities of muscles to be specified relative to different underlying bones. The muscle models manage the deformation of muscles due to isotonic contraction. Wilhelms, J. et al. [Wilh97]'s model consists of individual muscles, bones, and generalized tissues covered by an elastic skin. Muscles are considered as deformable object discretized cylinders lying between fixed origins and insertions on specific bones.

Despite these innovations, these models lack medical proof and their images are imprecise. Instead, we turned to a biomechanically-based approach. First, we extracted information from a volumetric dataset, such as slices of CT or MRI, through the use of image processing. Then we used biomechanically-based modeling, in which the behavior of objects is determined via simulations of biomechanical laws and math equations, to determine how bodies move and change shape over time.

Anatomy-based modeling may be useful for entertainment and art, but not suitable for medical applications. Biomechanically-based volume visualization, which applies volume graphics to real medical applications – including surgical simulation, treatment planning, and outcome prediction based on scanned images and biomechanical models – can give convincing simulation results.

2.2 Physically-based Modeling and Deformation

As the computer graphics field matures, there is growing interest in complex, physically-based models which combine elements of applied math, numerical analysis, physics, mechanical engineering and computational mechanics. Traditional computer graphics modeling includes modeling, defining the geometry of the objects in the picture, rendering, computing the image of those objects, animating and manipulating the object over time. Physically-based modeling, where the behavior of objects is determined via simulations of physical laws and math equations, focuses on how bodies move and change shape over time. It blurs the traditional distinction between modeling, rendering and animation.

During the past few years, physically based modeling has emerged as an important new approach to computer animation and computer graphics modeling. Witkin et al. [Witk93] summarized the methods and principles of physically based modeling. A good deal of work has been done toward physically-based models of objects such as rigid and nonrigid bodies, or natural terrain. Barzel [Barz92]thoroughly introduces a physically-based modeling approach in his book. Physically-based modeling, which incorporates physical characteristics into models, allows numerical simulation of their behavior and applies sophisticated rendering techniques to describe the real image.

Physically based modeling and manipulation of 3D graphical objects have important applications in computer graphics – particularly in animation and simulation. Current research in the area focuses on 3 topics: simulation and visualization of object behavior, including movement of rigid bodies; behavior of natural objects such as clothes, hair or soil; and simulation of object interactions or deformation. Most existing models are surface-based, where physical principles are applied to surface primitives or spline control points. However, these surface approaches are insufficient for simulating the effects of manipulation and modeling on object interiors. For example, analysis of the collision between two heterogeneous deformable objects would be impossible with surface techniques.

A physically-based modeling technique makes the deformation and behavior realistic. In medical applications of computer graphics cited above, there is a current need to model both rigid bodies and deformable soft tissues. The voxel-based object models of Volume Graphics provide a mean to represent the interior structure of heterogeneous, deformable tissues. In addition, it is possible to use 3D medical image data directly with this representation without requiring the approximation of edges and surfaces. This project investigates physically realistic manipulation of voxel-based objects.

There are two challenges for volume deformation of soft tissue objects. First, we need to build up a 3D dynamic model based on the plastics theory, which has traditionally proved difficult in biomechanical research for human muscle, and where little work has been done. Second is speed; the classical 3D elastic model is composed time dependent partial differential equations which generally need to be done by iteration, a major setback for speeding up the processes. Chen et al. [Chen96] use a two-stage dynamic deformation for construction of a 3D model. They reduce undesirable efforts of oversmoothness, local concentration, and folding from the sparsity and randomness of sampled data. Nielson et al. [Niel96] try to reduce matrix manipulation time. Little improvement was achieved in these trials.

2.3 Medical Application

Muscle deformation simulation is an essential part of virtual surgery, which could make a tremendous impact on surgical morbidity and mortality. Studies have shown that, for a wide range of diagnostic and therapeutic procedures, physicians performing their first few cases are much more likely to make errors. Physician error could be reduced by developing the virtual surgery that allows transference of skills from the simulation to the actual patient.

In orthopedic surgery planning, traditional X-ray based techniques are not able to sufficiently image the soft tissue components of the orthopedic structures of interest (tendons, ligaments, attached muscle, etc.)the superposition of structures in 3D, when projected onto

2D plain films, severely limits the usefulness of any one viewpoint in providing accurate diagnostic and/or therapeutic planning information [Robb]. Here the muscle volume deformation is useful for investigation of the tissue structures within the body and helps to design the surgery strategy.

Chapter 3

Literature Survey

3.1 The Application of FEM in Computer Graphics and Medicine

Generally, two modeling methods are used for physically-based modeling on the human body: a mass spring model and FEM model. The mass spring model is simple and gives acceptable and interactive first impressions of human body deformation, yet such a model cannot describe the exact physical behavior of the human body. Early works include: Waters[Water87]'s parameterization facial muscle model, which incorporates the use of a spring model to create a realistic three-dimensional facial expression. Provot et al. [Prov95] describes a 2D physically-based model for animating cloth objects derived from elastically deformable models, improved in order to consider the non-elastic properties of woven fabrics. Their models, composed of double cross springs, are hard to interpret and their results are quite different depending on various spring structures. A recent paper by Promayon et al. [Prom96] presents a method to constrain physically-based 3D deformable objects where an object can be defined locally in terms of kinetic and dynamic mass, position, speed, and physical parameters. They use only a mass-spring model and develop some control techniques, but only an approximate physical model is considered.

Gallagher, et al. [Gall92,95] reviews volume visualization for finite element analysis and its applications. Data volume visualization is based on discretization techniques. Celniker et al. [Celn91] apply FEM to generate primitives that build continuous deformable shapes designed to support a new free-form modeling paradigm. FEM generally deal with the physical model governed by partial differential equations; the complexity and computation time increase sharply, particularly when applying FEM to a dynamic system in the volumetric case. Additionally, there is great demand for memory storage and real time speed becomes very slow. FEM is also used in radiosity algorithms. Troutman [Trou93] selected a point collocation method based on finite element analysis to incorporate an interpolation function directly into factor computation.

