AN ACCURATE AND DYNAMIC COMPUTER GRAPHICS MUSCLE MODEL

by

David Asher Levine, B.S.

THESIS

Presented to the Faculty of the Graduate School
of The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of
MASTER OF SCIENCE IN ENGINEERING

The University of Texas at Austin
August 1997

AN ACCURATE AND DYNAMIC COMPUTER GRAPHICS MUSCLE MODEL

APPROVED BY

SUPERVISING COMMITTEE:

Ronald E. Barr

Ronald E. Barr

Lawrence D Abraham

Lawrence D. Abraham

Dedicated to my beloved parents

ACKNOWLEDGMENTS

I owe a great thanks to Dr. Ron Barr for his wise leadership and project management; to Brian Garner for his patience, command of program, and gifts of creativity and visualization; to Dr. Larry Abraham for taking time out of his personal vacation time to read and edit this thesis; to Dr. Marcus Pandy for his assistance with parallel research; Jim Maida and NASA for their financial support; to the faculty and my peers in the Department of Mechanical Engineering and the Department of Biomedical Engineering for their encouragement and help; and most importantly to my family for their love and support.

D.A.L.

The University of Texas at Austin
August, 1997

AN ACCURATE AND DYNAMIC COMPUTER GRAPHICS MUSCLE MODEL

by

David Asher Levine, M.S.E.
The University of Texas at Austin, 1997
SUPERVISOR: Ronald E. Barr

A computer based musculo-skeletal model was developed at the University in the departments of Mechanical and Biomedical Engineering. This model accurately represents human shoulder kinematics. The result of this model is the graphical display of bones moving through an appropriate range of motion based on inputs of EMGs and external forces. The need existed to incorporate a geometric muscle model in the larger musculo-skeletal model. Previous muscle models did not accurately represent muscle geometries, nor did they account for the kinematics of tendons. This thesis covers the creation of a new muscle model for use in the above musculo-skeletal model.

This muscle model was based on anatomical data from the Visible Human Project (VHP) cadaver study. Two-dimensional digital images from the VHP were analyzed and reconstructed to recreate the three-dimensional muscle geometries. The recreated geometries were smoothed, reduced, and sliced to form data files defining the surfaces of each muscle. The muscle modeling function opened these files during run-time and recreated the muscle surface. The modeling function applied constant volume limitations to the muscle and constant geomtry limitations to the tendons.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction and Literature Review

1.1. Project Background

1.1.1. NASA Project

1.1.1.1. Introduction to NASA Project

The growing power of computer graphics is influencing a wide range of fields including engineering, medicine, and entertainment. Graphical modeling of human movement is of particular interest in these areas. The exponential development of computer hardware in the 1980s and 1990s, and the availability of digitized anatomical data, has created an environment in which more details may be applied to computer-based models of human movement.

In 1992 NASA's Johnson Space Center issued a Regional University Grant to the University of Texas at Austin. The purpose of this grant is to create a musculo-skeletal model of the human upper extremity with application to manned space operations. This model must accurately portray human muscle forces and joint movements. The creation of this model has been broken into three segments: theoretical and computational model creation, experimental validation of the theoretical model, and graphical support for the model.

1.1.1.2. Modeling

The computational modeling is based on optimal control solutions to a desired input motion. The model's output will be an accurate computer graphics rendering of the shoulder's musculo-skeletal system moving as an actual musculo-skeletal system would for the selected motion. To create the model, muscle and bone geometries must be measured, muscle forces and their lines of action must be known, and muscle activation time-histories must be determined for given motions. To create the activation curves, a large-scale optimal control algorithm was employed [1,2]. Once values for all of the above are found, the result is a dynamic computer graphics model, typically for movements of short durations.

1.1.1.3. Experimentation

Data were collected to verify the models. Experiments on human subjects were performed to create data sets. These experiments included both isometric and dynamic trials. During the isometric trials, subjects' torsos were constrained as they gave 100 percent exertion for contractions of different muscle groups at different joint angles. A dynamometer measured the net joint torque of these contractions. The dynamic trials used the same subjects. During the dynamic trials, the research team tracked the motion of the subjects' limbs and recorded relevant muscle EMG activation levels during prescribed arm motions. The experimental data were analyzed to solve for net muscle forces and muscle activation levels during movements that had been modeled.

1.1.1.4. Graphical Support

The last step of the project is the computer graphics support. This support includes accurate geometric representation of bones and muscles. This thesis discusses the method by which the graphical muscle models were created. Many of the same computer programs used for bone modeling were employed for the creation of muscle models.

1.1.2 Visible Human Project

1.1.2.1. Introduction to the Visible Human Project

The National Library of Medicine decided in its 1986 long-range plan to make a complete set of anatomically detailed digital images of both the male and female bodies available to researchers. In August 1991 the National Library of Medicine awarded a contract for data acquisition to the University of Colorado at Denver [3].

The data acquisition team began by procuring the body of a 38 year old male convicted murderer who had donated his body to science [4]. The subject was five feet and eleven inches tall, weighed 199 pounds, and was described as having good to excellent general health [5]. The data acquisition team took magnetic resonance images (MRIs) and transverse computed tomography (CT) scans of the fresh cadaver at 4 mm and 1 mm intervals, respectively. The cadaver was then embedded in a gelatin and frozen. The data acquisition team took the transverse CT scans again now that the cadaver was frozen. The team then actually transversely sliced the cadaver in 1 mm intervals and snapped color axial photographs of the remaining cadaver. The team recorded 1871 cross-sectional color photographs and CT scans. The photographs were later digitized and all of the images were placed on a National Library of Medicine server [3].

1.1.2.2. Image Resolution

The Visible Human Male data set images vary in resolution. The MRIs are 256 pixels by 256 pixels with 12 bit grey scale resolution. The CT scans have a better resolution than the MRIs at 512 pixels by 512 pixels, but share the 12 bit grey scale resolution. More importantly, the radius of the CT scans was too small to capture the entire width of the cadaver. In the CT scans, the widest part of the cadaver, around its elbows, is cut off by the border of the scan itself. A CT scan is shown in Figure 1.1. The digitized color images are 2048 pixels by 2048 pixels and are defined by 24 bit color [3]. Each pixel has a resolution of 0.33 mm [6]. A color image is shown in Figure 1.2. The complete data set is 15 gigabytes [3].

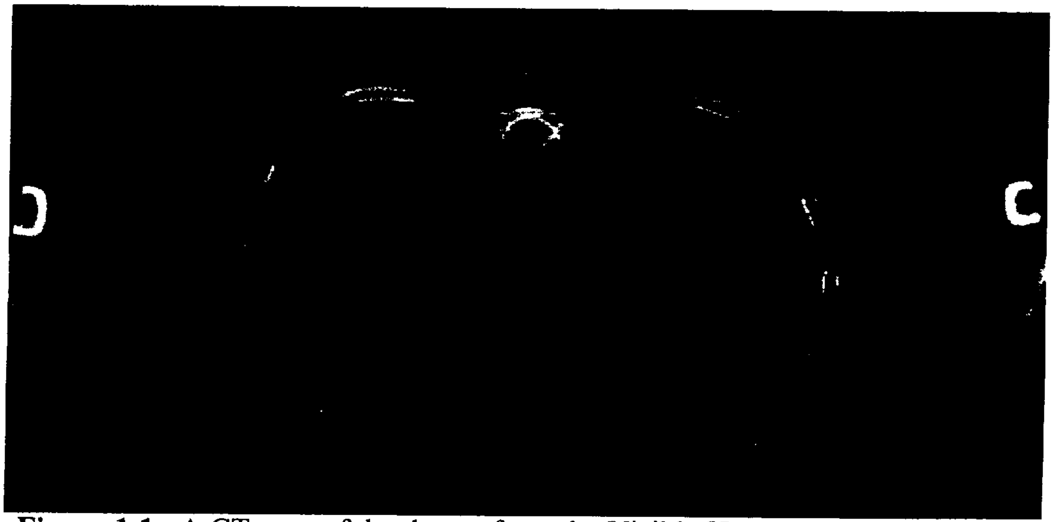

Figure 1.1. A CT scan of the thorax from the Visible Human Male data set [7]. Note that the elbows are cut off by the radius of the CT scan.

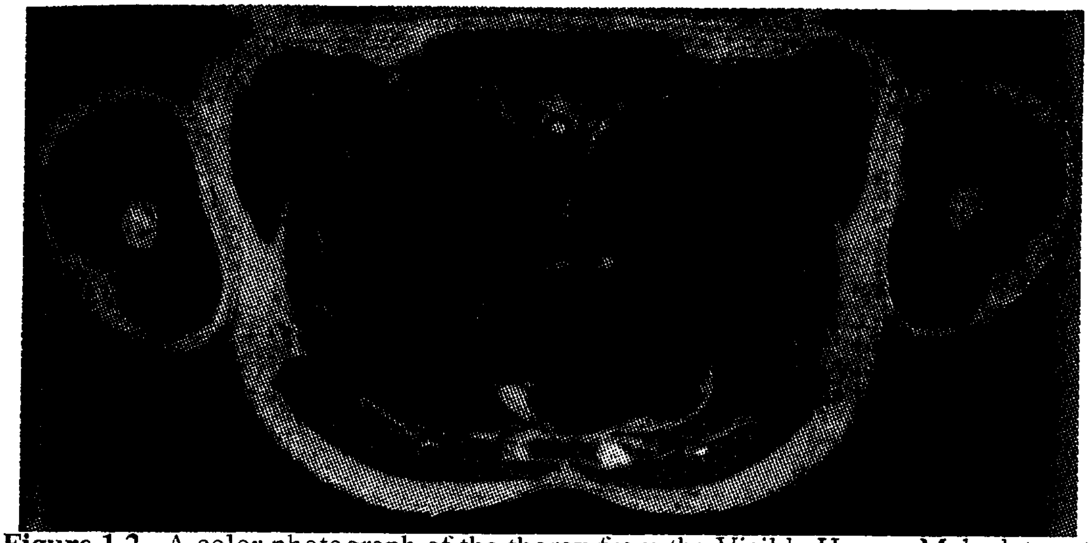

Figure 1.2. A color photograph of the thorax from the Visible Human Male data set [8].

1.2. Literature Review

1.2.1. Anatomy

The purpose of this research is to create accurate gross anatomical reconstruction of the shoulder muscles from the Visible Human Project images. To accomplish this goal, familiarity with the gross anatomy of the shoulder muscles is required. The dynamic model involves the motion of the clavicle, scapula, humerus, and wrist (including the radius, ulna, and several bones of the palm). All muscles originating or inserting on the major bones, and having a significant impact on the motion of these bones, are included. A list of those muscles is in Table 1.1 with their origin and insertion locations. A list of relevant muscles that attach to the clavicle, scapula, or humerus that were not dynamically modeled is shown in Table 1.2. The muscles listed in Table 1.2 were measured and decimated, but they were not sliced or dynamically modeled. Anatomy reference books provided guidance for tracking the morphology of the individual muscles [9,10].

Mammals' skeletal muscles all share similar morphology. Most skeletal muscles begin and end in tendons. These muscles are made of fibers. Each fiber is a long, cylindrical, multinucleated cell surrounded by a membrane. Muscle fibers are

Table 1.1
Muscles Modeled [10,11]

Muscle	Origin	Insertion
Biceps brachii	Scapula	Radius
Brachialis	Humerus	Ulna
Brachioradialis	Humerus	Radius
Coracobrachialis	Scapula	Humerus
Deltoid	Clavicle, scapula	Humerus
Extensor carpi radialis brevis	Humerus	3rd metacarpal
Extensor carpi radialis longus	Humerus	2nd metacarpal
Extensor carpi ulnaris	Humerus	5th metacarpal
Flexor carpi radialis	Humerus	2nd and 3rd metacarpals
Flexor carpi ulnaris	Humerus, ulna	Pisiform, hamate, 5th metacarpal
Infraspinatus	Scapula	Humerus
Latissimus dorsi	Scapula, thoracic and lumbar vertebrae, 10th, 11th, and 12th ribs, pelvis	Humerus
Levator scapulae	Scapula	1st through 4th cervical vertebrae
Palmaris longus	Humerus	Palmar aponeurosis
Pectoralis major	Clavicle, sternum, anterior sheath of rectus abdominus	Humerus
Pectoralis minor	3rd, 4th, and 5th ribs	Scapula
Pronator teres	Humerus, ulna	Radius
Rhomboideus major	2nd through 5th thoracic vertebrae	Scapula
Rhomboideus minor	7th cervical and 1st thoracic vertebrae	Scapula
Serratus anterior	1st through 9th ribs	Scapula
Subscapularis	Scapula	Humerus
Supraspinatus	Scapula	Humerus
Teres major	Scapula	Humerus
Teres minor	Scapula	Humerus
Trapezius	Skull, cervical and thoracic vertebrae	Clavicle, scapula
Triceps brachii	Scapula, humerus	Ulna

themselves a congregation of small bundles known as myofibrils. The myofibrils are composed of thick (myosin) and thin (actin) filaments. The filaments are aligned parallel to the muscle's line of action. This alignment causes the bundles' forces of contraction to be additive. The myosin and actin filaments are drawn on top of one another during muscle contraction. As the filaments move, the muscle shortens and

Table 1.2
Muscles Not Modeled that Attach to Relevant Bones [10,11]

Muscle	Origin	Insertion
Anconeus	Humerus	Ulna
Omohyoid	Scapula	Hyoid
Sternocleidomastoid	Skull	Clavicle, sternum
Sternohyoid	Hyoid	Clavicle, sternum
Subclavius	1st rib	Clavicle

widens relative to the filaments. Muscles also act as reservoirs for blood during muscle contraction [12]. Knowledge of muscle morphology and physiology is necessary for accurately mimicking muscle geometries during contraction and extension.

1.2.2. Prior Graphics Modeling

Several prior computer graphics models of human skeletal muscles have been created. Most have occurred since the Visible Human data became public on November 1, 1994 because of the ease of attaining that data [13]. Prior muscle models include numerous static models and a smaller assortment of dynamic models that respond to displacement of the insertion and origin points. These models' applications vary from determining lines of muscle action in precise biomechanics studies to pure artwork for movies.

1.2.2.1. Static Muscle Models

Numerous static muscle images have been created from cadaver data. The earliest muscle model based on Visible Human data was from the Visualization Group of the Scientific Computing Division at the National Center for Atmospheric Research (NCAR) in Boulder, Colorado. NCAR assisted the Visible Human project by restacking the photographic images and defining the three-dimensional data in terms of voxels or volumetric pixels. Each one of the voxels was then given an electron density value and red, green, and blue color values. An image of the knee as reconstructed by NCAR's Visualization Group is shown in Figure 1.3. NCAR made their model more adaptable for virtual surgery by defining it in voxels instead of by surface points. Using voxels allows the model to be cut and to show color attributes under the surface as shown in Figure 1.4. Applications for NCAR's model span from virtual surgery to identifying anatomical structures [13].

Figure 1.3. The human knee reconstructed by NCAR from Visible Human data [14].

The Institute for Mathematics and Computer Science in Medicine (IMDM) at the University Hospital in Hamburg-Eppendorf, Germany created a detailed static atlas of the human called Voxel-Man. The Voxel-Man atlas began before the Visible Human data were available, and therefore used only radiological data of the brain and skull. The Voxel-Man atlas contrasted many other anatomical atlases of its time by providing full three-dimensional models of gross anatomy, as opposed to two-dimensional slices, and used voxels to allow for surgical simulations. When the Visible Human data became available to the public, IMDM imported the Visible Human data into the Voxel-Man framework. The Voxel-Man atlas increases surface smoothness and realism by anti-aliasing. Anti-aliasing determines borders based on color intensity of the pixel data [16,17,18,19,20]. An image of the Voxel-Man shoulder is shown in Figure 1.5.

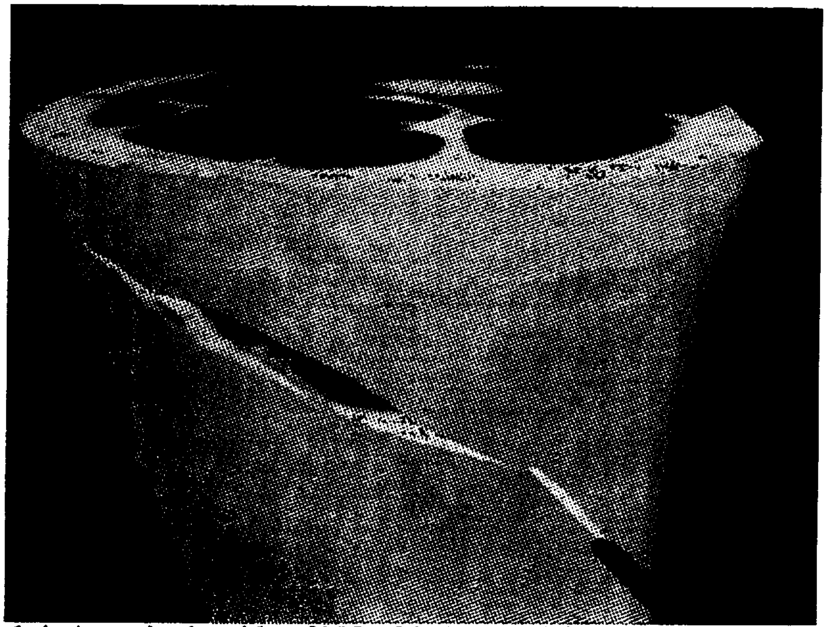

Figure 1.4. A cut in the side of NCAR's leg model displays the advantage of modeling with voxels. The knife can be seen in the lower right [15].

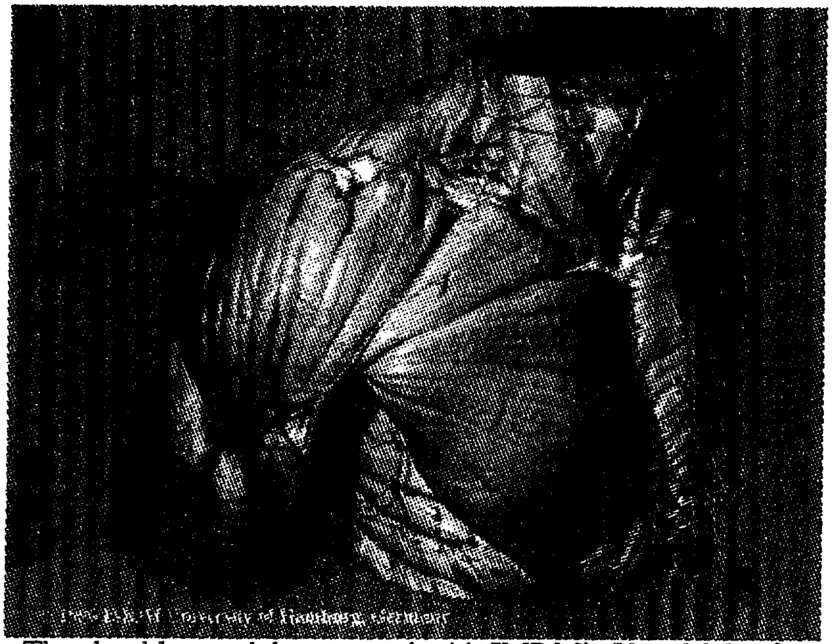

Figure 1.5. The shoulder model as created with IMDM's Voxel-Man framework and Visible Human data [21].

General Electric's Computer Graphics and Systems Program (CGSP) is a division that works on computer graphics for applications for aerospace, mechanical, business, and medical applications. CGSP tested their algorithms for constructing three-dimensional models on the Visible Human data [4]. Many of the algorithms used by CGSP were also used for static muscular reconstruction and decimation within this research and are discussed in more detail in Chapters 2 and 3. The model of the thigh muscles created by CGSP is shown in Figure 1.6.

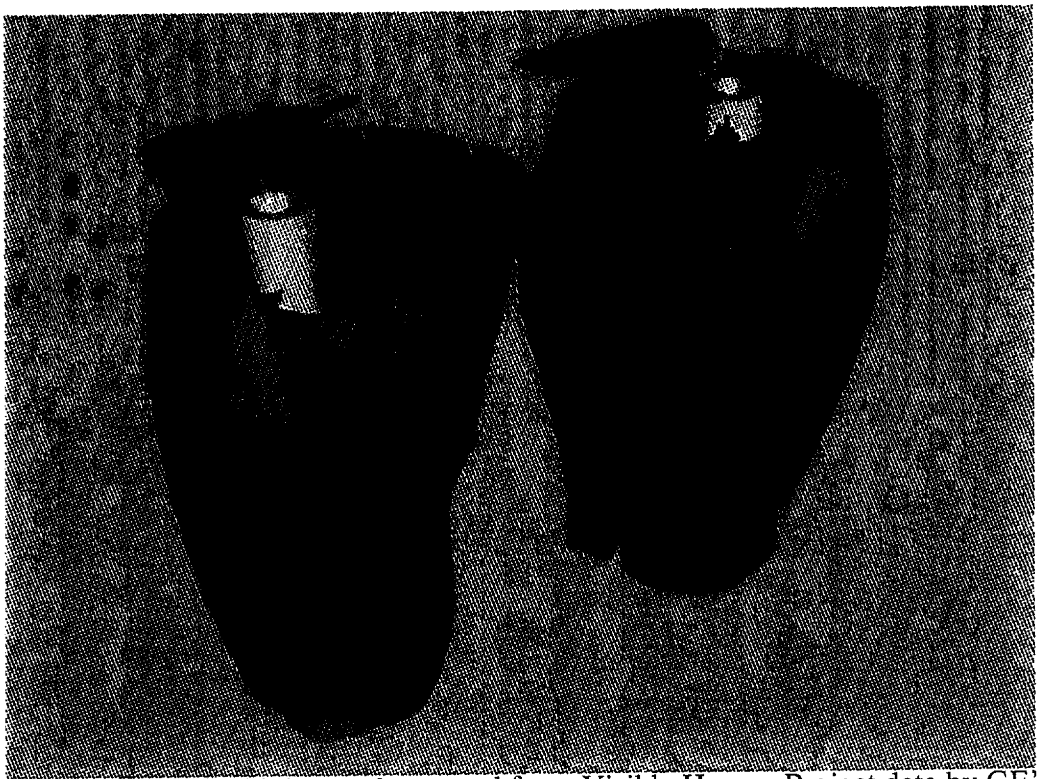

Figure 1.6. Static thigh muscle created from Visible Human Project data by GE's CGSP [22].

1.2.2.2. Dynamic Muscle Models

A research team at the University of Utah modeled muscle geometries to determine muscle forces and lines of action. The initial geometries were based on optical measurements of grids on muscle surfaces. The computer-generated muscles react to the skeletal movement, but fail to simulate volumetric changes or show tendons [23].

At the University of Toronto, a muscle model was created to aid the study of animal movement control. The muscle geometries are modeled as generalized cylinders. The geometries have circular cross-sections throughout, but can wrap around objects accurately. The muscle geometries bulge when contracted, but do not follow any anatomical rules when doing so. The primary purpose of this model is to provide a fast visualization of the musculotendon units for real-time viewing [24].

The Human Animation Research and Development project (HARD) began in 1994 at the computer graphics development house, Digital Domain (Venice, California), to simulate a graphical human being. The details of the model are proprietary, but the muscles are fashioned from high-resolution polygonal tubes. The muscles bulge in reaction to contraction, but tendons were not represented. The muscle geometries are based on observations and aesthetic judgment. This model clearly has the best appearance of the dynamic muscle models. It was even used to create entire computer graphics generated humanoids ("mice men") for the movie *The Island of Dr. Moreau* [25,26,27]. Images from the HARD model are shown in Figure 1.7.

Figure 1.7. Three views of the HARD arm muscles.

1.2.2.3. Graphics Modeling at the University of Texas

Prior work at the University of Texas was performed by Wood [28] and Marin [29]. Both researchers modeled muscles of the elbow joint as linear bodies with circular cross-sections. Wood originated the muscle model with circular cross-sections and a splined surface. Marin's model added bulge factors during contraction. Both models are applied to a dynamic bone model [28,29]. These models are shown in Figures 1.8 and 1.9. This thesis continues the work on their research.

Figure 1.8. Images of Wood's prior muscle model alone (left), and on a skeletal model (right) [28].

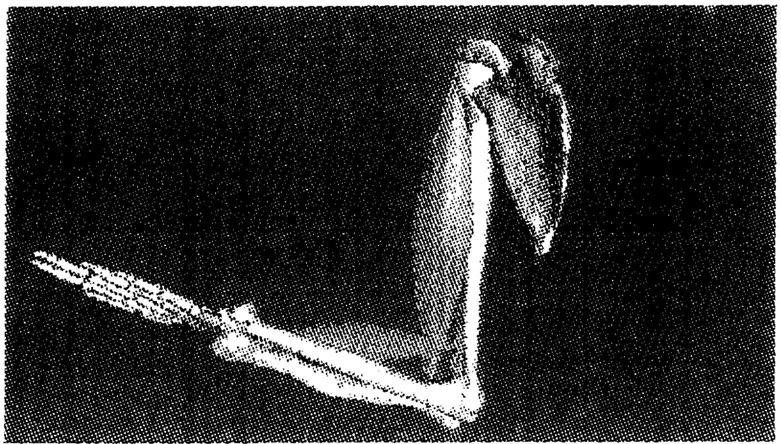

Figure 1.9. An image of Marin's prior muscle model at the University of Texas at Austin [29].

1.3. Research Objectives and Hypothesis

The static muscle models listed previously are based on accurate muscle geometries, but their geometries do not respond to movement of the skeleton. The dynamic muscle models listed above can alter their geometries, but fail to simulate accurate muscle geometries. The purpose of the present muscle model is to create a dynamic muscle model that is based on accurate muscle measurements and that bulges with flexion based on objective physical rules.

The proposed geometric muscle model is created by the methodology described within this thesis. In short, this methodology consists of measuring muscle geometries from the Visible Human Project data set, and altering these geometries based on inputs of anatomical reference dimensions over time. These dimensions include the locations of the origin and insertion bones for each muscle and obstacle locations. Through this methodology, a realistic and dynamic muscle model is created.

2. Muscle Measuring Methodology

2.1. Objective of Muscle Measuring

The construction of anatomically accurate muscle geometries begins by founding the geometries on measurements of actual muscles. Large numbers of points on the muscles' surfaces were measured from color photographs of transverse slices. The large number of points also allowed a realistic surface reconstruction to be performed. The steps of the methodology used to measure the initial muscle geometries are listed as follows.

2.2. Software and Hardware Tools for Muscle Measuring

2.2.1. Visible Man Project Data

The resources that are needed to examine an individual cadaver and record muscle geometries were beyond the scope of this research. Due to the existence of the Visible Human Project [3], a cadaver study was not needed. Images from the male Visible Human Project subject were used to attain muscle surface geometries. A license agreement with the National Library of Medicine provided permission to use the images.

2.2.2. "ftp" and "uncompress" Functions in IRIX

To begin the muscle measuring, the images needed to be transferred from the National Library of Medicine's server to the workstation where the muscle measuring would be carried out. The workstation where the measuring is performed runs from the IRIX operating system (IRIX is Silicon Graphics' version of UNIX). IRIX contains a function, "ftp" (file transfer protocol), that transfers files over a network. "ftp" was used to transfer the image files from the National Library of Medicine's server to the workstation where the measuring was done.

The National Library of Medicine compressed these image files (denoted by a ".Z" after the file name) to conserve memory on their server and to speed the transfer time of each file. These files were uncompressed with another IRIX function, "uncompress".

2.2.3. Shrinking Image Files

Due to the size of the color images and the amount of memory available on the workstation (see Appendix A), the size of the image files was reduced after the image was transferred to the workstation. This file size reduction occurred by passing the image files through a shrinking algorithm.

The shrinking algorithm reduced the size of the file in two ways. The shrinking algorithm retained pixels whose x and y values were both even. The algorithm also saved the colors of the pixels in a more efficient method than the original National Library of Medicine format. The size of the color images varied from 4.5 to 5.2 megabytes in its compressed, ".Z", format. Uncompressed and ready to use, the color images were roughly 7.5 megabytes each. Each color slice was reduced from its uncompressed size to 161 kilobytes after application of the shrinking algorithm.

After initial tests, only slices at every fourth millimeter were retained, thus reducing memory usage by three-fourths. More importantly, this reduction increased muscle smoothness by eliminating the effect of graphical anomalies from intermediate slices.

2.2.4 CT Image Editing

A first attempt to measure muscles was made with the CT images. A CT image is shown in Figure 1.1. The attempt using CT images was originally performed to save disk space. The lower resolution of the CT images, in both the number of pixels and the variety of coloring, made them a preferred mode for retrieving muscle geometries. The CT image editing algorithm was similar to the color photograph editing algorithm described later in Section 2.2.5. The primary difference between the two algorithms is the CT image editor's use of grey scale values to differentiate between pixels, and the color image editor's use of red, green, and blue (RGB) intensity values.

The CT image editor was originally written to measure the bones in the body. Inspection of the image in Figure 1.1 shows the bones are clearly discernible from the surrounding tissue. CT resolution relies partially on water content of the tissues being scanned. Because bone has a lower concentration of water than its surrounding tissue, it is clearly visible. However, after using the CT images, it became apparent that the poor resolution between adjacent muscles, and between muscles and the fat surrounding the muscles made it impossible to discern a distinguishing line of the muscles. The decision was made to abandon the use of CT images, and try to measure the muscles with the color photographs.

2.2.5 Color Cryogenic Image Editing

After the CT images proved unsuitable for reconstructing muscle surfaces, the color photographs of the frozen cadaver were employed. To create a three-dimensional surface structure from a series of consecutive two-dimensional images, outlines of the muscles were defined on the two-dimensional images and then stacked on top of each other and connected to form the three-dimensional surface. Outlining the muscles on the two-dimensional color photographs was accomplished using software created at the University of Texas [30].

Editing the two-dimensional images to generate outlines of a specific muscle occurred in a three step process: 1. referencing anatomy books, 2. rough outlining, and 3. refining the outline. In the first step, the muscle was identified by referencing anatomy books [9,10] and scanning through superior and inferior slice images. This proved useful for isolating each individual muscle on the transverse slices.

In the next step, the muscle was roughly outlined. This rough outlining was carried out by selecting a point inside of the desired muscle. The software calculated a line traveling anterior to the point. When the line first intersected a point outside of the given RGB threshold the software recorded this point as the first point on the outline, and tracked the pixels that form the outline to the right of the first outline point based on this border between pixels with acceptable and unacceptable RGB values. The acceptable red, green, and blue threshold values, although variable, were all set from 0 to 50 on a 0 to 255 scale for these measurements. This pixel thresholding method is similar to that used by Lorensen [4] for the same application. An example of rough outlining is shown in Figure 2.1.

Figure 2.1. The rough editor is shown on a color image. A line is drawn superior to the transverse slice (blue). The first point which crosses the threshold value is shown on the outline (yellow). The outline of the pectoralis minor then follows the border between pixels with acceptable and unacceptable RGB values (green) to the right of the first point.

Even though the color images provided better definition than the CT scans, the defining line between some muscles is still indistinguishable to the rough editor. Hence, the third step in the muscle editing process is refining the rough outline. The software allowed muscle outlines to be manipulated to correct errors. The outlines were manipulated to separate contiguous muscles that were both outlined by the rough editor, or to remove anatomical features, such as blood vessels, that fell within the RGB threshold. The manipulation also allowed for the inclusion of lighter colored tendon in the outline.

The locations of the outlines were saved in separate files than the color images. This allowed the color images to be used repeatedly and helped to conserve memory. Saving the outlines separately also provided for easy reconstruction of the muscle [30].

2.2.6 Muscle Reconstruction

The outlines of the two-dimensional muscle images were stacked on top of one another and the pixels were connected to form a three-dimensional muscle [31]. The method used to reconstruct the muscles is the *marching cubes* algorithm [32]. This algorithm produced high resolution three-dimensional surfaces from two-dimensional slice data, exactly what was needed when reconstructing these muscles.

Marching cubes reconstructs the three dimensional surface in small, easy-to-handle steps, one cube at a time. That cube is made of the eight pixels of its corners. Four of these pixels form a square on one image slice, and the other four form a square at the same two-dimensional coordinates, but on the next slice down.

Now that the cube has been defined, the algorithm checks the pixels to determine if they are, or are not on the outline defined during editing. Because there are eight corners to the cube, each of which may be either on the outline or not (two cases), there are 2^8 (256) possible combinations of corner definitions. Symmetries of the cube reduce the number of possible combinations from 256 to 14 unique, non-symmetric cases.

Graphical surfaces have long been defined with polygons, particularly triangles. Triangle orientations have been defined for the 14 unique cases. All the algorithm must do is look up the necessary triangles for each cube, speeding reconstruction. The trivial cases exist when all eight corners of the cube are either part of the outline, or all eight corners are not part of the outline. In these cases, no triangles are defined. Two of the other 13 cases are shown in Figure 2.2 as an example of the remaining cases. After defining the triangles for the particular cube being analyzed, the algorithm *marches* to the next cube, and so on until all of the data has been analyzed and the entire object has been reconstructed.

The result of using the *marching cubes* algorithm was an effective reconstruction of three-dimensional surface data. The biggest disadvantage of the reconstruction [32] was that it left ridges between slices as shown in Figure 2.3.

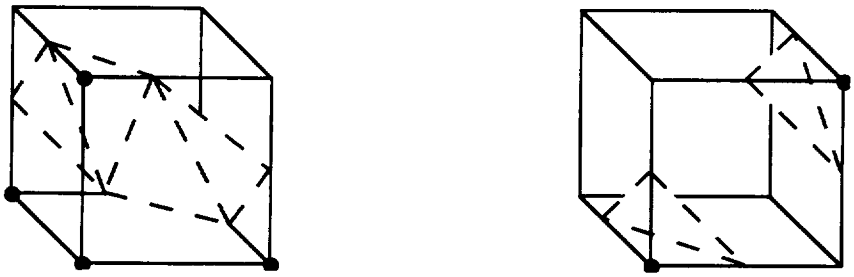

Figure 2.2. Two examples of the 14 possible combinations of triangles with the *marchingcubes* algorithm. Each corner ofthe cube represents one pixel. The corners with points represent pixels that are on (or not on) the outline of the muscle, whereas the other corners are not on (orare on) the outlineof the muscle [32].

2.2.7Muscle Smoothing

The *marching cubes* algorithm produced a high resolution, three-dimensional surface. The ridges that thereconstruction generated betweenslices are natural for the algorithmwith the given resolution and level of noise, but unnatural for actual muscle tissue. To reduce this artificial roughness, each reconstructed muscle surface was passed through a smoothing algorithm[33].

The smoothing algorithm was originally created at the University for the purpose of smoothing bones. The algorithm adjusts the coordinates of each point to coincidewith a three-dimensionalpolynomial surface of the surrounding points within a patch radius. To ensure consistency, all smoothed points were calculatedbefore any of the smoothed points replace the unsmoothed points [34]. Each muscle was passed through the smoothing algorithmthree times. The patch radii usedin these passes were 3.0 mm, 3.5 mm, and 4.0 mm, respectively. The result ofmuscle smoothing is shown in Figure 2.3.

2.3 Integration of Tools

In order to measure muscles from the Visible Human Project color images, eight steps were performed. First the image files were transferred to the workstation from the National Library of Medicine. Then the images were uncompressed and shrunk. Next, image editing occurred for each muscle in three steps: identifying the muscle to be measured, getting a rough outline of that muscle, and manually refining the outline. The latter two editing steps were repeated for each slice that contained the given muscle. The two-dimensional outlines were then reconstructed into three-dimensional muscle models, and the three-dimensional models were smoothed to finish the process. This methodology is shown in Figure 2.4. The results of the muscle measuring for all of the muscles involved are shown in Appendix B.

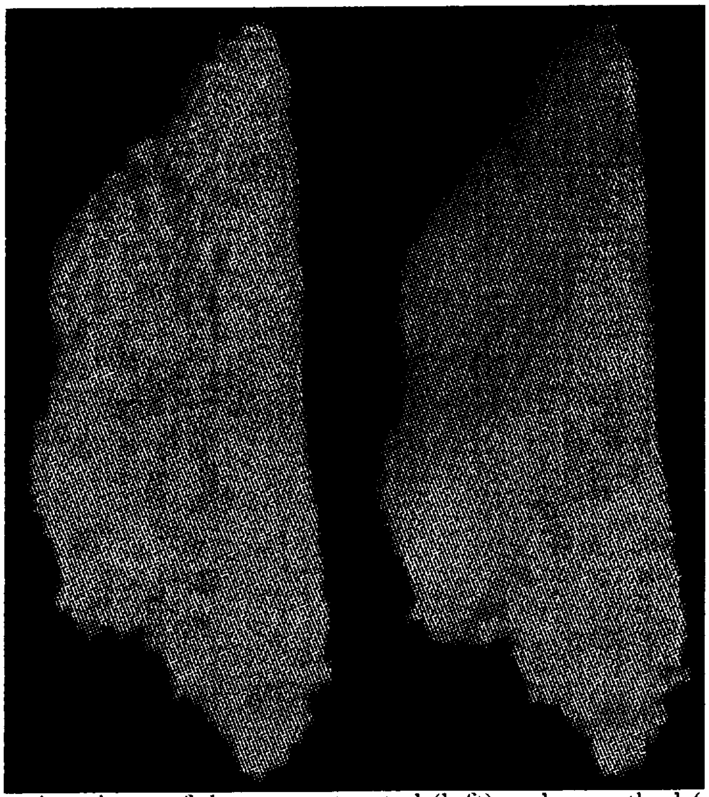

Figure 2.3. Anterior views of the reconstructed (left) and smoothed (right) pectoralis minor.