Terzopoulos et al. [Terz87, 88, 90, 92] used both finite different and finite element methods in modeling physical deformation based on the elasticity theory. They modeled both rigid and deformable objects and considered physically realistic interactions and animation. The simulation and animation were created by the dynamic model.

Tabatabai et al. [Taba94] presented the principle methods for volume rendering of non-regular grids as used in the finite element method or the method of control volumes. The most basic operation in the whole rendering process, namely the interpolation of geometry and function values, is fully based on the element type specific shape functions, ensuring a consistent mathematical treatment of the object in the sense of the underlying numerical simulation.

Pieper et al. [Piep92] have implemented a system for computer-aided plastic surgery which makes the analysis process easier by allowing the surgeon to draw the surgical plan directly onto a 3D model of the patient. An automatic mesh generator is used to convert that drawing into a well-formulated problem for finite element analysis. Regarding the visualization of the FE model, they subdivide the outer face of each element into micropolygons where the position of each micropolygon vertex is transformed back into cylindrical coordinates which are used to sample the cybernare range and color data. This method has greatly reduced both the number of nodes which implements the finite element matrix, and thus, computational time.

Chen et al. [Chen92] have done a simulation and animation of a 3D biomechanical FEM model. In their model, skeleton kinematics and physiology effect were considered. 20 node brick FEM mesh and some user interfaces were developed, but no volumetric case is implemented. Their work is interesting because they introduce an advanced engineering method to computer graphics and animation. Apparently, Chen's model is much less accurate in describing real muscle shape and rendering of the muscle. is also difficult to provide a medical application for it.

More recently, Keeve et al. [Keev96] developed an anatomy-based facial tissue modeling for surgical simulation using the FEM. They integrated an anatomy-based 3D finite element tissue model with a computer-aided surgical planning system, which allowed precise prediction of soft tissue changes resulting from realignment of underlying bone structure. However, in their model, they only consider a static model, and their nonlinear model is still under development.

Speed is the overriding concern in surgery simulation. Human organs and tissue have very complex elastic behaviors, so a surface simplified model is used to achieve realistic animation. Computation for simulation and visualization of a 3D FEM model is very time consuming; it is imperative to accelerate the simulation and rendering by using a fast algorithm. Multigrid methods and modal analysis have been used to reduce computational time Metaxas, et al. [Meta92]. Bro-Nielsen et al. [Niel96] discuss the application of 3D solid volumetric FEM to surgery simulation. They introduced three new ideas: the first is the condensation which compresses the linear matrix system by considering only the surface model; the second is the use of a linear matrix system for simulation, and the third is the use of a selective matrix vector multiplication. Despite claims that have achieved real-time performance, but they do not consider rendering and their images are not clear.

3.2 Volume Graphics

In the field of medical visualization(MRI, CT), a recently increasing trend is use of discrete 3D voxel representation called volume graphics, allowing us to observe not only the surface but also interior of the object. In volume modeling, we simulate the physically-realistic behavior of volumetric objects which include hundreds of thousands of volume elements. For real time interaction, a fast algorithm is imperative to make trade-off between physical realism and speed. This is very challenging work in the application of volume graphics due to the complicated 3D physical model and computation costs, as well as the large memory requirement for computing and rendering.

Kaufman [Kauf93] introduces the field of volume graphics where a voxel based data format is used to represent graphical objects, which includes volumetric data handling, voxelization of geometric models, volume viewing and shading in addition to volume rendering. In a recent paper, Hong et al.[Hong97] present an interactive virtual colonoscopy method, using a physically-based camera control model that employs a potential field and rigid body dynamics. It is considered as the pioneer work on volume graphics and physically-based modeling with application to the human colon. Their physical models, however, deal with a rigid object which is governed by ordinary differential equations. For soft tissue, we need to develop an advanced elastic model.

Gibson [Gibs95] proposed the use of a voxel-based data presentation not only for visualization, but also for modeling objects and structures derived from volumetric data. She describes the use of a voxel-based format for modeling physical interaction between virtual objects for the 3D Magnetic Resonance Image(MRI). More recently, Gibson[Gibs96] suggested a 3D chainmail algorithm which enabled fast deformation of volumetric objects. It can manipulate a large dataset in both static and dynamic volumetric deformation, but no physical model is included in her algorithm. Gibson[Gibs97] simply incorporates force feedback through haptic devices. Other approaches for the deformation problem include free-form deformation Sederberg et al. [Sede86], Hsu et al.[Hsu92]; the deformation can be easily applied either globally or locally for geometrical modeling but not in physics. Interacting particles modeling considers the physical changes, yet it was applied only for 2D woven cloth by Breen et al.[Bree94]. Another work by Desbrun et al. [Desb95] employs a hybrid model for animation of soft inelastic substances which used a particle system.

For volume rendering of the deformable objects, two methods are used. One method is direct rendering irregular meshes in the physical space. Related works can be found by Kaufman and his colleagues Silva, C., et al. [Silv96], Mao, X., et al[Mao95]. The other approach includes: doing math transformation, changing the irregular meshes on the physical space to regular meshes on the computational space, and rerendering. Work done by Fruhauf[Fruh94] is a interesting approach using Jacobi transformation, but there are still some math arguments such as the conditions and continuousness remaining in his method.

3.3 Hierarchcal Graphics Modeling

To render complex scenes in real-time using traditional means, the speed, the memory and network bandwidth become the bottleneck. Two major techniques can improve the rendering process. One is to use visibility culling to avoid displaying objects that are completely occluded. Another is the use of multiresolution or level-of-detail(LOD) modeling. The multiresolution method can accelerate both the simulation and rendering process without great loss of image quality.

In Heckbert et al[Heck94]'s paper, the data structures used in multiresolution modeling are summarized. The main data structures of multiresolution include: image pyramids, volume pyramids, ray space and polygonal models. Polygonal models are probably the core of a successful mutiresolution modeling system. Haley[Haley96] presented a new algorithm for efficient incremental rendering of volumetric datasets. They used the efficient Shear-Warp Factorization method and represented the volumetric data using a hierarchical data structure which provides for incremental classification and rendering of volumetric data. These factors make octree algorithms more suitable for implementation on average workstations. Chamberlain et al [Cham96] used the approach to construct a partial hierarchy of cells over the scene and associate a simplified representation of its contents within each cell. This method can accelerate rendering of complex static scenes. They apply the method to several different scenes and demonstrate significant speedups with little image degradation.

3.4 References of C/C++ Finite Element Codes

In recent years, there are number of finite element codes developed by C/C++ compared with traditional FORTRAN finite element codes. It is advantageous to have C/C++ finite element codes to implement graphics models. The following is a partial list of some popular C/C++ finite element packages:

(1). ClassLib:
developed by Michael M. Tiller at UIUC in 1993. He implemented an Object-Oriented Finite Element Code to solve 2D elasticity problems. The document is in Visualization Lab.

(2). KASTIO:
developed by Bodo Erdmann, Jens Lang and Rainer Roitzsch at Konrad-Zuse-Zentrum

fur Informationstechnik Berlin in 1994. It dealt with standard PDE. 2D/3D elasticity, 1D/2D/3D wave equation, convection/diffusion problems etc. Note that no time dependant problems were solved. It was written by C/C++ and can be found at:
http://ftp.ccs.uky.edu/mgnet/Codes/kaskade

(3). ug 3.0:
written by Peter Bastian et al. at University of Stuttgart. Application included 2D Navier-Stokes equations, Elasticity PDE etc. It requires a license to run this code. C source is available at:
http://ftp.ccs.uky.edu/mgnet/Codes/ug/

(4). FElt-3.00:
written by J. Gobat(MIT) and D. Atkinson (UCSD) which dealt with 1D/2D/3D static and dynamic elasticity and heat transfer problems. They solve simple cases and graphics tools are used for 2D display only. C source code can be found at:
http://www-cse.ucsd.edu/users/atkinson/FElt/

(5)Diffpack:
an object-oriented (C++) framework for solving PDE's using finite element methods written by Are Magnus Bruaset and Hans Petter Langtangen in Norway in 1997. Applications include standard pde's, 2D/3D elasticity, 1D/2D/3D wave equation, non-linear 3D wave equations, convection/diffusion problems, Boussinesq equations, two-phase porous media flow, incompressible Navier-Stokes equations, etc. Reference manuals and tutorials are available. C++ source codes can be found at:
http://www.nobjects.com/prodserv/diffpack

Chapter 4

System Overview

4.1 System Analysis and Design

In the following section we present a human anconeus muscle model that can enable prediction of anconeus deformation after a force is employed. In contrast to prior approaches, our system is a combination of scan-based visualization, automatic hierarchy mesh generation algorithms, and physically-based biomechanical modeling using FEM and real time interactive volume rendering. The components of our FEM Muscle Volume Deformation Simulation System are presented in Fig. 4.1.

In summary, we develop the models described earlier in four new ways:

- implement a muscle biomechanical model which has sound proof and is well recognized by the medical experts.
- generate muscle voxel mesh in the level of detail; Use Gaussian Filter to smooth mesh.
- build up an elastic soft tissue model for muscle deformation; techniques that specify physically based behavior of both object surfaces and interiors are realized.
- apply volume graphics in the elastic soft tissue model for the first time.

In this report we first give an overview of the system in this section and describe the general set-up as a combination of graphics tools and individual components in the following section. In Section 3 we elaborate the hierarchy voxel mesh reconstruction by scan-based muscle image, which is the prepossessing step to build the muscle model. Section 4 illuminates the famous biomechanical model for muscle movement by Dr. Jack Stern. Section 5 reviews the finite element method we employ with particular emphasis given to the direct integration algorithm for a 3D voxel-based FEM model. We present different volume graphics models for muscle images in section 6, ranging from a 3D wireframe model to volume rendering. Finally, we demonstrate the proposed system with experimental results obtained from the Visible Human Data Set(VHD) [National, 96].

4.2 System Pipeline

This section expands on the different procedures, data processing steps and general set-up of our simulation system for the process of the volume muscle deformation illustrated in the Fig 4.2.

Our data sources consist of laser range scans and CT data or, for the VHD, photo slices. In order to generate a model for an individual patient, the following steps are necessary:
(1). an initial muscle surface is extracted from the data sources and a hierarchy voxel mesh is computed for further processing.
(2). the biomechanical model and data must be synthesized, and for this the finite element method is used. All deformation is performed interactively on the precomputed FEM mesh. The FEM pipeline is comprised of computations for local and global stiffness matrices, preloading, assembling, solving and disassembling to obtain the new shape.
(3). the 3D mesh is voxelized and rendered by *VolVis*.

Chapter 5

Biomechanical Model for Anconeus Muscle Movement

5.1 Stern's Muscle Model

In 1974, Stern devised a mathematical model incorporating mechanical and physiological parameters of an idealized bone-muscle system, in which the movement of a limb under the action of a muscle, anconeus is used in our experiment See Fig. 5.1.

The model considers two aspects of muscle physiology: the effect of shortening lengthening on tension development and of length on tension. His model employs the following assumptions:
(1) At any moment in time during the contraction of a muscle, its exerted force depends only on its level of activation, instantaneous length and instantaneous velocity of contraction.
(2) The idealized muscle is parallel-fibered, crossing one joint, and with point attachments;
(3) The effort of moving the muscle is disregarded;
(4) The tendon and contractile element are inelastic;
(5) When comparing muscles with different sites of attachment on the moving limb segment, the effect on the moment of inertia of the segment is ignored.

Given the above, the force produced by a muscle at maximum activation obeys the following expression:

$$P = \left\{ \frac{(P_0)_l/a + 1}{\frac{1}{K_i}\frac{1}{(B/C+2+C/B)}\frac{sin(\alpha_i \dot{\alpha}_i)}{j.Y.K_l} + 1} - 1 \right\} a \tag{5.1}$$

The newly introduced variables are:
K_i = the straight line distance from one attachment to the other divided by the sum of the distances from the joint to each attachment point.
K_l = the value of K_i when the joint is positioned so that the length of the contractile tissue is L_0.
j = the value of b expressed in muscle lengths per sec.
α_i = the angle between the bones.
$\dot{\alpha}_i$ = the angular velocity of movement.
Y = the fraction of the distance A_i devoted to contractile tissue when the joint is positioned so that the contractile tissue is at L_0.