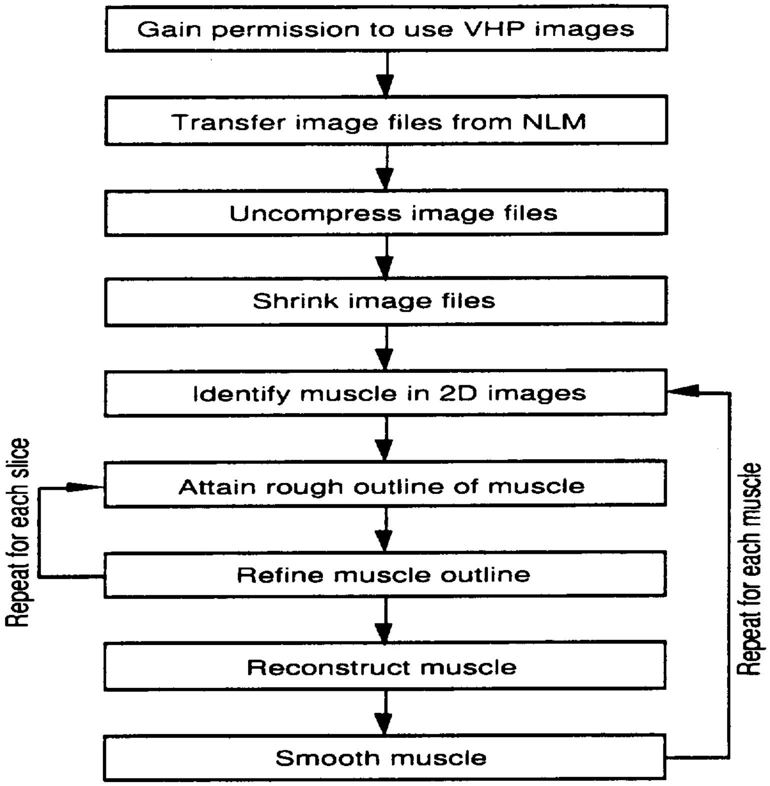

Figure 2.4. Flow chart of muscle measuring methodology.

3. Muscle Data Reduction Methodology

3.1. Data Reduction Reasoning

In their reconstructed state, the muscles were defined by large sets of geometrically unorganized vertices. These sets of vertices slowed screen refresh and were not conducive to a structured method to bulge the muscles. Primarily for these two reasons, the data sets that composed the muscle surfaces were reduced to a decimated and organized set of vertices.

The original method for decimation included a four step process. First, the muscles were decimated as described in Section 3.2. Second, several points were manually selected from the muscle surface with the aid of a mobile cursor existing in the graphics software package [35]. Third, these points were run through a normalizing algorithm to ensure that they lay in a plane normal to the particular muscle's line of action [36]. Fourth, these points would be inserted into a computer-aided design (CAD) program expressly made for creating muscle surfaces [37]. The normalizing algorithm and CAD program were facilitated, and several trials were attempted to decimate the muscles in the above manner. With not much success, this data reduction methodology was discarded and a new methodology was employed. This methodology increased accuracy of the decimated muscle surfaces and the automation of decimation tasks.

3.2. Muscle Decimation

The reconstructed and smoothed muscle models were checked for anomalies to ensure that the slice editing was performed correctly. For this reason, the smoothed muscle models must be translated and rotated on the screen at reasonable speeds. Several of the smoothed muscle files were over 1.6 megabytes large. Examining large muscle models becomes very non-interactive when the screen takes over twenty seconds to refresh after a single translation or rotation of the muscle model. The solution to this problem was to reduce the number of triangles and vertices that define the muscles' surfaces.

A decimation algorithm was selected that had been used on similar three-dimensional *marching cubes* reconstructions [32]. Because muscle surfaces are continuous, non-ending, and do not intersect themselves (like three-dimensional, Cartesian x-y and x-z planes intersect each other), the decimation is simplified tremendously. The algorithm traveled from vertex to vertex, determining if each vertex fit within a set of decimation criteria. For each vertex, the algorithm computed a plane from all of the triangles that used that vertex as a corner. The plane's location was based on a weighted average of the triangles' centerpoint coordinates. The weighting was based on each triangle's area. The plane's surface normal was also based on a similarly weighted average of the triangles' surface normals. The distance to this new plane was then calculated. If the distance from the vertex to the plane was below a set tolerance, the point was removed. The gap that was left was retriangulated using the neighboring points as new corners of the replacement triangles. This decimation removed points which added little detail to the topology of the muscle [38].

Two sets of decimated muscles were created, a high-resolution set, and a low-resolution set. Each muscle was passed through the decimation algorithm three times. The high-resolution set was decimated with a tolerance of 0.1 mm. The low-resolution set was decimated with a tolerance of 0.25 mm. The low-resolution muscles had very few visible differences from the high-resolution muscles. After reviewing the results, only the low-resolution muscles were used and will be discussed from this point on. The high-resolution muscles were saved for possible future research. The low-resolution pectoralis minor is shown in Figure 3.1. All of the decimated muscles are shown in Appendix B.

The decimation algorithm works well for smooth muscle surfaces. The highest level of decimation occurred on the latissimus dorsi. This was most likely due to the latissimus dorsi's large anterior and posterior sides that have little curvature, and few protrusions. After decimation, its file was reduced from 1,616 kilobytes to 76 kilobytes, 4.7 percent of its original smoothed size. On the average, the decimation algorithm reduced the smoothed muscle files to 8.7 percent of their original size.

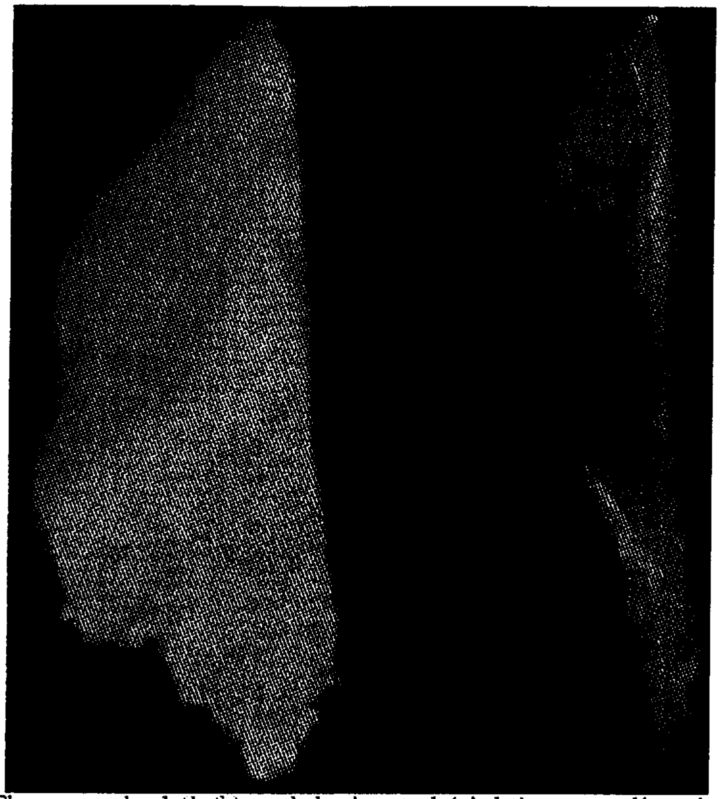

Figure 3.1. The smoothed (left) and decimated (right) pectoralis minor. Note that the decimated model has different lighting and color settings.

3.3. Muscle Slicing

At this point the data sets defining the muscles were small enough to be transformed with reasonable refresh rates. However, the data were not organized in a manner that was conducive to dynamic modeling the muscles as described later in Chapter 4. The muscles needed to be reorganized, so they were redefined by vertices that lie on planes perpendicular to their lines of action.

Software was developed at the University of Texas to find muscles' lines of action for the dynamic model [39]. The software was extended to find the points that lie on planes perpendicular to these lines of action. The software allowed the manual manipulation of a visible plane in the graphical space containing the muscle. The user positioned the plane to intersect the muscle perpendicular to the muscle's line of action. The software recorded the plane's location and orientation, and a slice was created. The process was then repeated for each muscle. The number of slices taken for each muscle is subjectively based on the muscle's topology, size, curvature of the line of action. Muscle slicing is shown in Figure 3.2.

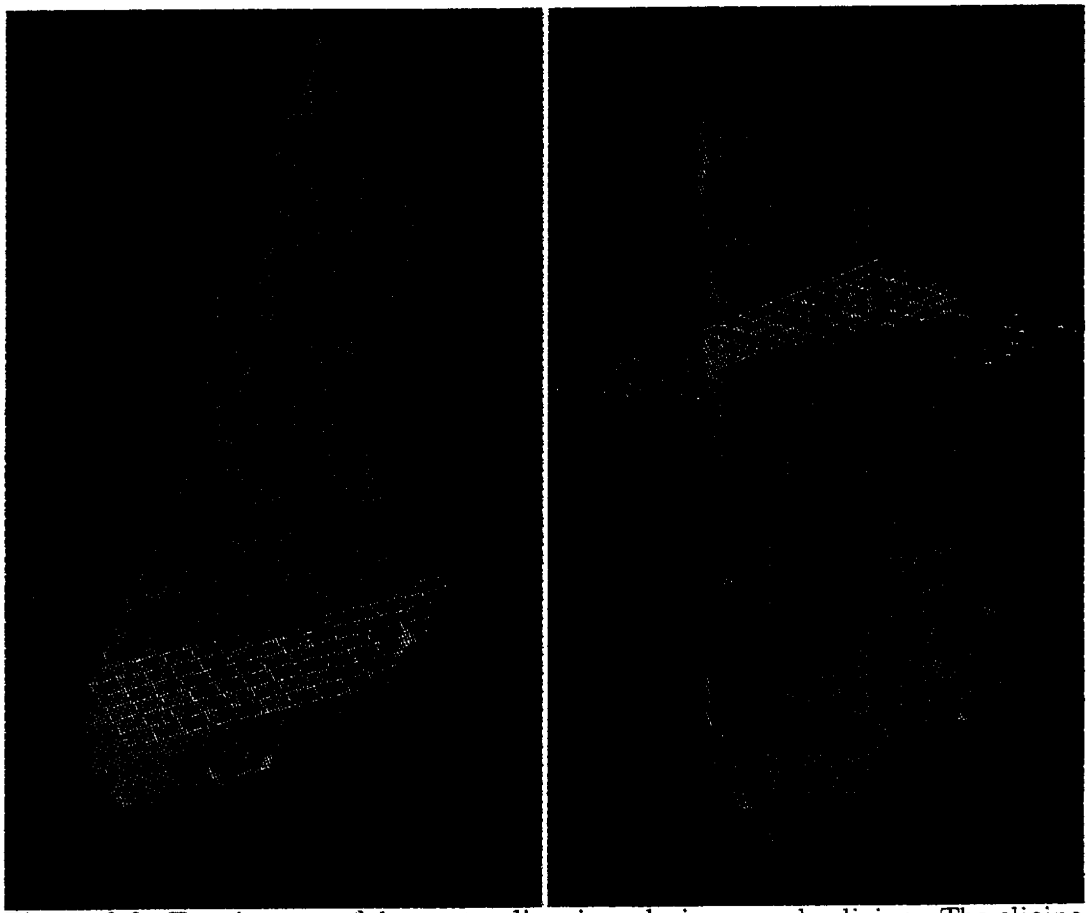

Figure 3.2. Two images of the pectoralis minor during muscle slicing. The slicing plane is shown as the blue grid. The green lines are defined slices.

The software calculated the points on the muscle's surface that intersect the plane. This was done by analyzing each of the triangles that define the muscle's surface. As long as all three of the triangle's vertices lay on one side of the plane, no action was taken. If one of the triangle's vertices lay on a different (odd) side of the plane than the other two, two points were recorded where the plane intersects the two lines of the triangle that emanated from the odd vertex. All of the original points were discarded, and the muscle was then defined by several slices of points.

The muscle's line of action was then determined in three-dimensions with these slices of points. The points were tessellated and centroids were found for the area of each of the resulting triangles. The locations of these centroids were then weighted by their corresponding triangle's area and averaged. This produced the centroid for each slice, and a point on the muscle's line of action. Figure 3.3 shows the results of muscle slicing with slices and the line of action. In this state, the data defining the muscles surfaces' were now completely prepared for dynamic modeling.

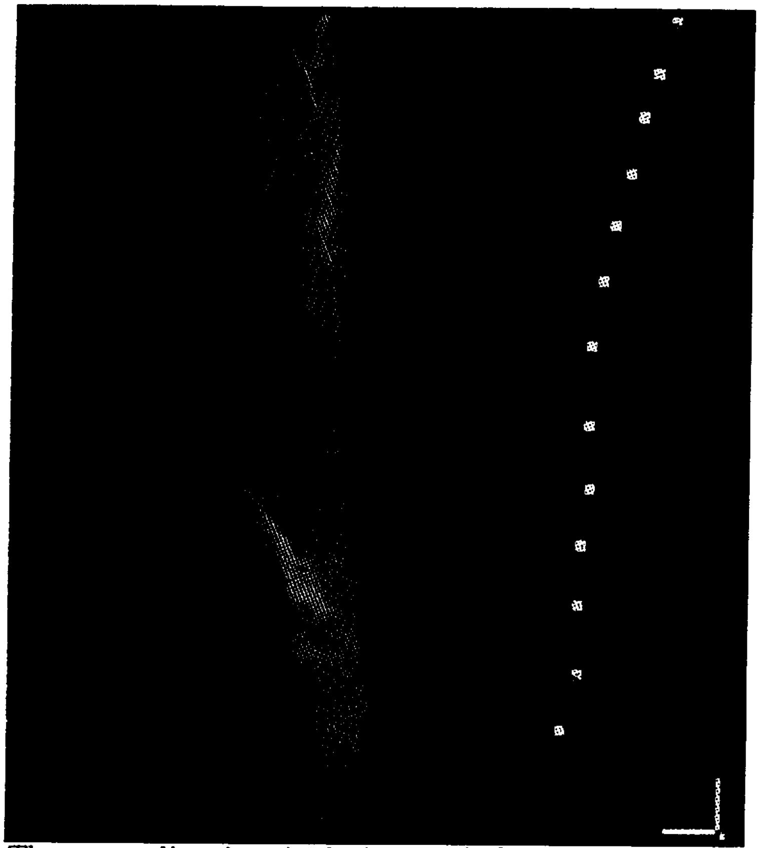

Figure 3.3. The pectoralis minor in decimated (left) and sliced (right) forms. The red line in the sliced muscle is the muscle's line of action.

The data from muscle slicing were arranged in text and stored in a separate file for each muscle. Each file was identified by the muscle's name followed by ".points" (e.g. pectoralis minor.points). The file holds data on each slice: tendon value, offset, transformation matrix, centroid, number of points, and point x, y, and z locations. The tendon level for each slice was listed first, but was added in manually after slicing.

Tendon values vary from zero to one as a measure of the fraction of the slice that is tendon. If no tendon figure was listed, the default value is zero. The offset value was computed by averaging the locations of all of the points in the slice. The centroid for each was computed by the method described above. Although not specifically studied, for all observed slices, the centroid lies within 0.0001% of the offset value. As an example of the file format, the teres minor's file, one of the smaller muscle files, is listed in Appendix C.

4. Muscle Modeling

4.1. Theoretical Muscle Modeling

Equations exist for modeling the geometry of skeletal muscle. However, these equations are limited to those that solve for a physiological property given a geometrical state. For example, Schneck and Hatze propose methods of relating Taylor series of muscle strain to the Calcium concentration in the sarcoplasm [40,41]. Solving for the strain from these equations is both mathematically difficult and not relevant for this application. The direct application of the proposed model had no physiological input to determine muscle length. As mentioned later, the model may one day be expanded to account for electromyographic activity.

4.2. Modeling Assumptions

Two major assumptions were made when modeling the muscle geometry. It was assumed that the muscles had constant volume and that the tendons would hold a constant geometry. Insertion and attachment patch areas were also assumed to be planar. This assumption is minor, in that it affects only one aspect of the modeling function, as described in Section 4.6, from which no other aspects are driven. This assumption was justified by visual examination of the sliced planes at the ends of the muscles from Chapter 3 compared to the actual measured origin and insertion sites of the muscles.

Skeletal muscles act as blood reservoirs during physical activity [12]. During rest, the skeletal muscles receive 15% to 20% of the total cardiac output. When muscles become active, their blood flow can increase as much as 20 times. The cause of the increase in blood flow is chemical. As a simplified example, during skeletal muscle activation, oxygen will be diverted to skeletal muscle from smooth muscle in the blood vessels. The lack of oxygen to the smooth muscle prevents it from contracting. Thus, the blood vessels dilate, and blood flow to the muscles is increased. This process alone causes the blood flow to skeletal muscles to increase by a factor of three [40]. However, the skeletal motions that the muscle model was applied have a duration of less than one second. The relative lag times for the chemical factors to increase blood flow to the skeletal muscles and the durations of skeletal motion are such that the increase in skeletal muscle volume was assumed to be negligible.

The second assumption was that tendons possess relatively constant geometry compared to muscle fibers. Although mechanical properties of most biological materials vary slightly from sample to sample, Young's Modulus (E) for collagen, the primary protein of tendons, is on the order of 10^9 dynes/cm^2; and Young's Modulus for myofilaments is on the order of 10^4 dynes/cm^2. It should be noted that muscle fibers are made up of a mix of myofilaments and collagen resulting in a strain modulus of approximately 10^7 dynes/cm^2 [40]. Even then, tendon's strain is about 1/100th the strain of the surrounding muscle. With this in mind, the tendon was assumed to have negligible strain relative to the surrounding muscle.

4.3. Wireframe Reconstruction

The first active step in the creation of the muscle geometry was to form a wireframe of the surface. This wireframe was needed to facilitate the change in the muscle geometry. Each muscle was reconstructed using a bicubic spline wireframe to define control points on a surface grid.

4.3.1. Cardinal Splines

Cardinal splines were selected to form the wireframe. Cardinal splines are bicubic, thus providing C2 surface continuity and inherently fixing the frame to the control points. The formula for a generic Cardinal spline is given below:

$$\vec{p}(t) = [\, t^3 \quad t^2 \quad t \quad 1 \,] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{p}_1 \\ \vec{p}_2 \\ \alpha(\vec{p}_2 - \vec{p}_0) \\ \alpha(\vec{p}_3 - \vec{p}_1) \end{bmatrix}$$

where the **p-vector** is the resulting interpolated point for the given parametric value, **t**. The p_x-vectors are the four surrounding points from which the p-vector is being interpolated. The **p_1-vector** and **p_2-vector** are the boundary points. (When t equals zero, the p-vector equals the p_1-vector. When t equals one, the p-vector equals the p_2-vector.) The **p_0-vector** is the point before the p_1-vector, and the **p_3-vector** is the point after the p_2-vector. The α value characterizes the proximity of the curvature of the spline to the control points. The α value was set at 0.5 to evenly distribute the curvature along each splined section [42].

An important property of Cardinal splines is that they pass through their defining control points. This was important for the model because the frame needed to fit directly to the control points measured from the Visible Human Project's muscles. This varies from the formulas of Bezier curves and general B-splines which do not necessarily fit through their control points. These splines can be forced to fit through their control points, but that property is not an identity of the spline itself [43].

Cardinal splines were also chosen because they are third-order. This characteristic provided an optimal cost-benefit ratio. Because they are third order, Cardinal splines provided C2 continuity; therefore the first and second derivatives (with respect to three-space) of all lines defining the wireframe were continuous. C2 continuity is usually used as an upper limit in industrial design and was thus deemed sufficient for this model [43]. Higher order splines also provided better than C2 continuity, but the number of computations needed increased exponentially and the resulting higher continuity was not needed.

4.3.2 Wireframe Organization

The wireframe of the model was set up as a square grid. Muscle defining data were read from previously made files. The latitudinal slices defined in these data sets were then reformed to standard resolutions. Additional latitudinal slices were then interpolated between the aforementioned slices to create the entire wireframe. This grid system simplified the dynamic movement of the muscle body. All control points were held in both Cartesian coordinates - to aid in the formation of splines - and radial coordinates - to aid bulging calculations.

The sliced muscle data described in Section 3.3 were organized as points that lay in latitudinal planes along the muscle. The number of points varied from plane to plane, and the number of planes varied from muscle to muscle. Each plane was also given a tendon value. If no tendon value was assigned to a given plane, it was assigned the default value of 100 percent muscle.

The data points for each latitudinal slice were recalculated. The number of new data points for each slice was set within the modeling function. During testing, and for all of the images seen here, the latitudinal resolution was set between fifteen and forty points. (This value should be increased when the modeling function is run on a workstation with a faster processor and more powerful graphics hardware. This value is stored in the "domuscles.h" header file as "LATITUDINAL RESOLUTION.") Cardinal splines were fit through the points measured by slicing the muscles in Section 3.3. The splines were wrapped around the beginning and end of the slice as well. New control points were then interpolated from the splines at equal distances along the perimeter of the slice. Each recalculated latitudinal slice then had the same number of points.

Control points were then calculated in the longitudinal direction. New points in the longitudinal direction were formed in a similar manner to those in the latitudinal direction, but no wrapping was performed at the ends as muscles do not close at their attachments site, but form to the bone. Similar to the number of new latitudinal control points, the number of new control points in the longitudinal direction for each strip along the side of the muscle can also be set within the modeling function. Values for longitudinal resolution during testing were set between fifteen and forty points. (This value should be increased when the modeling function is run on a workstation with a faster processor and more powerful graphics hardware. This value is stored in the "domuscles.h" header file as "LONGITUDINAL RESOLUTION.")

Each of these new points in the longitudinal direction was grouped with the other new points at its longitudinal length to form a new latitudinal slice (commonly referred to simply as "slices" later in this thesis). Tendon values were assigned to the newly formed latitudinal slices. Tendon values were interpolated linearly from the slice's distance between two previously defined planes and those planes' tendon values. The wireframe of the long head of the biceps brachii is shown in Figure 4.1 below.

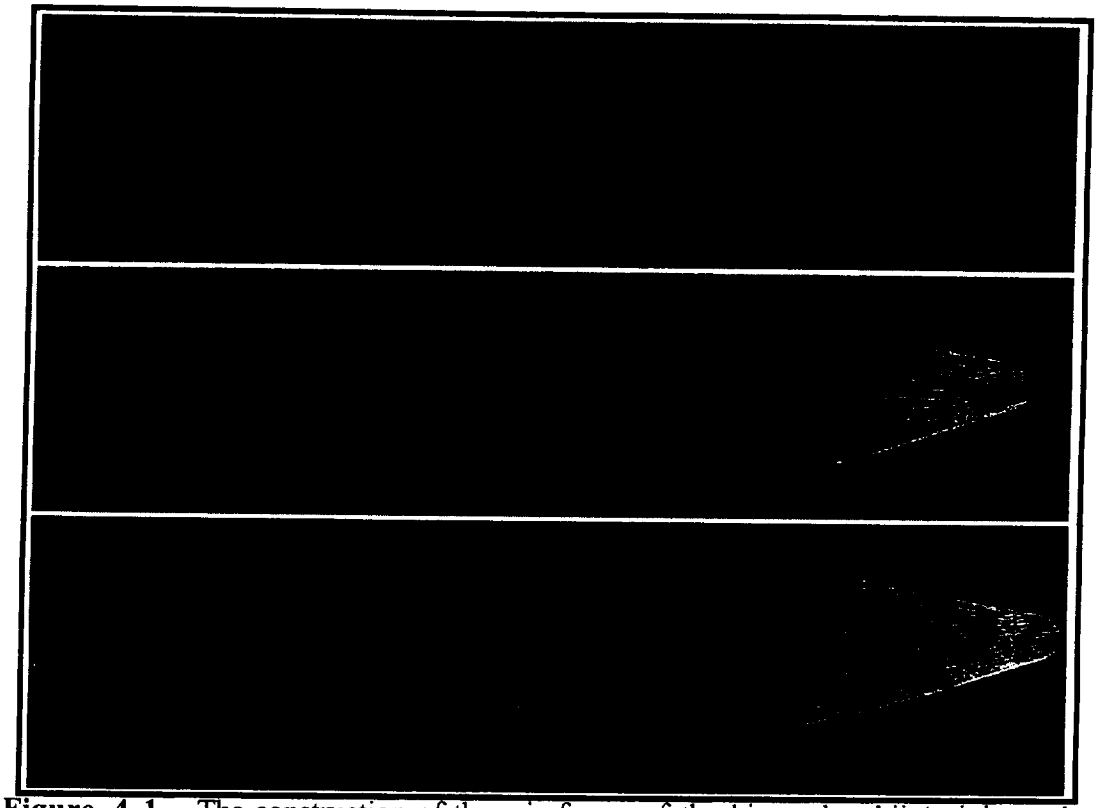

Figure 4.1. The construction of the wireframe of the biceps brachii (origin at the right) is shown above in three stages. The top image shows lines connecting the interpolated latitudinal points. The middle image shows the longitudinal interpolation. The bottom image is the completed wireframe.

4.4. Contraction

The contraction of muscles that must wrap around obstacles is discussed in Section 4.7. It is assumed that other muscles travel in straight lines. By "straight lines", it is meant that the centroid of each of the muscle slices lines up in a line.

The modeling function first accounted for muscle dynamics in the longitudinal direction: muscle contraction. Three passes were made over each muscle to set the longitudinal location of the slices that hold the control points. One pass is made to place the slices at each end that contain any amount of tendon, and one pass is made to longitudinally place slices that are considered all muscle.

The modeling function was fed the origin and insertion point of each muscle as input. The original origin and insertion points were recorded as the centroids of the end slices of the muscle during slicing. The original wireframe must now be sized to fit to the new muscle length.

In a simplified and idealized situation, the latitudinal slices - evenly spaced in the longitudinal direction due to the wireframe methodology - would all be spread evenly over the new length from the new origin point to the new insertion point. However, this model characterized tendon as a constant geometry tissue and allowed latitudinal slices to have varying amounts of tendon from zero to one hundred percent. Thus, the modeling function placed tendon slices in a partially parametric manner.

During the passes over each end of the muscle, the modeling function examined each slice as long as the percent of tendon was less than that of the previous slice and higher than a tolerance of 0.0001 percent. (This tolerance level was needed because the tendon value was held as a floating point variable, which rarely lands exactly on zero.) Each slice's length from its respective end, its tendon value, and the location of the previously placed slice was examined. The modeling function then computed the longitudinal length of the slice based on a weighted balance between parametric placement and placing the slice at its original length. The placement of each slice was based on the formula below:

$$L_{slice} = \text{tendon}\,(L_{oslice}) + (1 - \text{tendon})(L_{slice} - 1 + \left(\frac{L_{endslice} - L_{slice-1}}{\text{endslice} - \text{slice}}\right))$$

Where $\mathbf{L}_x$ is the longitudinal distance of slice x from the end of the muscle that the tendon in question originated, $\mathbf{L}_{0x}$ is the original longitudinal distance from the static measured muscle of slice x from the tendon's end of the muscle, **tendon** is the percent of the slice that is considered tendon, **slice** is number of the slice that is being placed as counted from its end of the muscle, and **endslice** is the number of the last slice that was placed at the other end of the muscle as counted from the end of the muscle that the tendon in question originated.

Once the slices that are associated with the tendons were placed, the modeling function fixes the longitudinal positions of the slices that are all muscle. The modeling function evenly distributes these muscle slices throughout the remaining length. The placement of these slices followed the function below:

$$L_{\text{slice}} = L_{\text{origin tendon}} + (L_{\text{insert tendon}} - L_{\text{origin tendon}})\left(\frac{\text{slice - origin tendon}}{\text{insert tendon - origin tendon}}\right)$$

Where L_x is the longitudinal distance of slice x from the origin point of the muscle, **slice** is the number of the slice that is being placed, **origin tendon** is the number of the last slice that was placed as a part of the origin end of tendon, and **insert tendon** is the number of the last slice that was placed as a part of the insertion end of the tendon.

4.5 Bulging

After the modeling function finalized the longitudinal locations of the control points on a slice-by-slice basis, it computed their latitudinal shift on point-by-point basis, thus accounting for muscle bulging. The modeling function based the bulge of each point on the constant volume assumption. As the slices spread apart or closed together as described in the previous section, the individual control points were then pulled in toward the muscle's line of action, or pushed away from the line of action, respectively, in order to hold the muscle at a constant volume.

Each slice was considered to contain the volume of its cross-section from half the longitudinal distance from the previous slice to half the longitudinal distance to the next slice. For end slices the volume was of the slice's cross-section from the location of the given end slice to half the longitudinal distance to the next slice inward on the muscle. The cross-sectional area of each slice was computed by tessellating the slice's control points. An example of the method of associating volume with respective slices is shown in Figure 4.2.

Each of the control points was defined in two-dimensions for each slice in both Cartesian coordinates and in polar coordinates with the line of action acting as the origin. Holding the control points in polar coordinates simplified bulging. Angles defining control points in the polar coordinate system were held constant throughout the muscle dynamics. However, the radii defining the control points were shortened or lengthened to adjust for a lengthening or contracting muscle, respectively.

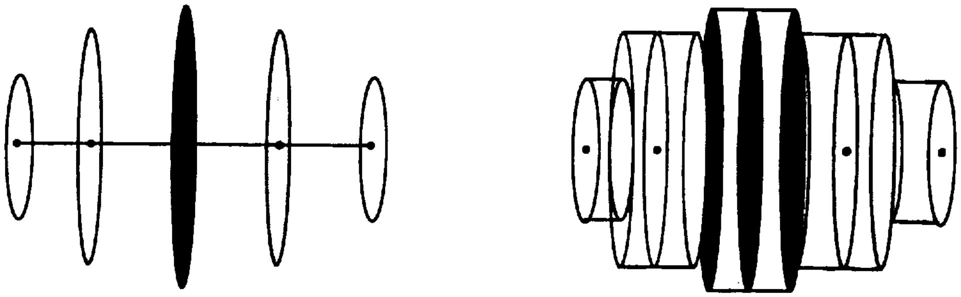

Figure 4.2. An example muscle on the left is shown by five circular slices and the muscle's line of action. The right image shows cylindrical volumes associated with each of the slices.

Tendons were being accounted for as constant geometry, but because the associated volume of a given slice was based on the location of its neighboring slices, increases or decreases in area for each slice were calculated dependent on other slices. This also perturbed the fixation of the geometries of slices composed entirely of tendon. If a longitudinal resolution of infinity is used, slices of pure tendon will still hold constant geometry. In realistic situations, however, a high enough longitudinal resolution should be used to reduce this affect. (Longitudinal resolutions of fifteen vertices reduced the problem tremendously, and resolutions of forty made tendon geometry changes disappear to the naked eye.)

The function used to control vertices' radii is shown below.

$$r = r_0\sqrt{\frac{depth_0}{depth}}$$

Where **r** is the radius of the point in question, $\mathbf{r_0}$ is the radius of the point in question before the muscle is perturbed from its original position, **depth** is the length from the slice holding the point in question half way to the slices on both sides of it, **depth$_0$** is measured in the same manner as depth, but from before the muscle was perturbed.

4.6 Plane Rotations

The last geometric adjustment performed by the modeling function was rotation of the individual planes holding the control points to account for rotations of the origin and insertion attachment patches. The ends of the muscle were assumed to be perfectly planar to simplify rotation of the intermediate planes. The difference in rotation between the origin and insertion sites were then distributed evenly over the intermediate slices, taking into account the rotation differences of the ends of the measured muscles from the Visible Human Project.

First, the Eulerian angles defining the origin and insertion end slices were computed for three-dimensional space. These α, β, and γ values, corresponding to the x, y, and z-axes, respectively, simplified the distribution of the rotations. These Eulerian angles were solved from the transformation matrices defining the slices [43]. If the rotations were held in matrix form, as they were throughout the modeling function, then differences in the matrices could not have been distributed over the slices defining the muscle. This is not possible because four-by-four transformation matrices contain more transformations than rotation alone, and distributing the difference between two of these matrices would have also transformed and scaled the intermediate muscle slices. However, the difference between the Eulerian angles was distributed evenly across the slices with no detrimental side effects.

After the differences in Eulerian angles between ends of the muscles were solved, the modeling function performed four quick steps to rotate the planes. First, it subtracted the original rotations of the origin and insertion planes, so they wouldn't be accounted for twice. Second, it linearly distributed the difference in each angle over the slices based on the given slice's longitudinal length from the origin point. Third, it converted the needed rotations from Eulerian back into matrix forms. Finally, the modeling function applied the four-by-four transformation matrices to the slices. A generic example of plane rotation is shown in Figure 4.3.

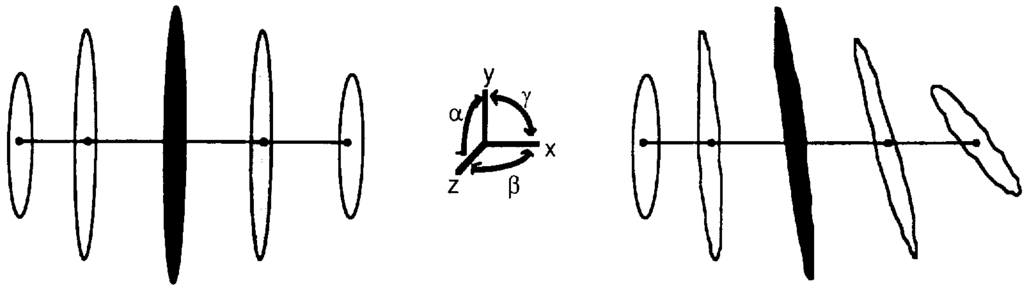

Figure 4.3. An example muscle on the left is shown by five circular slices and the muscle's line of action. The right image shows the slices evenly distributed in the γ-direction by the γ deflection of the rightmost slice.

4.7 Graphics Complement

With the muscle model near completion, finishing touches were applied with the benefit of the Silicon Graphics GL (Graphics Library). The GL is a library of graphics pipeline functions. The GL simplified rendering of the outer surface of the muscle. This included setting normal vectors, surfacing, and lighting. The final application program that will house the muscle modeling function has its own lighting definitions. Thus lighting was arbitrarily defined during testing stages only.

4.7.1 Normal Vectors

First, normal vectors were set for each vertex defining the muscle surface. Normal vectors defined surface orientations for the lighting. Normal vectors were set for each vertex in a three step process illustrated by Figure 4.4. First, the location was attained of the vertices in the corners of the local three-by-three grid surrounding the vertex in question. This method smoothes the reflected lighting from surface patch to surface patch compared to normal vectors solved from a local two-by-two grid without adding computational steps. This works due to diluting variances of the local grid vertices' locations. This principle is highlighted in Figure 4.4 by the large distances from vertices 0,-1 and 0,1 to vertex 0,0 and the short distances from vertex-1,0 and 1,0 to 0,0. If a two-by-two local grid had been used, the change in the magnitudes and possibly the directions of normal vectors from surface patch to surface patch would have varied more than using a three-by-three grid. Four-by-four or larger local grids were not used to determine normal vectors because at least two of the vertices in the local grid would no longer be adjacent to the vertex in question, and would therefore not have any relevance to its lighting.

After the locations of the corner vertices on the grid were found, the differences between the corners were solved. These distances are shown in Figure 4.4 as the $d_{x,-1}$-vector, $d_{x,1}$-vector, $d_{-1,y}$-vector, and $d_{1,y}$-vector. Then, the corresponding x-vectors were averaged and the corresponding y-vectors were averaged. This provided two vectors that characterized the local surface. The normal vector, the $n_{0.0}$-vector in Figure 4.4, was solved as the cross-product of the averaged y-vector with the averaged x-vector.