Fig. 5.2 illuminates the schematic drawing of a parallel-fibered one-joint muscle with point attachments. In the text the phrase "maximum attachment distance" refers to C in the figure, "minimum attachment distance" to B and "attachment ratio" to C, B. The fomula at the upper left expresses the instantaneous force of the muscle.

5.2 Zajac's Muscle Model

Another popular model used in determining the muscle force is Zajac's force model. Zajac gives the non-specific, dimensionless functions to model a musculotendon actuator. The isometric muscle force function can be written,

$$F = F_{fa} \tag{5.2}$$

$$F = (F_{iso} + F)cos\alpha \tag{5.3}$$

In the dynamic case, the total active force from the contractile element is seen to be a function of activation a(t), the force function and the normalized muscle velocity v_r so that the force in the whole muscle is:

$$F_M = (F + F(l^M))cos(\alpha) \tag{5.4}$$

Chapter 6

Hierarchical Voxel Mesh Reconstruction

In this part, image processing techniques and a hierarchical method are used for the generation of voxel meshes. Our data source is the CT anconeus dataset from Visible Human Dataset (VHD)[11]. We used 71 slices, with dataset resolution of 140x220x71.

6.1 Muscle Image Segmentation

The anconeus muscle image is generated through manual segmentation. Fig. 6.1(a) shows a muscle image through manual segmentation. The algorithm uses a multi-scale filtering (wavelet) and fuzzy based image segmentation approach. In this method, the wavelet filtering technique is utilized to automatically find the optimal segmentation threshold, and the classification of image is performed by means of multi-threshold and fuzzy clustering techniques. This method is used to segment the anconeus muscle areas from biceps and triceps of the VHD to reconstruct the 3D muscle volume. Fig. 6.2(b) uses a ray casting algorithm to show the muscle image obtained from this segmentation method. Muscle's reconstructing voxel meshes are based on the segmented images.

6.2 Hierarchical Voxel Mesh

IN the subsequent phase, a hierarchical voxel mesh is reconstructed and serves as a multiresolution volumetric approximation of the original muscle data. For each level of detail, we first read the muscle raw file generated from manual segmentation, then after filtering, construct the voxel mesh based on the threshold value of the muscle image. The basic operation consists of constructing the voxel mesh inside of the 3D muscle data and forfeiting other information outside of it. In order to carry out the FEM simulation, we need to build the voxel mesh according to the data structure determined by FEM simulator , which contains 3D geometric coordinates and topology information between each voxel.

In order to obtain a high resolution image, each voxel is subdivided into micro-voxels by the octree method. Consequently, the hierarchical voxel meshes are produced. Based on the different resolution requirements of the application, we construct different level of details of the muscle meshes. Because it takes so long to simulate high-resolution muscle deformation using 3D FEM, it is advantageous to have muscle shape change in low and medium-resolution, and apply FEM simulation to it. With the muscle shape deformation at low, medium-resolution, it is then relatively easy to predict high-resolution deformation image by comparison, mapping and interpolation. Similarity theory widely used in engineering design is used to compare the deformation picture between low, medium and high-resolution images. Pieper et al. [Piep92]used another approach by carrying out FEM simulation with a smaller number of nodes and elements and then subdividing each element into micropolygons to create a high-resolution image.

The voxel mesh is used in the rendering as well, so we can integrate real-time simulation and rendering based on the same mesh structure.

Chapter 7

FEM Simulation for Muscle Deformation

We have developed our algorithms for both static and dynamic systems based on elastic theory and finite element principles. In the following, we discuss our FEM algorithms.

7.1 Theory

In our model, we assume the muscle is behaving as an elastic soft tissue. We consider it a 3D muscle with an 8-node 3D brick element, equivalent to the voxel structure in volume graphics. In the problem-solution considered here, the linear analysis conditions required are that:

• the displacement be infinitesimally small so that the equilibrium of the body can be established with respect to its unloaded configuration;

• the stress-strain material matrix vary as a function of X, Y, Z, but as a constant otherwise.

To calculate muscle response, we establish the governing differential equations of equilibrium for both the surface and interior of the muscle, which then must be solved subject to boundary conditions. In the following we discuss two basic systems: static and dynamic as used in our model.

7.2 Static System

The linear elastic model can be obtained by the general elasticity equilibrium equation. The effect of body forces is ignored and the system equation can be written as:

$$K\vec{x} = b \tag{7.1}$$

where $\vec{x}$ is the displacement vector and b is a nodal load vector. The coefficient matrix K is a stiffness matrix. The boundary conditions of the system are given by muscle nodes which belong to the FEM mesh. In 3D problems, the size of the stiffness matrix increases rapidly even with the banded method of handling. We use a frontal method where the order of element numbering plays a more important role than the order of node numbering. The frontal method relies on the fact that a degree of freedom can be eliminated as soon as all stiffness values in rows and columns for that dof are complete. The dof(degree of freedom) corresponding to all these nodes can be eliminated as soon as element is assembled. Once a dof is eliminated, the corresponding equation is no more necessary until backsubstitution. This equation can be written to an external device such as a tape or hard disk, for back-substitution in reverse order. As we assemble an element the active matrix size grows, and when some degrees of freedom are eliminated the matrix size shrinks. The active matrix size can be compared to the action of an accordion. The largest block size needed can be determined using a prefront routine that employs a modified element connectivity matrix. By solving the linear sparse system we obtain displacements for all the nodes of the FEM mesh. These displacements are implemented into the graphical modeling system. The muscle shape changes can be visualized using different graphical techniques.

7.3 Dynamic System

Based on Lagrangian dynamics, the deformable model equations of motion can be expressed in 3D vector form by the second-order ordinary differential equations:

$$M\frac{\partial^2\vec{x}}{\partial t^2} + C\frac{\partial\vec{x}}{\partial t} + K(\vec{x})\vec{x} = \vec{f}(\vec{x}, t) \tag{7.2}$$

where $\vec{x}$ is a 3n vector of nodal displacement, M, C *and* K are 3n×3n matrices describing the mass, damping and stiffness between nodes within the volumetric object respectively, and f is a 3n vector of forces applied to each node.