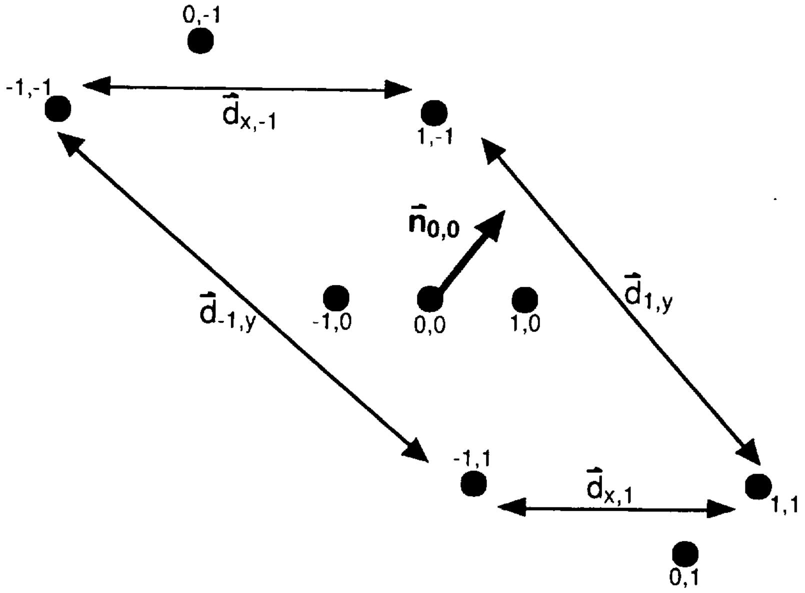

Figure 4.4. Solving for a normal vertex. The normal vector for vertex 0,0 is solved by taking an average of the distance between vertices -1,-1 and -1,1 and the distance between 1,-1 and 1,1 and taking the cross-product of that vector with an average of the distance between vertices -1,-1 and 1,-1 and the distance between -1,1 and 1,1.

4.7.2 Material Properties

Scientific replication of the color and shininess of the muscles and associated tendons was not within the scope of this research, but a subjective attempt was made using the prior models described in Section 1.2.2 as references. Colors needed to be defined for both a muscle material and a tendon material. After comparing color gradients from the Silicon Graphics GL with the prior models, all of which varied, a color scheme was set as shown in Table 4.1. In the Silicon Graphics GL, color settings vary from a minimum of 0 to a maximum of 1 with a shininess value from 0, most shiny, to 255, least shiny. The values in Table 4.1 are stored in the "domuscles.h" header file.

Table 4.1
Muscle and Tendon Color Settings

Characteristic	Muscle Material	Tendon Material
Ambient Red	0.80	0.75
Ambient Green	0.05	0.75
Ambient Blue	0.04	0.85
Diffuse Red	0.80	0.65
Diffuse Green	0.05	0.65
Diffuse Blue	0.04	0.75
Specular Red	0.85	0.65
Specular Green	0.15	0.65
Specular Blue	0.10	0.75
Shininess	50	10

After color and shininess properties were defined for muscle and tendon material, materials were then applied to each latitudinal slice individually. All points within slices composed entirely of muscle or entirely of tendon received the appropriate material. Vertices within slices that were a combination of both tendon and muscle received properties that were linearly interpolated between the two defined materials. The modeling function holds the color variables as real numbers, and thus has an infinite resolution of the color of the model between muscle and tendon.

An attempt at striating the muscle was also performed. Lines were drawn in the longitudinal direction along the wireframe, but - as can be seen somewhat in Figure 4.5 - the resulting striation was not realistic. Thus, the final version of the modeling function did not striate the muscles.

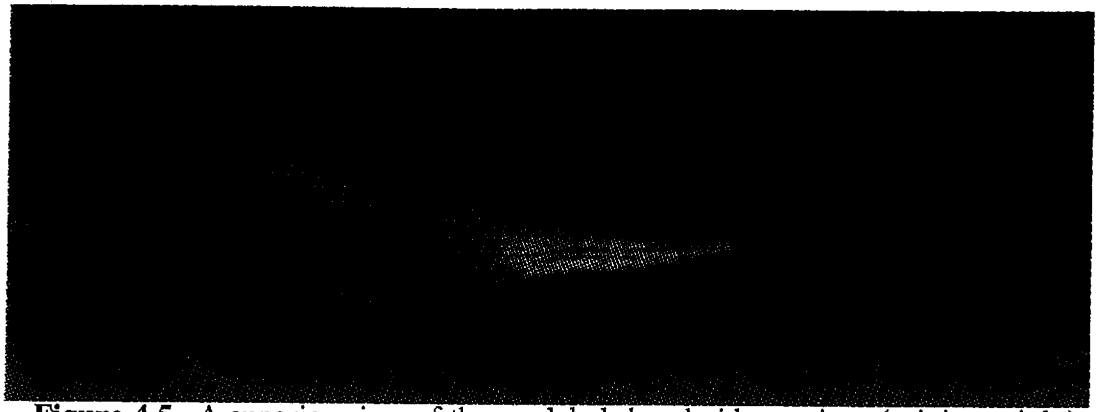

Figure 4.5. A superior view of the modeled rhomboideus minor (origin at right) with its wireframe. The slice at the insertion end of the muscle was denoted as tendon.

5. Results

The completed project including muscle measuring, data decimation, and dynamic muscle modeling provided a bevy of output data. The two major results from this project were the static models of the thirty-one skeletal muscles and the dynamic model of twenty-six of those thirty-one muscles. Several residual benefits were also produced from the two major results.

5.1. Static Models

The static muscle models described in Section 3 were the first of the major results garnered from this research. These models show the full geometry of the twenty-six muscles listed in Table 1.1 as well as the five muscles listed in Table 1.2. Figure 5.1 shows three views of the static muscle models on a relevant portion of the bone model created by Garner.

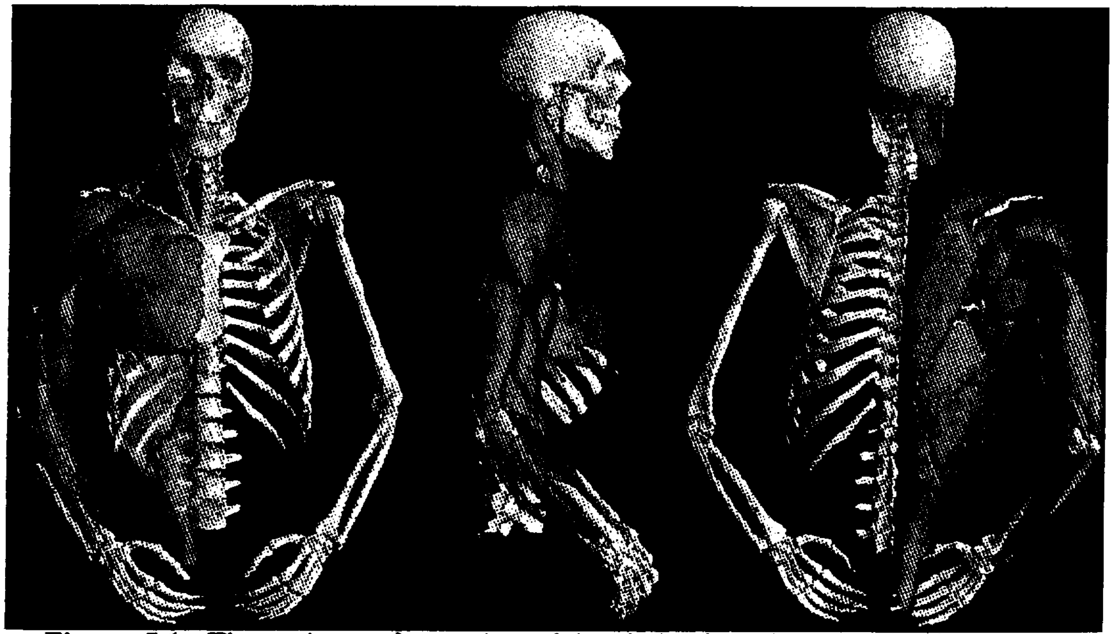

Figure 5.1. Three views of a portion of the skeletal model with all of the static modeled muscles applied.

An immense amount of geometric data was gathered from the static muscle models. Cross sectional outlines, areas, and centroids were determined for slices of all of the muscles. These data can also be determined in any cutting plane in any muscle with the help of the software used in Section 3.3 to slice the muscles [39]. The same applies to lines of action on a muscle by muscle basis. Volumetric data were also solved for muscles, but not needed. Volumes could also be solved as parts of muscles.

5.2. Dynamic Models

Of the thirty-five muscles listed in Tables 1.1 and 1.2, thirty were decimated and sliced. ".point" files described in Section 3.3 were created for all thirty of these muscles and stored within proximity of the muscle modeling function. The dynamics of the muscle models were tested using this library of files with dummy origin and insertion points that varied over time. These tests proved the muscle modeling function worthwhile and usable with the library of static muscle files. Figure 5.2 shows the rhomboideus minor during dynamics testing. As a side effect of the dynamic muscle model, all of the above geometrical data listed for the static muscles were now made available for the muscle in any position.

5.3. Video Presentation

A video presentation is in the process of being assembled at the time of this writing. The video will include a straight line muscle linked to the skeleton. The skeleton will undergo a dynamic movement to drive the muscle dynamics. The muscle should perform similar to the rhomboideus minor in Figure 5.2.

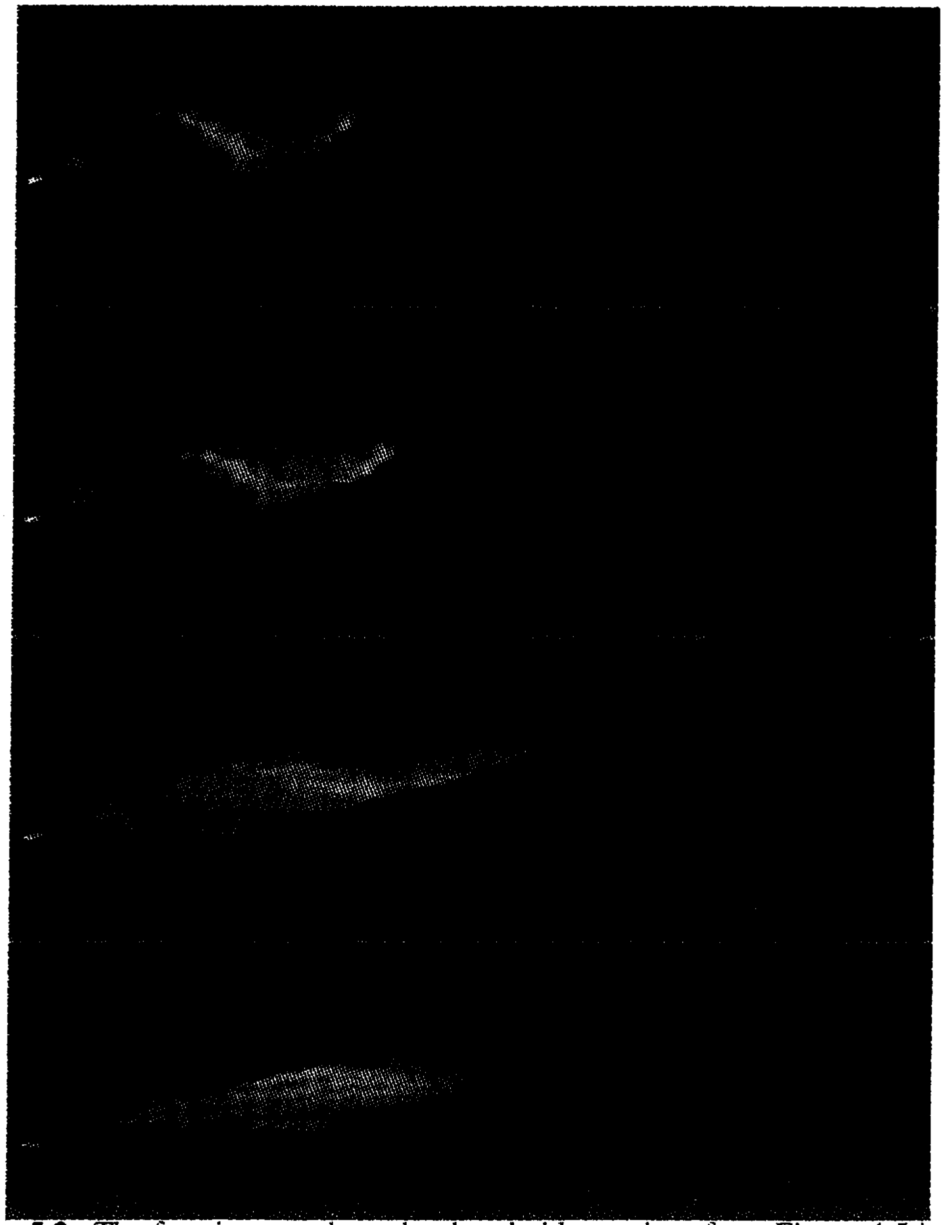

Figure 5.2. The four images show the rhomboideus minor from Figure 4.5 in four stages of flexion.

6. Conclusions

6.1. Original Contributions

This research accomplished several original contributions significant to the specialty of modeling muscle geometries. The most significant original contribution of this research was the creation of a dynamic muscle model based on actual anatomical measurements. As described earlier, several static muscle models have been created based on anatomical measurements, and some muscle models have been able to change their geometries dynamically, but the two characteristics have never before been linked.

As a residual effect of using anatomically based muscles, this research also began the use of attachment patches for dynamic muscle models. In all of the dynamic models before this, the muscles have all tapered at their ends with their attachment sites represented by points. Although the actual attachment site would better be represented by a three-dimensional curved patch, the improvement from point to two-dimensional area was clearly significant.

Another significant contribution of this research was the incorporation of tendon as a unique tissue in dynamic models. Because this was the first dynamic graphical model to account for tendon, all characteristics related to tendons in this model were original. This research began the use of dynamic models accounting for tendon as a constant geometry. This model also was the first dynamic model to assign separate properties to tendon and muscle tissue. The mixture of tendon and muscle within the same latitudinal slice was also an original contribution.

A minor contribution was the creation of the static muscle models shown in Figure 5.1. Although similar static surface muscle models were created in the past, and the method used in this research was similar to that of General Electric's CGSP, the muscles in this research were manually edited on a slice by slice basis before reconstruction to remove graphical anomalies. GE edited their muscle models after reconstruction. The results of this research appear to have a smoother surface. The downside was that the slice by slice editing method was extremely time intensive. Two graduate students and nine undergraduate assistants spent time manually editing muscle slices.

6.2. Future Work

6.2.1. Short Term Goals

The next step in this model is its integration with the skeletal model. As yet, all tests of the model have been performed without the skeletal model. The skeletal model will send the muscle model the translations and rotations of the origin bones and insertion bones for the muscles as well as obstacles in the muscles' paths. The muscle modeling function will then determine the new origin and insertion points as well as the appropriate rotations of the muscles' planes. As soon as the two models are integrated, the video presentation described above will also be made.

Another short term addition to on the model would enable it to wrap muscles around obstacles. Muscles in the skeletal model possess one of three geometric states: straight, wrapped around one obstacle, or wrapped around two obstacles. The modeling function can only display the straight muscles. Thus, creation of a wrapping function around obstacles is needed.

All obstacles are characterized by the skeletal model as cylinders or spheres. The skeletal model will feed the muscle model the points and rotations listed above as well as the points that the given muscle's line of action intersects an obstacle and departs from the obstacle as well as the obstacle's radius. The muscle model's job would be to compute the curve around the obstacle and place the latitudinal slices appropriately. The curve should be computed fitting a parabola to the line of action's intersection and departure points with the obstacle and the radius of the obstacle. A simplified example is shown in Figure 6.1.

Tendon values must still be defined in the files holding the static muscles' data, other than for testing purposes. No clear process was determined for objectively defining the percent of tendon of each latitudinal slice measured in Section 3.3. The default tendon value is zero percent tendon. It is imperative that tendon values are defined, subjectively at least, for the muscle slices if the modeling function's ability to account for tendon is to impart any real benefit.

During creation of the skeletal model, several muscles were divided into segments to more accurately represent their true kinematics. These were typically larger muscles with large attachment sites: the pectoralis major, and the latissimus dorsi for example. Since the skeletal model now views these muscles as their segments, and not the entire muscle, the static muscle files must also be divided accordingly. The muscle modeling function must be able to get data defining the geometry of just one segment, as opposed to the entire muscle. Unfortunately, the segmentation does not occur in the latitudinal direction, as the data are stored. The muscles are split along the longitudinal direction. Thus, splitting the data files will entail more than merely copying and pasting sections of data into different files.

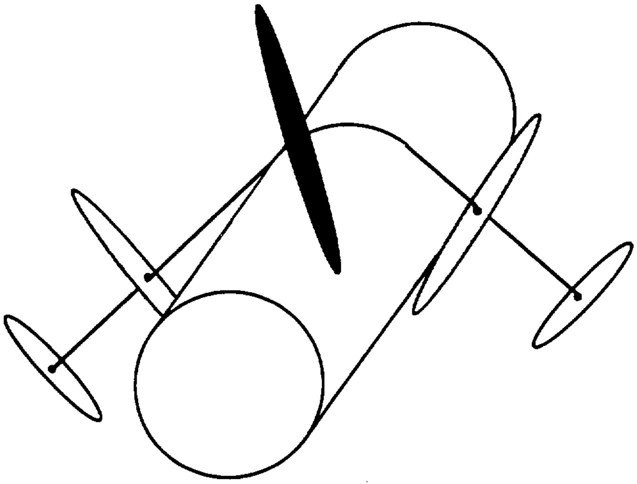

Figure 6.1. An example of parabolically fitting a muscle's line of action around a cylindrical obstacle.

6.2.2. Long Term Goals

This model has yet to be validated by experimental methods. Validation would add credibility to the model, but the resources needed to perform validation are presently not available. Suggested validation would include the digitization of points on the surface of the appropriate muscles on a cadaver in both relaxed and stimulated states. Due to the difficulties associated with cadaver studies, validation with cadaver studies is listed as long term future work.

Another long term project is the addition of fat and skin to the model. The HARD model had fat and skin applied to it [25,26]. This was likely done with a method of sweeping and revolving radii attachedto the surface verticesand determining their farthest distance from the surface. These radii's extremity distances were then likely connected to form the surface layer representing the outside surface of the fat. To increase, or decrease fat, the radii would be lengthened or shortened, respectively. Skin was then added by attaching image patches to the surface patches [25,26].

The muscle model may also be made to alter muscle geometries during static contractions. When electrical signals are fed to muscles, the muscle geometries change regardless of the three-space transformations of the insertion and origin sites. These changes are typically smaller than geometric changes when transformations of the attachment sites occur. Thus, the geometric changes for contracting muscles that are held still were neglected in this model.

A beneficial application of this research involves the facilitation of virtual surgery. To enable such an application, the model would need the addition of calculations to interpolate and create a volumetric model. The volume of each slice is currently calculated by the modeling function, but no interpolation is performed to "voxelate" the model.

6.3. Applications

This research is being directly applied as a graphical aid for a dynamic musculo-skeletal model being produced at the University of Texas at Austin. The muscle modeling function will be incorporated into this larger musculo-skeletal model to drive the graphics of the skeletal muscles. Other current and potential applications of this research vary widely by discipline.

The HARD model is already being used in the entertainment industry. Its use in major motion pictures and planned use in more validates the potential application of this research in the entertainment field [25,26]. This research surpasses the HARD model in all aspects of realism with the exception of the application of fat and skin. After such future work is accomplished to include fat and skin, this model could be used in the creation of industrial special effects.

The Voxel-Man muscle model and NCAR's voxel model are currently being applied to virtual surgery [13,16,17,18,19,20]. The barrier standing in the way of this model's use in virtual surgery is its surface definition. As described earlier, to apply this model to virtual surgery, each muscle would likely need to be filled with interpolated voxels. This voxel interpolation would allow the model to be cut away as seen on NCAR's model in Figure 1.4.

This model could also be used on biomechanical models with a minor amount of future work. The muscle model would benefit biomechanical model accuracy by accounting for forces and moments of inertia applied on the joints by the weight of the muscles. The modeling function calculates the volume of each slice, and a density would need to be applied to that volume, and the modeling function would need to be integrated with the biomechanical model.

A final suggested application of this model lies in the field of ergonomics. The model can determine the proximity of a subject's body. This ability can be enhanced with the addition of fat and skin as mentioned above. This could have a residual impact in clothing design.

APPENDIX A

Workstation Parameters

Internet Protocol Name: armus.me.utexas.edu
Location: Engineering Teaching Center II 3.110, The University of Texas at Austin
Type: Silicon Graphics Iris Indigo X24
Operating System: IRIX 4.0.5
Processor: 1 - 33 MHZ IP12
Hard Drive: 1 external - 1235 Mbytes
FPU: MIPS R2010A/R3010 VLSI Floating Point Chip Revision: 4.0
CPU: MIPS R2000A/R3000 Processor Chip Revision: 3.0
On-board serial ports: 2
Data cache size: 32 Kbytes
Instruction cache size: 32 Kbytes
Main memory size: 48 Mbytes
Integral Ethernet: ec0, version 0
CDROM: unit 4 on SCSI controller 0
Tape drive: unit 3 on SCSI controller 0: DAT
Disk drive: unit 1 on SCSI controller 0
Integral SCSI controller 0: Version WD33C93B, revision C
Iris Audio Processor: revision 10
Graphics board: GR2-XS24 with Z-buffer

APPENDIX B

Static Muscles

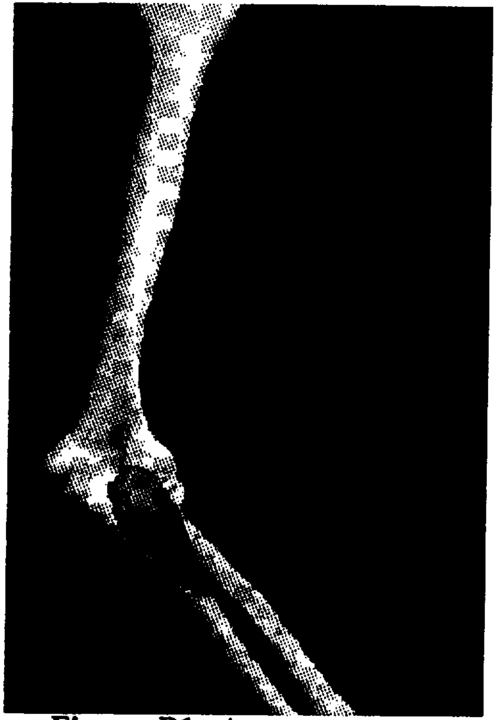

Figure B1. Anconeus.

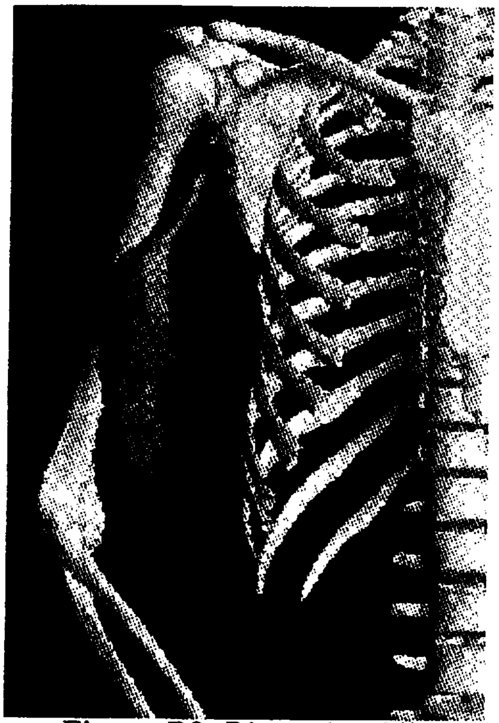

Figure B2. Biceps brachii.

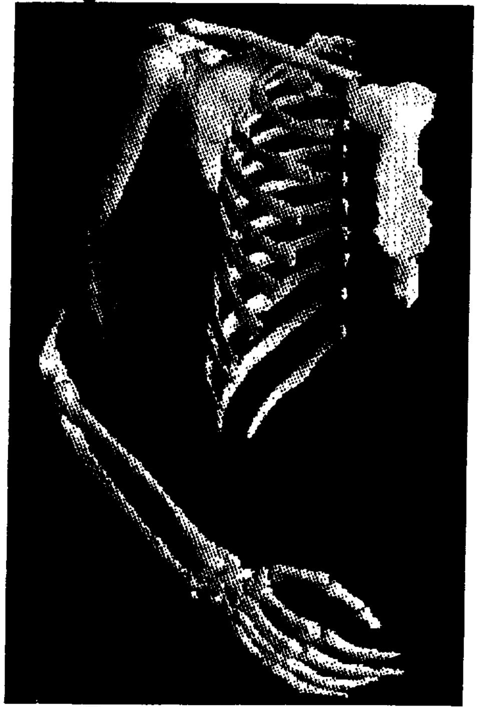

Figure B3. Brachialis

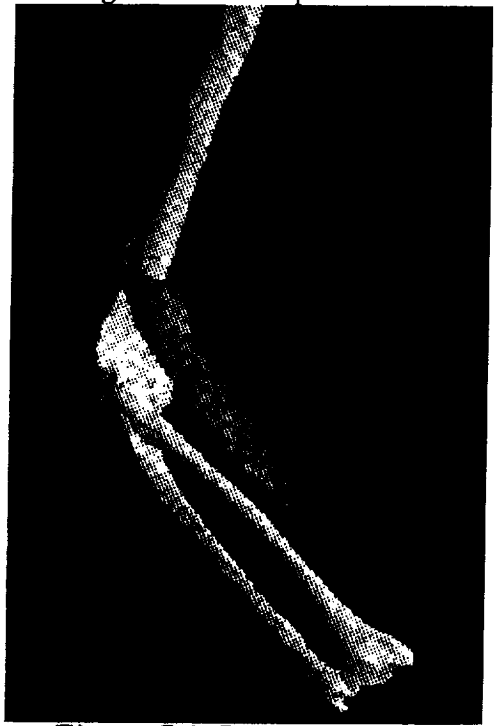

Figure B4. Brachioradialis.

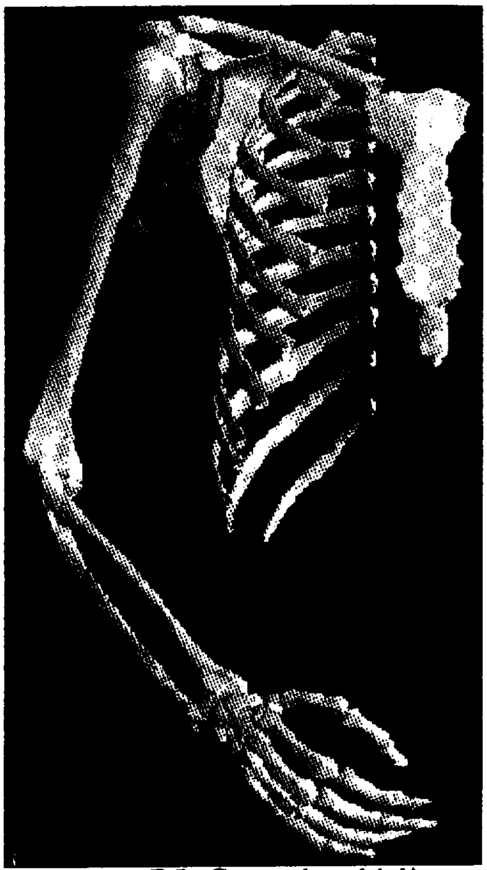

Figure B5. Coracobrachialis.

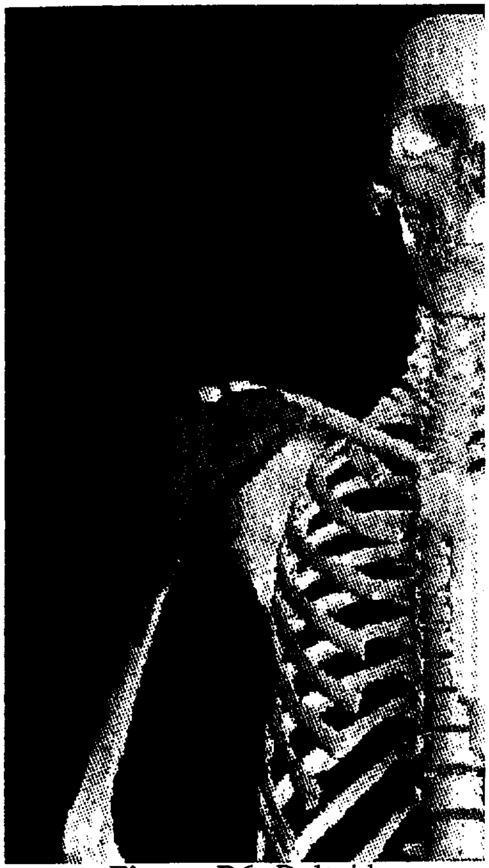

Figure B6. Deltoid.

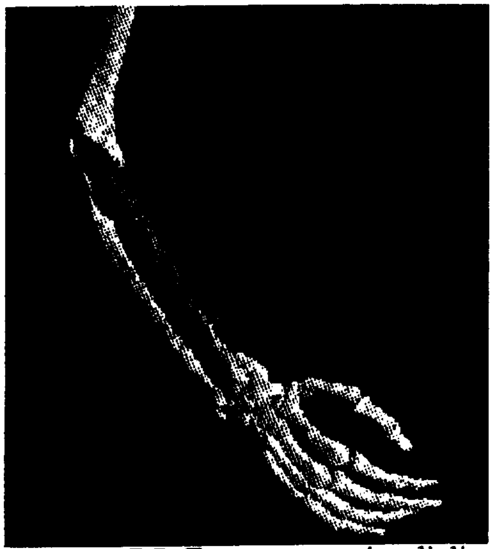

Figure B7. Extensor carpi radialis brevis.

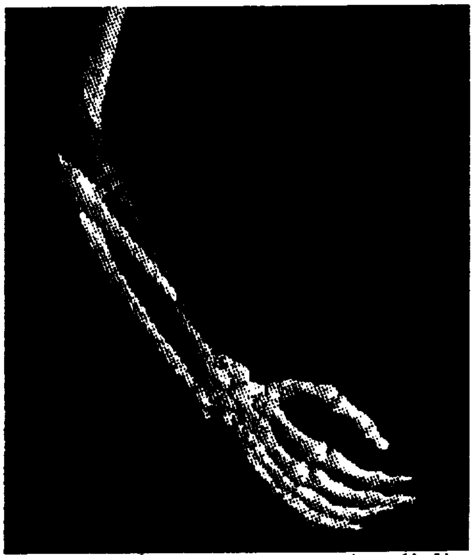

Figure B8. Extensor carpi radialis longus.

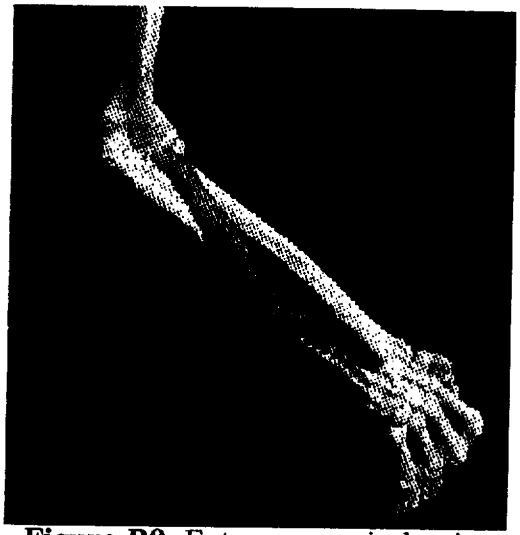

Figure B9. Extensor carpi ulnaris.

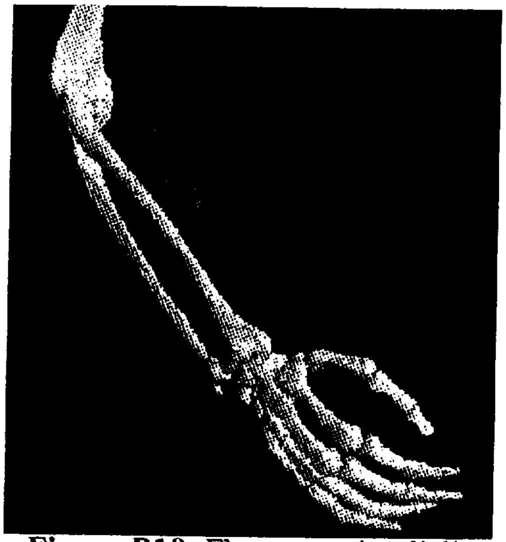

Figure B10. Flexor carpi radialis.

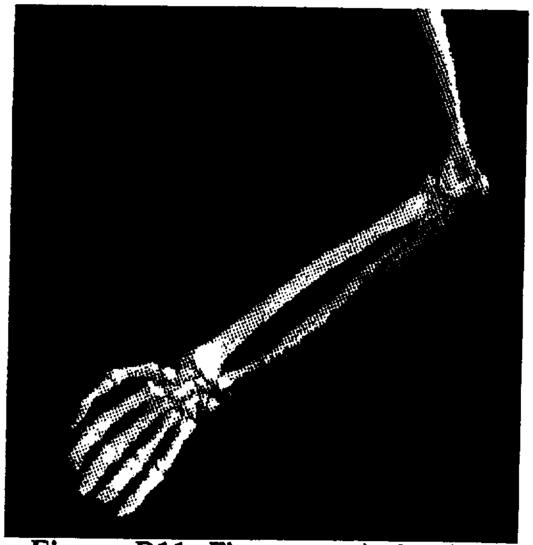

Figure B11. Flexor carpi ulnaris.

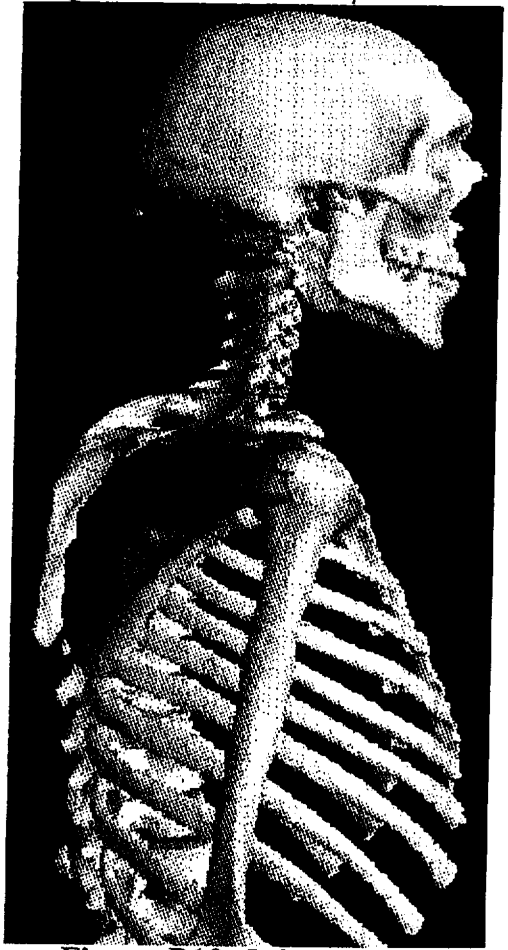

Figure B12. Infraspinatus.

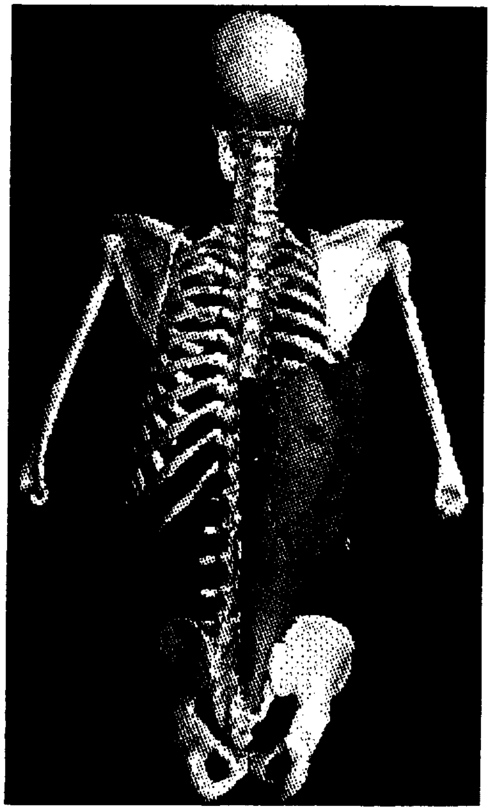

Figure B13. Latissimus dorsi.

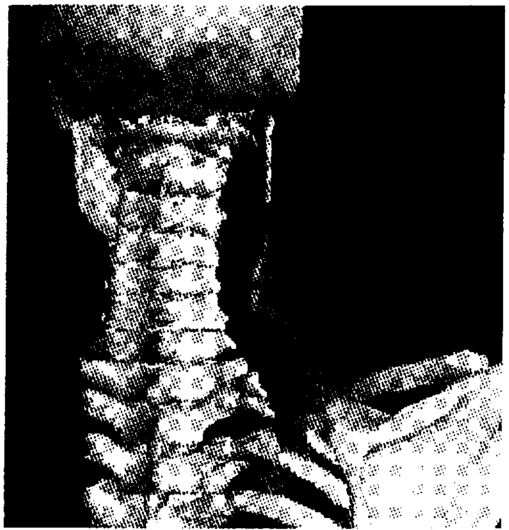

Figure B14. Levator scapulae.