The solution of the displacement, velocity and acceleration at time $t + \Delta t$ can be obtained by using center and Euler finite differences written by Lagrangian equation for the full system as:

$$\frac{\partial^2 \vec{x}}{\partial t^2} = (x_{t+\Delta t} - 2x_t + x_{t-\Delta t})/\Delta t^2 \tag{7.3}$$

And

$$\frac{\partial \vec{x}}{\partial t} = (x_{t+\Delta t} - x_{t-\Delta t})/2\Delta t. \tag{7.4}$$

Let's define

$$K^* = K(x_t) + \frac{M}{\Delta t^2} + \frac{C}{2\Delta t} \tag{7.5}$$

$$u_t = (x_t - x_{t-1})/\Delta t. \tag{7.6}$$

$$f^*_{t+\Delta t} = (\frac{1}{2\Delta t}C + \frac{M}{\Delta t^2})x_t + (\frac{M}{\Delta t} - \frac{C}{2\Delta t})u_t + f_{t+\Delta t}. \tag{7.7}$$

Substituting (4), (5), (6), (7) and (8) for (3), we can obtain

$$K^*(x_{t+\Delta t}) = f^*_{t+\Delta t} \tag{7.8}$$

Consequently, the explicit procedure evolves the dynamic solution from the given initial conditions x_0 and u_0 by solving a time sequence of static equilibrium problems for the instantaneous configurations x_{t+1}. Thus, the original nonlinear partial differential equations have been reduced to a sequence of sparse linear algebraic equations.

In order to solve the sparse linear system equations quickly, we assume $K^*_t = K$ is time-variant; thus the matrix decomposition solver need only perform a single initial decomposition of K. which significantly reduces the total amount of computation required. Furthermore, the mass matrix and damping matrix are generated at each node by using diagonal damping and mass matrices,

$$M_{ii} = \frac{1}{3}\rho V, C_{ii} = \beta M_{ii} \tag{7.9}$$

where ρ is the mass-density, β is a scaling factor and V is the volume.

7.3.1 Algorithm for a Modified Direct Integration Method

The algorithm is as follows:

(1) Initial calculations:

- form 3D brick stiffness matrix K, mass matrix M, and damping matrix C;
- calculate initial displacement, velocity and acceleration: $x^0, \dot{x}^0, \ddot{x}^0$;
- store matrix value in memory;
- calculate static deformation based on frontier solver.
- calculate static deformation based on frontier solver.

(2). Precomputing:

- select time step size Δt and calculate integration constant;
- derive dynamic difference scheme based on static deformation and its matrix;
- store matrix in one dimension form using back substitution.

- calculate new sparse matrix based on stored matrix;
- solve for displacements at time $t + \Delta t$ using explicit differential scheme based on the stored matrix values (K, M, C);
- calculate new velocity and acceleration at time $t + \Delta t$.

7.3.2 Algorithm for a Simplified Modal Analysis Method

Time history modal superposition is probably the most commonly used method to calculate dynamic response of systems of finite extent subjected to loads with a known time variation for a large number of nodes. It is advantageous to use this method to show realistic animation for high-resolution muscle. The algorithm for modal analysis is described as follows:

(1).formulation of the equation of motion;

(2). modal analysis provides the modal matrix and natural circular frequencies;

(3). uncouping of the equations of motion;

(4). combination of the modal responses;

For a 2-degree-of-freedom system, this can be written:

$$x_1(t) = \phi_{11} z_1(t) + \phi_{12} z_2(t) \tag{7.10}$$

$$x_2(t) = \phi_{21} z_1(t) + \phi_{22} z_2(t) \tag{7.11}$$

where

$$z_1(t) = \frac{c_0}{\omega_1^2}\left[1 - e^{-\beta\omega_1 t} f_1(sin(\beta, \omega t))\right] \tag{7.12}$$

$$z_2(t) = -\frac{c_0}{\omega_2^2}\left[1 - e^{-\beta\omega_2 t} f_2(sin(\beta, \omega t))\right] \tag{7.13}$$

β is the dumping coefficient, ω_1 and ω_2 are eigenvalues and $\phi_{11}, \phi_{12}, \phi_{21}, \phi_{22}$ are eigenvectors. These coefficients can be obtained by solving eigenmatrix. For the large number of nodes, computation time for each node displacement x_i is very small, therefore it is possible for us to obtain a realistic animation picture for high-resolution muscle images. In our experiment, we used a simplified model in which the displacement is the function of $e^{g(t)}$ and we have some promising results. (See Video Type)

7.4 Graphics Implementation

Our FEM algorithms are integrated with wireframe and shaded polygon graphics so that real-time display is realized in our FEM simulation system. The algorithms are written with C/OpenGL. In the following section, we review several graphical techniques and show how we integrate them together.

7.5 Hexahedral/Voxel-based Input Fils for FEM analysis

7.5.1 FEM Input File

(1). Sample Input File:

```
'Next Line is the Problem Title
3-D ANALYSIS USING HEXAHEDRAL ELEMENT
NN NE NM NDIM NEN NDN
8 1 1 3 8 3
ND NL NCH NPR NMPC
12 3 1 3 0
Node # X Y Z
1 0 0 0
2 100 0 0
3 100 100 0
4 0 100 0
5 0 0 100
6 100 0 100
7 100 100 100
8 0 100 100
Elem # N1 N2 N3 N4 N5 N6 N7 N8 MAT # Char1(NCH)
1 1 2 3 4 5 6 7 8 1 0
DOF # Load
13 0
14 0

10000000
MAT # Prop1 Prop2 Prop3
1 2000 0.3 0
2 500 0.3 0
3 1000 0.3 0
B1 i B2 j B3 (Multipoint constr. B1*Qi+B2*Qj=B3)
```

(2). Definition:

NN: Number of Node;

NE: Number of Element;

NM: Number of Material;

NDIM: Number of Dimension;

NEN: Number of Node in Each Element;

ND: Number of Total Freedoms of the Fixed Nodes;
NDN:
NL: Number of Load

Node #: The Serial Number for Each Node;
X: x-coordinate for Each Node;
Y: y-coordinate for Each Node;
Z: z-coordinate for Each Node;
Elem #: The Serial Number for Each Element;
MAT #: Number of Material;
DOF #: Degree of Freedom;
Prop1: Yong's Coefficient;
Prop1: Elasticity Coefficient;

Chapter 8

Visualizing Volumetric Muscle Deformation

We have developed three visualization techniques to visualize our deformed muscles: wireframe, shaded polygons and volume rendering.