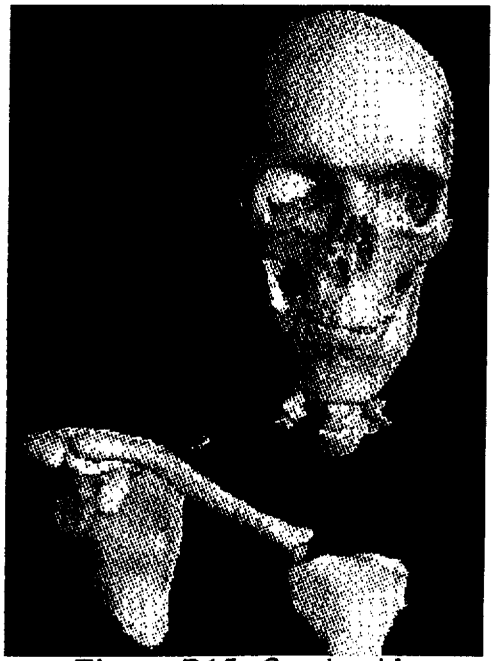

Figure B15. Omohyoid.

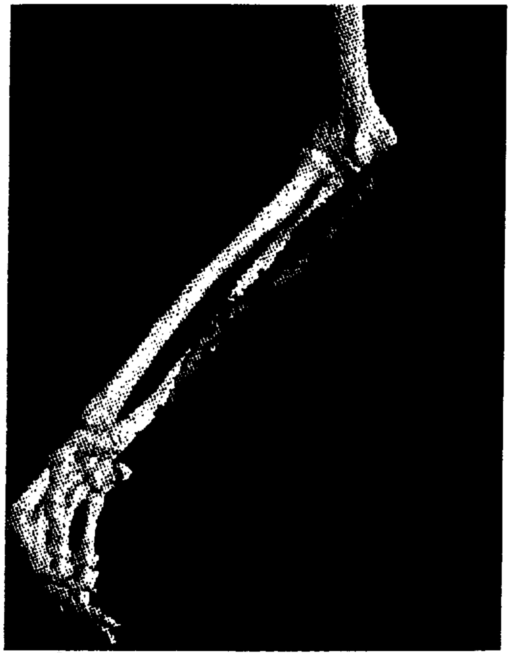

Figure B16. Palmaris longus.

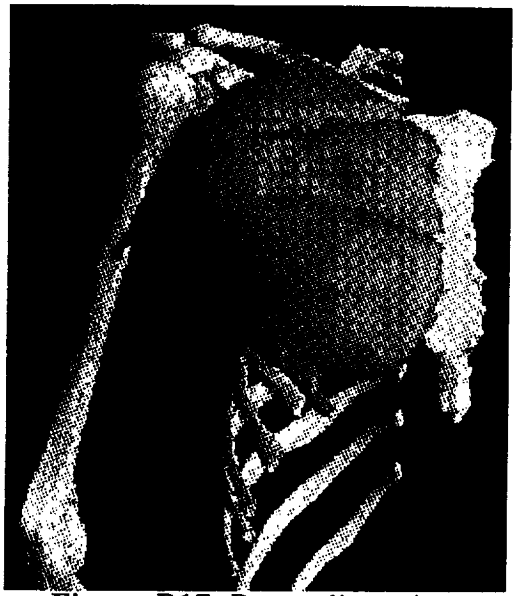

Figure B17. Pectoralis major.

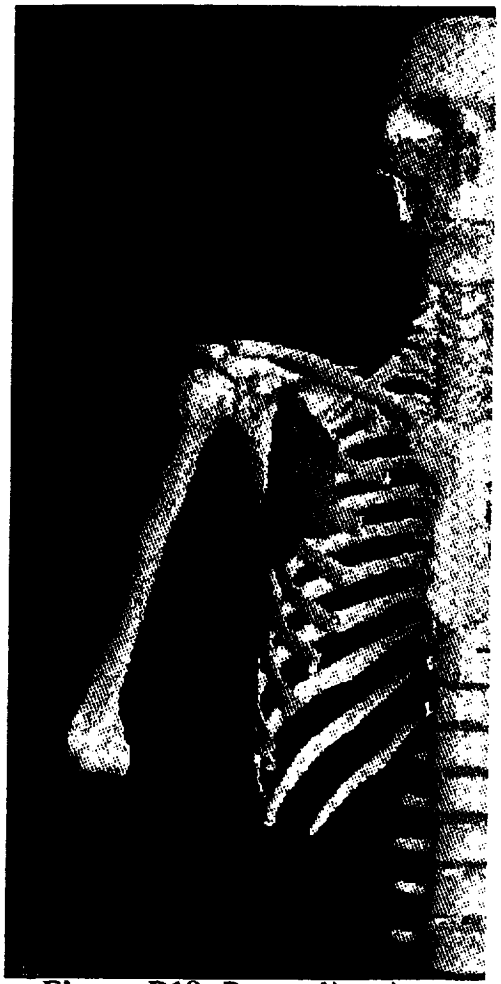

Figure B18. Pectoralis minor.

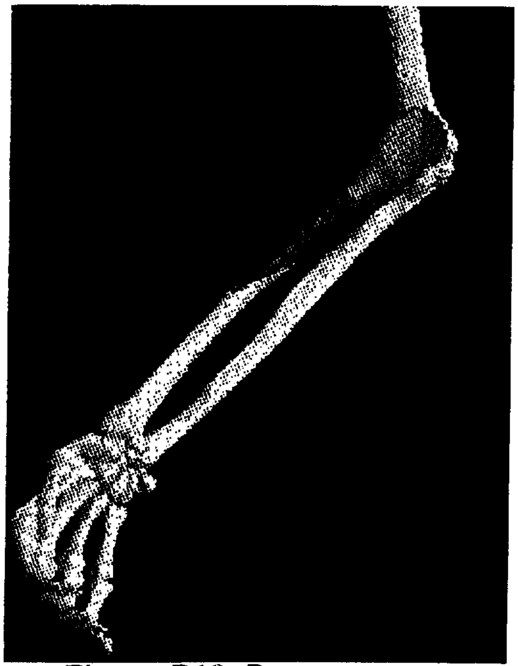

Figure B19. Pronator teres.

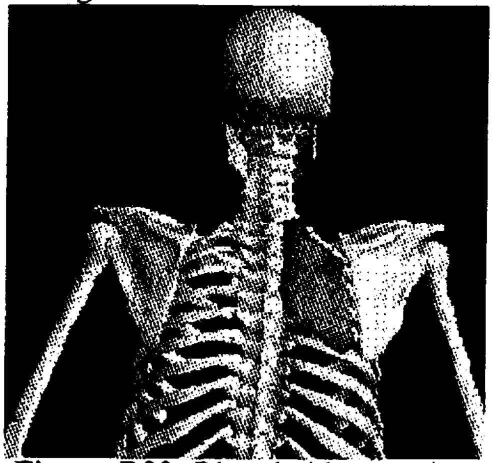

Figure B20. Rhomboideus major.

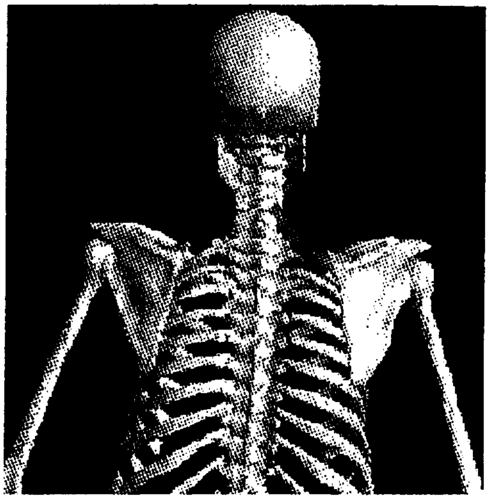

Figure B21. Rhomboideus minor.

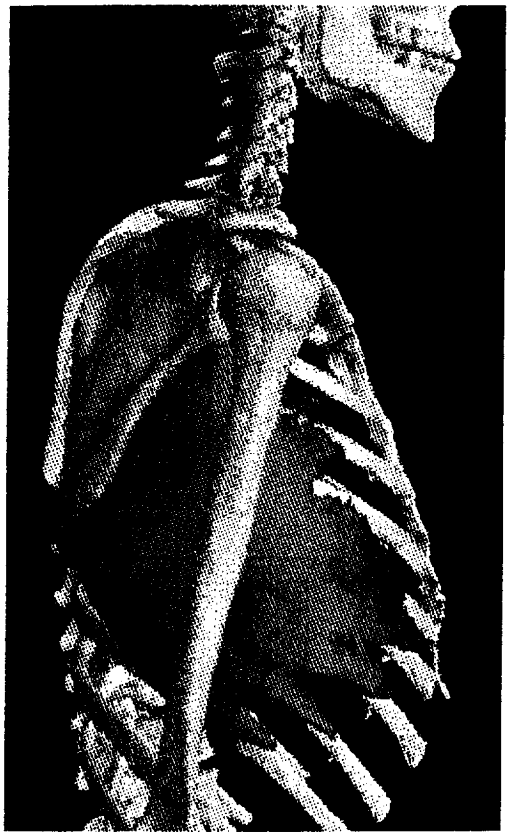

Figure B22. Serratus Anterior.

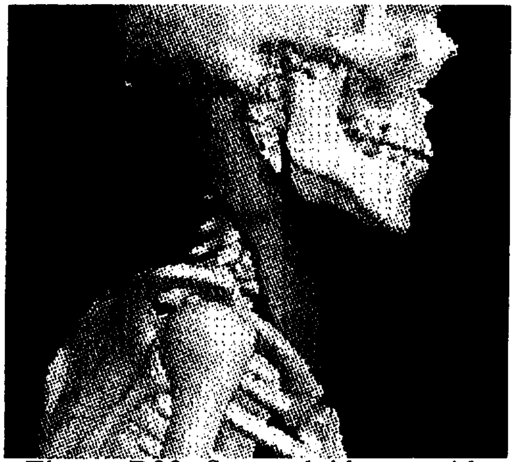

Figure B23. Sternocleidomastoid.

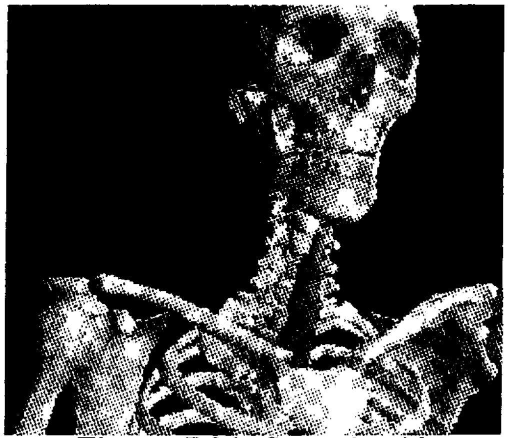

Figure B24. Sternohyoid.

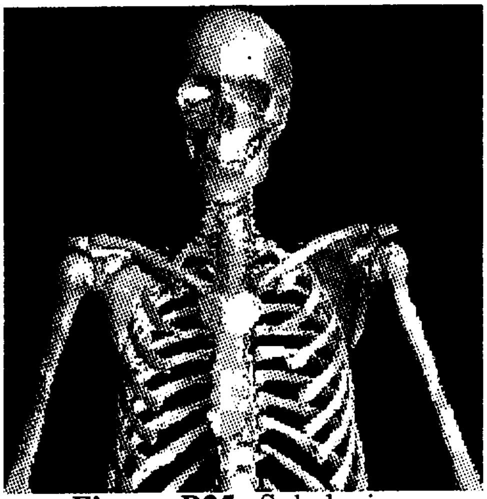

Figure B25. Subclavius.

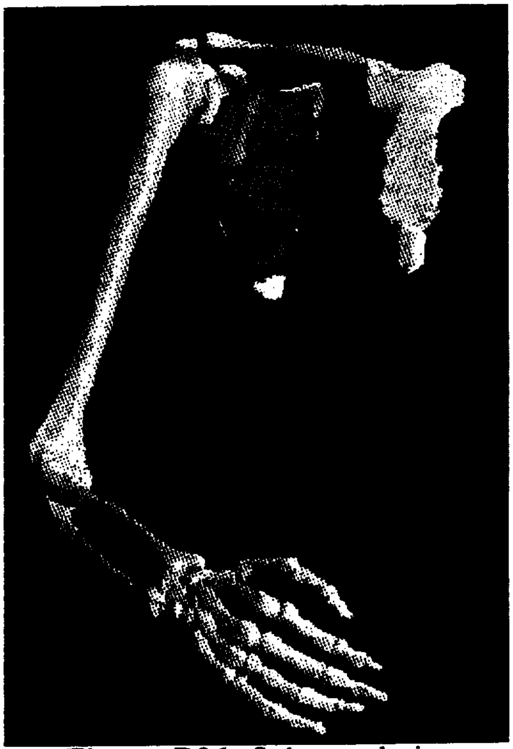

Figure B26. Subscapularis.

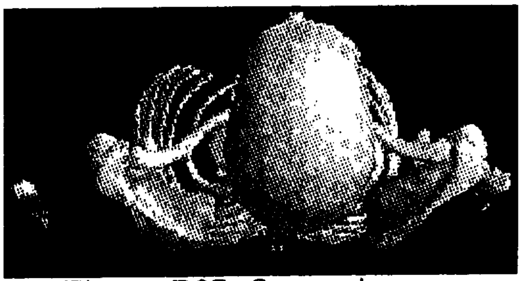

Figure B27. Supraspinatus.

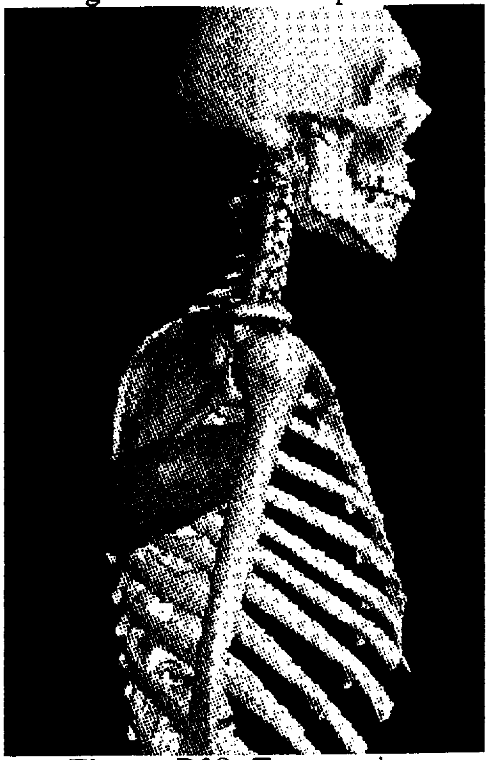

Figure B28. Teres major.

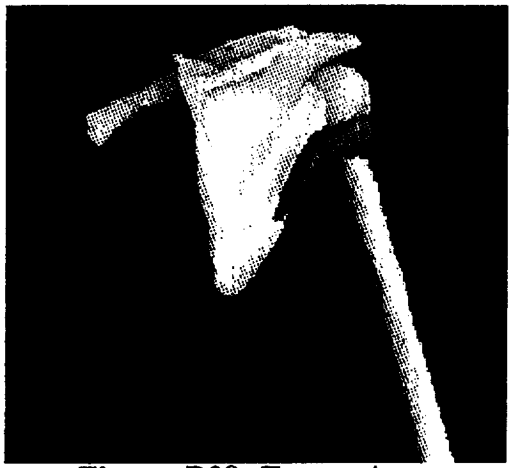

Figure B29. Teres minor.

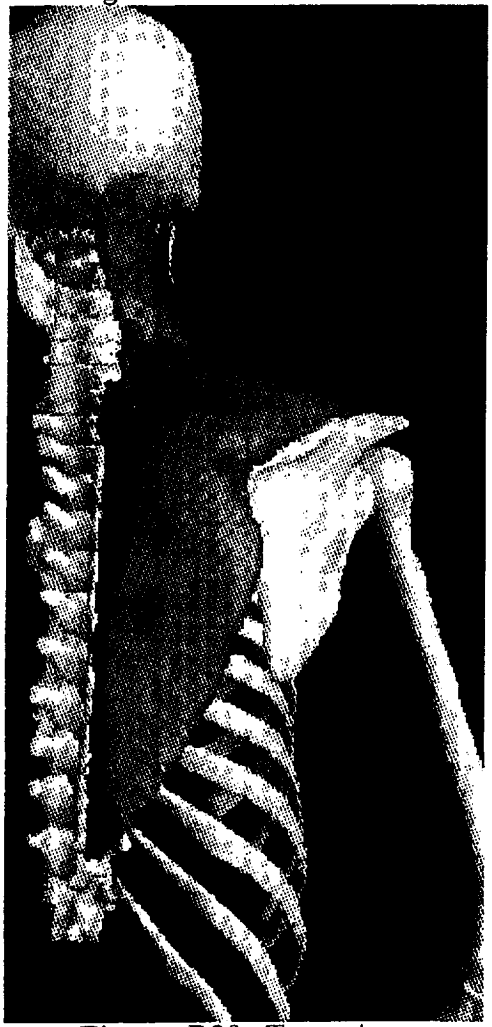

Figure B30. Trapezius.

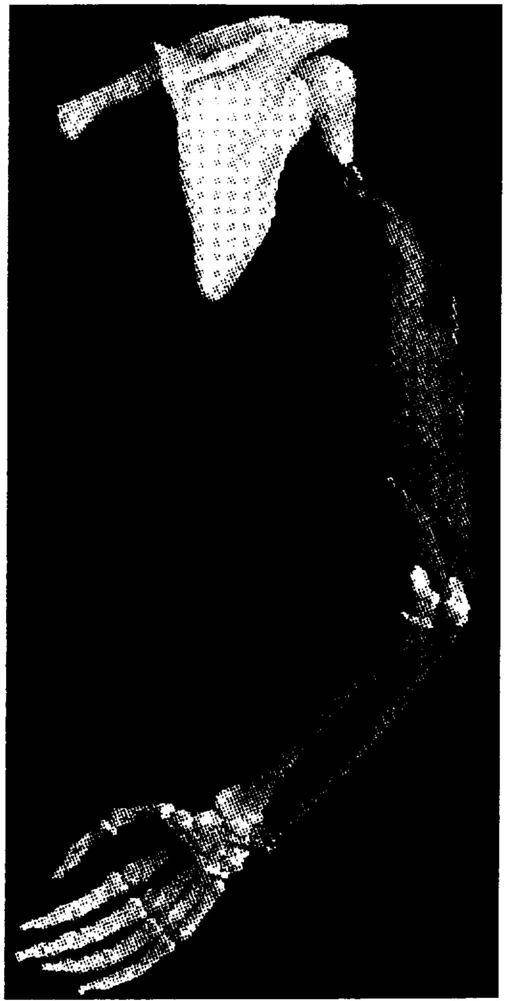

Figure B31. Triceps brachii.