3D wireframe is the fastest but least realistic form of display. Objects are drawn as though made of wire, with only their edges showing. The visible (within the view volume) portions of all edges of all objects are shown in their entirety, with no hidden-edge removal. All edges attributes affect screen appearance in their designated way in this model. We use this model with OpenGL in Fig. 6.

Shaded polygons display filled areas and polyhedra in a more realistic fashion. The addition of shaded areas to the rendering process increases the complexity significantly, because spatial ordering becomes important. Figs. 7 and Figs. 9 are rendered by using OpenGL in this way. Both wireframe and polygon shadow models are integrated into FEM routine to create a real-time simulation and display system.

Volume rendering is the process of displaying 2D composited images of a 3D scalar field. Here, the scalar field is the density. After deformation, the data are located on a curvilinear grid, then we resample the data at the points of a regular grid. We take the approach of resampling and volume rendering the regular grid instead of directly volume rendering the curvilinear grid.

With the condition of FEM linear analysis (Sec. 5.1), the displacement should be infinitesimally small. Thus, the changes in *x, y and z* coordinates of voxels are quite uniform, i.e., there are no cells of drastically different sizes. Since resampling can be performed only once with multiple rendering, the whole process is still quite efficient. In our implementation, we compute the isosurfaces as well as display integrals of density along rays by assigning density only to certain isosurfaces, and then ray trace the results (which amounts to creating different volumes to be displayed). Fig. 5 is a filtered image of the original anconeus volume data which includes 14,578 voxels, rendered by ray tracing. We also use it in Fig. 8.

Chapter 9

Results and Discussions

The anconeus muscle from VHD, as an example, is used to demonstrate our techniques. The simulation and rendering were carried out on a SGI Power Onyx/RE2, R10000, 195MHz, 640MB.

Fig. 6 and Table 1 represent reconstructed hierarchical meshes based on the segmented 3D muscle image. The meshes are constructed with 3D eight-node brick elements which are equivalent to the voxels of volume graphics. The muscle geometrical coordinates and topological information are obtained by this method. Fig. 6 show the hierarchical muscle voxel model in five different levels of detail with binary increment of the voxel size. (refer to Table 1). The meshes produced here can be used for real-time simulation and rendering. The more voxels, the more accurate the muscle image is. However, it is very slow for FEM dynamic simulation without preprocessing of the high-resolution muscle mesh. The hierarchical structure is very helpful for predicting real muscle deformation through FEM simulation with low and medium-resolution muscle meshes.

Figs. 7 and 8 show five different muscle images according to different forces applied. The bottom of the muscle (blue points) is fixed, while forces are applied at the top (yellow points) in three directions. The FEM simulation is completely volumetric so that we can see changes of the muscle shape in the various cases. It is very important to both know and view the muscle geometric change imposed by the external force when we carry out surgery simulation and cutting. The images of Fig. 7 were generated using OpenGL shaded polygons with 1,840 voxels, and those of Fig. 8 were produced by ray-casting with 14,578 voxels each.

Fig. 9 presents several snapshots of our dynamic FEM simulation rendered with OpenGL shaded polygons. The dynamic model shows the dumping process of the muscle when a sudden external force is applied at the top of the muscle (yellow points) while the bottom (blue points) is fixed at the lowest points. The muscle oscillates with decreasing frequency. The FEM simulation is carried out with 450 nodes and 219 voxels. The whole process took 0.76 seconds for 22 frames.

Chapter 10

Source Codes

10.1 Part I

The following source files can be found at:
home/fs8/deformation/users/qzhu

10.1.1 Mass-spring

C++ code for volumetrical mass-spring model for ball and other 3D objects. Modify from S. Muraki's work.

10.1.2 Image-raw

Mannual segmentation volumetric images with size: 140x220x71. The files are:
m140-220-71.raw,
m140-220-71.slc,
We have another two segmentation image as follow:
r10r2.raw: one of segmented muscle image.
a3c-71-14.raw: one segmented muscle image.

10.1.3 FEM-mesh

111, 222, 444, 888, 161616: hierarchical meshes for FEM input file;
mesh1.c, mesh2.c, mesh3.c: wireframe mesh display functions for different camera positions.

10.1.4 FEM-preprocess

Down.c: read the raw file and create volumetric dataset.
FEM-input.c: create 3D FEM inpute file from volumetric dataset.
chair.c: create FEM input file for a chair.
bar.c: create FEM input file for a bar.
eight.c: create FEM input file for a 8-node brick.
table.c: create FEM input file for a table.
muscle.raw: one of muscle segmented image.
OGLdisplay.c: OpenGL display function for this part.
material.h: OpenGL light model.
eight.inp: 8-node brick element used for FEM input file and display.
chair.inp and schair.inp: 108-node and 128-node brick element for a chair used for FEM input file and display.
bar.inp: 36-node brick element for a bar used for FEM input file and display.
big1.inp: 2665-node brick element for a large muscle dataset used for FEM input file and display.
table.inp: 138-node brick element for a table used for FEM input file and display.
muscle.inp, muscle.4, muscle.5, muscle.ren: 450-node brick element for a medium muscle datasets used for FEM input file and display.

10.1.5 FEM-postprocess

This section lists all the codes for producing the voxelization mesh based on resampling method. The SLC or RAW files then can be rendered on *VolVis* or other raycasting method.

10.1.6 Gallery-gif

This section partially lists muscle images generated using GIF, RGB forms for different resolutions.

10.1.7 FEM-modal

This section lists all the codes for modal.c: a simplified modal analysis method to create fast animation picture
modalbar, modalfhh, modalframe, modalmuscle: the execution files for fast animation picture by revoking the relevant datasets.
jacobi2.c: using Jacobi method to generate eigenvalue and eigenvector for a beam.
EX115.INP, INV-JAC.INP: sample input file for the modal analysis.
beamkm2.inp: beamk input file for modal analysis.

10.2 Part II

The following parts are located at:
home/fs2/qzhu

10.2.1 Dataset-bob

This part was created by Bob to show a dynamic picture where a chair is falling onto the floor and bouncing back again. The rendering is done by direct ray casting.

10.2.2 FEM-bar

bar.b: bar element input file.
modal.c: a simplied modal analysis code.
OpenGL1.c, OpenGL2.c, OpenGL3.c: OpenGL display function with different camera positions, floor position and mouse control. It considers the material's color.
demo2.c: FEM simulator with above display functions.

10.2.3 FEM-brick

Brick element FEM simulation for dynamic case and compared with soft and hard materials
.
The primary codes are following:

hexafnt-t3.c: dynamic FEM simulation and display.
8h, 8s, 8ss, 8sss: input file for single brick with material become softer.
33s, 44s, 55s: input files for soft material for 3x3x3, 4x4x4, 5x5x5 nodes brick ;
33h, 44h, 55h: input files for soft material for 3x3x3, 4x4x4, 5x5x5 nodes brick.

10.2.4 FEM-input

The sample input files for FEM simulation.

10.2.5 FEM-muscle

The earlier simple wireframe model for muscle static and dynamic FEM simulation.
The primary codes are following:
hexafnt.c: 3D 8-node isoparametric hexahedral finite element using frontal solver. It is used for static stress analysis.
hexafnt-t.c: add OpenGL function to hexafnt.c and FEM simulation can be displayed simutaneously.
hexafnt-t2.c: add dynamic term for static FEM simulator and display function as well.
material.h: OpenGL light model.
mesh.c: hexahedral mesh display function by OpenGL.

10.2.6 Paper

Eurographics 98 paper Qinghong Zhu with Yan Chen and Arie Kaufman.

10.2.7 Report

All reports, references, project summary for the deformation project.

Chapter 11

Conclusion

We have presented a biomechanically-based 3D FEM muscle model which allows accurate prediction of muscle deformation changes. Our reconstructed muscle mesh is based on individual patient CT/MRI volume data. We use a hierarchical voxel mesh which it is suitable for both FEM simulation and volume graphics. The physiological muscle force is considered and linear elastic muscle model for both static and dynamic cases are simulated by FEM. 3D wireframe, polygon surface and volume rendering techniques are applied to show real-time muscle deformation processes as well as realistic animation in the low and medium-resolution. The simulations and renderings of VHD anconeus muscle demonstrate an excellent performance of the new approach.

Our future research is directed towards improving the quality and speed of deformed volume rendering. We plan to parallelize the simulation and rendering. Another possibility is to use modal analysis for FEM dynamic system analysis. While the direct integration method is suitable for a smaller number of voxels, modal analysis is more appropriate for a larger number of voxels.

Chapter 12

References

Avila, R. S., Sobierajski, L. M. and Kaufman, A.,"Towards a Comprehensive Volume Visualization System," *IEEE Visualization Conference*, 1992, pp.13–20.

Bathe, Klaus-Jurgen, *Finite Element Procedures*, Prentice Hall, 1996.

Bro-Nielsen, M., and Cotin, S., "Real Time Volumetric Deformable Model for Surgery Simulation using FE and Condensation," *EUROGRAPHICS '96* Vol 15, 1996.

Barzel, R., *Physically-based Modeling for Computer Graphics, A Structure Approach*, Academic Press, Inc. 1992.

Breen, D., House, D., Wozny, M., "Predicting the Drape of Woven Cloth Using Interacting Particles," *SIGGRAPH '94*, 1994, pp. 365–372.

Celniker, G. and Gossard, D., "Deformable Curve and Surface Finite -Elements for Free-Form Shape Design," *Computer Graphics*, 1991, Vol. 25, No. 4, pp. 257–266.

Chen, D., "Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method," Ph.D dissertation, MIT,1992.

Chamberlain, B., DeRose, T., Lischinski, D., Salesin, D., Snyder, J. , "Fast Rendering of Complex Environments Using a Spatial Hierarchy," *Graphics Interface 96*, pp. 132–141.

Christopher, W., "Interactive Multiple Representation Editing of Physically-based 3D Animation," Ph.D Dissertation, University of California at Berkeley, 1995.

Cover, S., Ezquerra, N., O'Brian, J., Rowe, R., Gadacz, T., Palm, E., "Interactively Deformable Models for Surgery Simulation," *IEEE Computer Graphics and Applications*, Vol. 13, 6, 1993, pp. 68–75.

Desbrun, M., Gascuel, M., " Animating Soft Substances with Implicit Surfaces," *Computer Graphics(Proceeding SIGGRAPH)*, 1995, pp. 287–290.

Fruhauf, T., "Raycasting of Nonregularly Structured Volume Data ," *The International Journal of the Eurographics Association, Computer Graphics Forum*, 1994, pp. 293–303.

Gallagher, R. S. and Selker, P. J., "Three-dimensional Volume Visualization in Finite Element Analysis," *Mechanical Engineering*, May 1992, pp. 54–57.

Gallagher, R. S.(ed), *Computer Visualization Graphics Techniques for Scientific and Engineering Analysis*, CRC Press, 1995, pp. 61–86, pp. 171–201.

Gibson, S.F.F. et al., "Simulating Arthroscopic Knee Surgery using Volumetric Object Representations, Real-Time Volume Rendering and Haptic Feedback", *Proceedings of the First Joint Conference CVRMed-MRCA'97*

Gibson, S.F.F., "3D ChainMail: a Fast Algorithm for Deforming Volumetric Objects," MERL Tech Report TR96-22, 1996.

Gibson, S.F.F., "Beyond Volume Rendering: Visualization, Haptic Exploration, and Physical Modeling of Voxel-based Objects," *Eurographics Workshop on Visualization in Scientific Computing*, 1995, pp. 171–201.

Gourret, J.P.,Thalmann N. M. and Thalmann., "The Use of Finite Element Theory for Simulation Object and Human Body Deformations and Contacts," In *Eurographics 89*, September, 1989, pp. 477-487.

Haley, M., and Blake, E. H., "Incremental Volume Rendering Using Hierarchical Compression, " In *Eurographics 96*, Volume 15, pp. 45–55.

Heckbert, P.S. and Garland, M., "Multiresolution Modeling for Fast Rendering, " *Proceedings of Graphics Interface*, May, 1994, pp. 43–50.

Hong, L., Muraki, S., Kaufman, A., Bartz, D., He, T., "Virtual Voyage: Interactive Navigation In The Human Colon," *SIGGRAPH 97*, August, 1997.

Hohne, K.H.(ed), *3D Imaging in Medicine*, Springer-Verlag, 1990.

Hsu, W., Hughes, J., Kaufman, H., " Direct Manipulation of Free-Form Deformation," *ACM Computer Graphics*, July, 1992, pp. 176–184.

Kaufman, A., *Volume Visualization*, IEEE Computer Society Press Tutorial, 1991.

Kaufman, A., Cohen, D. and Yagel, R., "Volume Graphics," *Computer*, 27, July 1993, pp. 51–64.

Keeve, E., Girod, S., Pfeifle, P., Girod, B., "Anatomy-Based Facial Tissue Modeling Using the Finite Element Method," *IEEE Visualization 96*, pp. 21–28.

Lee, Y., Terzopoulos, D. and Waters, K., "Realistic Modeling for Facial Animation," *Computer Graphics(Proceedings SIGGRAPH)*, Vol 26, 2, pp. 309-312, 1992.

Lin, F., Seah, H. S. and Lee, Y. T., "Deformable Volumetric Model and Isosurface Exploring A New Approach for Surface Boundary Construction," *Computer and Graphics*, Vol. 20, No.1, pp.33–40.

Metaxas, D., Terzopoulos, D., " Dynamic Deformation of Solid Primitives with Constraints," *Computer Graphics(Proceeding SIGGRAPH)*, Vol 26, 2, 1992, pp. 309-312.

Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.

Pentland, A., Williams, J., "Good Vibrations:Modal Dynamics for Graphics and Animation, "*Computer Graphics*, Vol 23, 3, July, 1989, pp. 215–222.

Pieper, S., Rosen, J. and Zeltzer, D., "Interactive Graphics for plastic Surgery: A Task-level Analysis and Implementation," *1992 Symposium on Interactive 3D Graphics*, pp. 127–134.

Platt, J. Barr, A., "Constraint Methods for Flexible Models," *Computer Graphics*, Vol 22, 4, Aug., 1988, pp. 279–288.

Provot, X., "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior," *Graphics Interface*, 1995, pp. 147–154.

Promayon, E., Baconnier, P., Puech, C., " Physically-Based Deformations Constrained in Displacements and Volume," *EUROGRAPHICS*, 1996, pp. c155–c164.

Scheepers, F. et al "Anatomy-Based Modeling of the Human Musculature" *SIGGRAPH 97 Conference Proceedings*, pp. 163-172.

Sederberg, T., Parry, S., "Free-form Deformation of Solid Geometric Models," *ACM Computer Graphics*, August, 1986, pp. 151–160.

Smith, I.M. and Griffiths, D.V., "Programming the Finite Element Method," *John Wiley & Sons Ltd*, 1988.

Speke, T. and Christon, M.A., "Visualization of High Resolution 3-D Nonlinear Finite Element Analyses, " *IEEE Visualization Conference*, 1992, pp. 324-332.

Stern, Jack T., "Computer Modeling of Gross Muscle Dynamics," *Journal of Biomechanics*, 1974, Vol. 7, pp. 411-428.

Szeliski, R., Tonnesen, D., "Surface Modeling with Oriented Particle Systems," *Computer Graphics*, Vol 26, 2, July, 1992, pp. 185-194.

Takanashi, I., Muraki, S. and Doi, A. and Kaufman, A., " Three-dimensional Active Net for Volume Extraction," to be presented at *SPIE Electronic Imaging '98* , 3298, January 1998.

Tabatabai, B, et al, "Volume Rendering on Non-regular Grids," *The International Journal of the Eurographics Association, Computer Graphics Forum*, 1994, pp. 247–258.

Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K., "Elastically Deformable Models," *Computer Graphics* 21, July, 1987, pp. 205–214.

Terzopoulos, D., Fleischer, K., “Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture ,” *Computer Graphics* 22, August, 1988, pp. 269-228.

Terzopoulos, D., Waters, K., “Physically-based Facial Modeling, Analysis and Animation,” *Journal of Visualization and Computer Animation*, 1990, pp. 73–80.

Terasawa, M. and Kimura, F., “Collision Response for Deformation Models Based on the Hertz’s Contact Theory,” *Proceedings of the First Pacific Conference on Computer Graphics and Application*, World Scientific, S.Y. Shin et al (Ed), 1993. pp. 45–60.

Tonnesen, D., “Modeling Liquids and Solid Using Thermal Particles,” *Graphics Interface*, 1991, pp. 255–262.

Troutman, R., and Max, N. L., “Radiosity Algorithms Using Higher Order Finite Element Methods, ” *Computer Graphics Annual Conference Series*, 1993, pp. 209–212.

Waters, K., “A Muscle Model for Animating Three-dimensional Facial Expression, ” *Computer Graphics*, Vol. 21, 4, July, 1987, pp. 17–24.

Wilhelms, J. and Gelder, A., “Anatomically Based Modeling,” *SIGGRAPH 97 Conference Proceedings*, pp. 173-180.

Witkin, A. et al, *An Introduction to Physically Based Modeling*, Course Notes, 1993.

Zienkiewicz, O.C., *The Finite Element Method in Engineering Science*, Third Edition, McGraw-Hill, London, 1977.

Zhu, Q., Chen, Y., and Kaufman, A., "Real-time Biomechanically-based Muscle Volume Deformation using FEM, " Submitted to *Eurographics 98* .

Chapter 13

Appendix