APPENDIX C
Example Static Muscle File

The defining file for the teres minor.

```
numplanes 9

offset
        46.602020     250.142609    -1339.928101
matrix
        -1.646424     -1.389782     0.959142      0.000000
        0.987287      0.400216      2.137462      0.000000
        -1.460389     1.976335      0.298976      0.000000
        0.000000      0.000000      0.000000      1.000000
centroid
        46.602009     250.142624    -1339.928467
points 52
        33.853664     241.085602    -1342.329224
        32.950348     240.205109    -1340.921265
        32.413204     239.840286    -1341.133423
        32.744488     239.553558    -1337.619751
        32.788509     238.989243    -1333.674438
        32.834305     238.430359    -1329.756348
        33.717262     238.820084    -1328.019653
        35.008755     239.586105    -1326.774902
        35.183319     239.663300    -1326.432373
        35.442623     239.851959    -1326.412964
        36.098763     240.353409    -1326.522705
        37.322506     241.181992    -1326.022339
        41.747375     244.414413    -1325.775879
        46.136711     248.618561    -1332.126465
        48.132381     250.413162    -1334.241211
        50.647659     252.121246    -1333.246094
        52.491241     253.653488    -1334.369507
        54.392365     255.157242    -1335.023560
        55.336826     255.639465    -1333.597778
        58.793644     258.122864    -1333.128662
        59.072369     258.347992    -1333.255493
        63.610039     262.050415    -1335.564697
        63.726429     262.174164    -1335.814209
        63.963711     262.341370    -1335.760620
        64.071091     262.431763    -1335.833496
        65.044563     263.150024    -1335.826538
        63.976696     262.554749    -1337.107666
        63.999378     263.290741    -1341.862061
        62.986233     262.746429    -1343.212769
        62.426018     262.458862    -1344.048340
```

```
          58.051994     260.092346    -1349.770508
          57.774101     259.897766    -1349.841431
          57.232166     259.465546    -1349.631470
          56.365143     258.851929    -1349.810425
          52.340893     255.772705    -1349.112671
          52.263054     255.712112    -1349.092285
          51.786140     255.423096    -1349.511353
          50.208103     254.346252    -1350.101196
          49.573124     253.919159    -1350.379639
          48.020222     252.944443    -1351.521851
          46.612518     251.928970    -1351.685303
          46.109085     251.320908    -1350.124878
          43.275452     249.466248    -1351.706177
          42.700096     249.046097    -1351.739258
          40.373043     247.440170    -1352.490356
          39.715458     247.003204    -1352.813965
          38.969639     246.403900    -1352.495361
          37.491283     244.904144    -1349.802734
          35.498959     242.951889    -1346.629395
          35.066608     242.376465    -1344.937500
          34.343880     241.585632    -1343.240112
          33.853664     241.085602    -1342.329224


offset
          48.668663     258.739929    -1339.103394
matrix
          -2.072740     -0.840069     0.994908      0.000000
          0.998415      0.358338      2.241035      0.000000
          -0.928762     2.414078      0.035481      0.000000
          0.000000      0.000000      0.000000      1.000000
centroid
          48.668606     258.739929    -1339.103760
points 42
          49.691154     258.977692    -1328.514404
          47.098480     257.959412    -1327.099243
          46.078522     257.569977    -1327.301880
          43.628563     256.634613    -1327.791016
          38.572914     254.691391    -1327.916504
          35.066269     253.347626    -1328.279663
          34.557552     253.195648    -1331.255371
          34.504028     253.176971    -1331.385132
          34.536591     253.193405    -1331.651001
          31.750603     252.159836    -1334.255737
          34.739216     253.386475    -1339.483032
          36.110107     254.008102    -1345.892944
          36.598080     254.210846    -1346.914429
          37.844788     254.701996    -1347.697021
```

```
        42.723499     256.627808    -1351.020020
        43.165413     256.810822    -1351.904419
        45.018883     257.518555    -1351.540527
        46.519531     258.091278    -1351.226807
        49.015980     259.028503    -1349.645386
        49.908459     259.378113    -1350.071899
        50.477798     259.598938    -1350.191650
        56.423157     261.903687    -1351.377441
        57.267506     262.218536    -1350.696899
        58.308109     262.614441    -1350.394287
        61.414631     263.741302    -1345.746460
        62.626472     264.183563    -1344.117432
        62.757038     264.230865    -1343.918213
        64.022987     264.692322    -1342.176147
        64.022835     264.691345    -1342.114624
        67.941353     266.102570    -1335.559448
        64.743843     264.867737    -1335.242554
        63.947819     264.550629    -1334.503906
        61.788822     263.721405    -1334.599487
        61.291145     263.528259    -1334.484619
        59.841274     262.932892    -1331.929077
        58.862499     262.538666    -1330.726929
        56.616241     261.656219    -1329.484741
        55.923870     261.390594    -1329.536865
        52.526726     260.069733    -1328.590698
        51.844780     259.805389    -1328.456055
        51.388260     259.627075    -1328.274048
        49.691154     258.977692    -1328.514404


offset
        55.530693     268.780640    -1340.442871
matrix
        -2.479644     0.078542      0.942203      0.000000
        1.002617      0.297107      2.469714      0.000000
        -0.006775     2.838681      -0.314677     0.000000
        0.000000      0.000000      0.000000      1.000000
centroid
        55.530720     268.780609    -1340.442993
points 47
        72.881844     268.639954    -1342.085571
        73.036583     268.844055    -1340.247803
        71.872345     269.125580    -1337.682861
        70.920525     269.212372    -1336.879639
        69.897514     269.303406    -1336.036255
        68.026932     269.660339    -1332.776245
        67.921783     269.667755    -1332.707031
        67.865524     269.664337    -1332.736694
        61.999939     269.492004    -1334.164917
        61.687614     269.503632    -1334.053467
        58.105701     269.584015    -1333.251099
        57.766323     269.589844    -1333.191162
        57.615761     269.604614    -1333.054810
        57.347691     269.614655    -1332.958252
        53.730221     269.615326    -1332.874268
        52.831093     269.680328    -1332.268799
        49.845188     269.961243    -1329.670288
        49.215141     270.006714    -1329.246582
        48.944149     270.059021    -1328.768799
        48.432148     270.113861    -1328.263184
        46.662724     270.177704    -1327.649170
        44.815990     270.034271    -1328.903320
        44.748013     270.028229    -1328.956299
        44.731304     270.024780    -1328.987061
        44.717327     270.021362    -1329.017578
        43.155678     269.888458    -1330.182983
        41.982769     269.637756    -1332.419067
        42.728157     269.348969    -1335.040405
        42.651623     269.306366    -1335.422974
        42.577374     269.271210    -1335.738647
        42.669640     269.228607    -1336.125000
        42.847553     268.426697    -1343.362793
        44.176723     268.033661    -1346.936890
        44.825935     267.881897    -1348.319824
        45.513821     267.820251    -1348.890625
        46.932884     267.704254    -1349.967896
        50.549282     267.499390    -1351.893799
        51.094742     267.470703    -1352.164062
        51.263138     267.471527    -1352.160522
        55.586735     267.581360    -1351.262695
        56.428276     267.655060    -1350.615967
        60.652687     267.792114    -1349.470459
        61.934284     268.023224    -1347.413452
        65.052521     268.098969    -1346.797119
        67.053482     268.149445    -1346.384888
        69.318520     268.308502    -1344.998901
        72.881844     268.639954    -1342.085571


offset
        62.710762     276.845398    -1342.095947
matrix
        -2.459187     1.443759      0.600771      0.000000
        0.862245      0.233719      2.773098      0.000000
        1.411245      2.736840      -0.648455     0.000000
        0.000000      0.000000      0.000000      1.000000
```

```
centroid
        62.710758     276.845398    -1342.096558
points 37
        60.267490     280.025665    -1333.990723
        59.715954     280.411102    -1333.564331
        57.756958     281.720917    -1332.299683
        57.654766     281.780579    -1332.270142
        55.961964     282.614014    -1332.436646
        55.241722     282.635071    -1333.915405
        54.599461     282.841492    -1334.441895
        53.055717     283.190338    -1336.329224
        51.860821     283.021698    -1339.641479
        51.845425     282.910187    -1340.145630
        51.825336     282.669525    -1341.204956
        52.308739     281.840820    -1343.650635
        52.148075     281.729553    -1344.469849
        52.208439     281.588562    -1344.933594
        52.566605     281.276428    -1345.471436
        56.051262     278.554047    -1349.377686
        57.879612     276.978699    -1352.047485
        58.118080     276.809784    -1352.241333
        58.578636     276.583984    -1352.192017
        62.363018     274.506470    -1352.724365
        65.375732     273.550751    -1350.201416
        67.674637     272.384705    -1350.119629
        69.381958     271.737396    -1349.135986
        73.123611     270.720276    -1345.285767
        73.955666     270.457062    -1344.585693
        74.332008     270.387329    -1344.061035
        74.819969     270.993774    -1340.439575
        74.850677     271.033051    -1340.207031
        74.841034     271.047424    -1340.167236
        74.799606     271.077148    -1340.131958
        73.110657     272.814636    -1336.474487
        71.806969     273.710876    -1335.529053
        68.400352     275.503754    -1335.376099
        66.451408     276.747253    -1334.369385
        65.145714     277.599396    -1333.614502
        62.464863     279.128387    -1332.995605
        60.267490     280.025665    -1333.990723


offset
        64.997856     279.941315    -1344.276245
matrix
        -2.323323     1.758390      0.472969      0.000000
        0.779351      0.203038      2.833130      0.000000
        1.764060      2.571812      -0.649376     0.000000
```

```
          0.000000      0.000000      0.000000      1.000000
  centroid
          64.997833     279.941284    -1344.276489
  points 36
          60.387009     285.258514    -1335.743408
          59.489273     286.037567    -1335.096802
          58.663727     286.590607    -1335.149170
          58.132462     286.981628    -1335.043701
          57.324490     287.235779    -1336.232056
          56.347679     287.542389    -1337.671265
          56.245876     287.408295    -1338.478882
          55.552399     287.289673    -1340.832642
          54.864998     287.473724    -1341.970947
          54.808678     286.992096    -1344.031494
          55.714062     286.186646    -1344.761841
          56.179268     285.565613    -1345.957764
          56.113937     284.954285    -1348.556274
          59.262089     281.324371    -1354.380249
          59.907913     280.868866    -1354.429810
          61.064110     279.904480    -1355.108276
          61.864113     279.302612    -1355.318726
          63.525749     278.223114    -1355.080200
          64.872513     277.586304    -1353.943604
          68.029442     276.176697    -1350.950317
          70.297661     274.660858    -1350.791992
          71.421463     273.911865    -1350.705444
          74.218506     272.708435    -1347.873291
          75.830910     272.238708    -1345.353516
          76.136551     272.618622    -1343.018555
          75.842476     273.886688    -1338.795288
          75.846313     273.895111    -1338.751587
          75.754044     274.103149    -1338.178345
          75.672058     274.316010    -1337.557983
          74.848877     274.849365    -1337.681763
          73.855690     275.623199    -1337.315063
          71.858513     276.973022    -1337.394653
          71.398277     277.347443    -1337.161987
          71.116989     277.581696    -1336.998535
          60.791447     285.032593    -1335.539551
          60.387009     285.258514    -1335.743408


  offset
          69.649223     287.327850    -1346.266724
  matrix
          -1.437539     2.767686      0.186817      0.000000
          0.185707      -0.148480     3.076282      0.000000
          2.934977      1.604688      -0.085764     0.000000
```

```
        0.000000     0.000000     0.000000     1.000000
centroid
        69.649216    287.327881   -1346.266724
points 37
        64.588295    297.050598   -1337.541992
        65.576096    295.282593   -1336.818359
        65.810905    294.830078   -1337.249268
        66.237244    294.057404   -1337.116455
        67.603981    291.500549   -1338.184814
        69.831528    287.450256   -1337.737549
        70.382202    286.472198   -1337.192749
        73.718575    280.324158   -1338.049561
        74.950462    278.060364   -1338.249512
        76.136406    275.913910   -1337.826050
        76.470482    275.175781   -1340.203979
        76.835419    274.388885   -1342.438843
        76.840668    274.366089   -1342.685425
        77.084496    273.788239   -1345.153076
        77.062859    273.760345   -1346.415771
        76.349167    274.924835   -1349.050781
        76.317421    274.977600   -1349.150269
        75.137062    277.052429   -1350.723145
        74.843056    277.562958   -1351.231812
        74.838882    277.561920   -1351.393799
        74.573944    278.014954   -1351.984497
        72.192596    282.207275   -1355.037354
        71.721115    283.079956   -1354.843384
        71.413231    283.661316   -1354.502441
        70.310333    285.725677   -1353.620239
        68.745972    288.443665   -1356.299927
        67.404449    290.880737   -1356.610352
        66.680069    292.124542   -1358.127075
        66.433350    292.568085   -1358.271240
        65.720535    293.914520   -1357.472412
        64.832802    295.638611   -1355.593750
        63.332527    298.650604   -1350.579468
        63.178238    299.078430   -1347.854858
        63.123672    299.295166   -1345.666870
        63.287689    299.187347   -1342.071167
        64.403275    297.350555   -1338.261475
        64.588295    297.050598   -1337.541992


offset
        74.984940    294.395874   -1351.298584
matrix
        -1.470565    2.972327     0.343865     0.000000
        -0.860745    -0.812272    3.028404     0.000000
```

```
         2.979376      1.434713      1.234596      0.000000
         0.000000      0.000000      0.000000      1.000000
centroid
         74.984940     294.395874    -1351.298584
points 50
         68.025902     296.302063    -1336.719971
         68.167030     296.149597    -1336.883301
         68.302139     295.828461    -1336.836182
         70.537315     292.456238    -1338.311401
         72.979446     286.848511    -1337.688110
         73.062126     286.499664    -1337.482300
         74.333252     284.141968    -1337.809937
         76.620216     279.724945    -1338.195923
         76.935127     279.240448    -1338.392822
         77.348122     279.247742    -1339.398071
         79.550819     281.188629    -1346.969116
         80.523148     280.692932    -1348.739502
         81.008369     281.119873    -1350.406616
         81.588120     281.217194    -1351.918823
         81.530479     281.821930    -1352.482422
         81.464340     282.739807    -1353.389526
         81.751373     282.814636    -1354.169189
         81.644104     283.018433    -1354.147095
         82.680534     284.439301    -1358.299438
         82.812363     285.012482    -1359.283691
         83.545494     286.342316    -1362.598267
         83.559715     286.542450    -1362.865234
         82.320938     288.907776    -1362.624390
         82.197861     289.166351    -1362.627930
         81.806686     291.921844    -1364.885986
         79.772278     297.609497    -1366.586060
         78.395126     299.226654    -1365.141968
         73.844727     307.177429    -1363.400269
         72.464005     308.361725    -1361.444580
         71.637680     308.703979    -1359.848145
         70.217224     309.671967    -1357.545166
         70.235794     308.845978    -1356.630127
         69.538986     309.400116    -1355.592529
         69.301590     309.803070    -1355.487915
         69.216484     310.300781    -1355.860840
         69.004906     309.948730    -1354.941162
         68.998138     309.479828    -1354.379883
         69.067139     308.114227    -1352.959473
         68.255066     306.353882    -1348.954102
         68.173500     305.945526    -1348.282715
         68.648590     304.929382    -1348.248413
         68.679001     304.677032    -1348.028442
         68.122490     304.474121    -1346.449707
         67.835579     304.459381    -1345.740234
```

```
          67.412407     303.803467    -1343.956665
          66.943420     301.460785    -1340.102539
          67.493599     299.474274    -1339.121704
          67.248062     298.692078    -1337.620239
          67.594734     297.615540    -1337.205811
          68.025902     296.302063    -1336.719971



offset
          77.268608     297.786011    -1354.199219
matrix
          -0.570417     0.497311      0.868534      0.000000
          -0.080801     1.000167      -0.573907     0.000000
          -0.988286     -0.357563     -0.454564     0.000000
          0.000000      0.000000      0.000000      1.000000
centroid
          77.268593     297.785980    -1354.199219
points 64
          69.514267     296.921448    -1336.660156
          69.524223     296.909668    -1336.672485
          69.532196     296.883484    -1336.669189
          71.768570     292.857391    -1338.364502
          73.674561     286.715637    -1337.677246
          73.691986     286.506134    -1337.550293
          74.430580     284.746490    -1337.771973
          76.881485     278.619751    -1338.281250
          76.901550     278.546906    -1338.267578
          76.992615     278.658691    -1338.553467
          79.582542     281.625702    -1346.518188
          80.846283     281.013000    -1348.783813
          81.326073     281.444092    -1350.166016
          81.909576     281.579285    -1351.541016
          82.178909     282.872101    -1353.143555
          82.202599     282.995270    -1353.291870
          82.217209     282.988678    -1353.318481
          83.110550     283.922302    -1355.995117
          83.217346     284.001831    -1356.289917
          83.206551     284.037933    -1356.294800
          83.223618     284.056885    -1356.346802
          83.849472     285.942963    -1359.191162
          84.115753     286.368225    -1360.104614
          84.446449     288.295105    -1362.339233
          85.099586     290.569855    -1365.548584
          85.093582     291.420563    -1366.204712
          85.077515     291.531616    -1366.257080
          85.054947     291.652740    -1366.303345
          85.099144     291.585205    -1366.346313
          85.699432     293.025879    -1368.784668
```

```
         85.550072    293.373749    -1368.733521
         85.684303    294.075134    -1369.577148
         84.992271    296.941986    -1370.327637
         84.798889    298.719452    -1371.305420
         82.966187    305.201691    -1372.419800
         81.137100    309.481964    -1371.809937
         80.500679    310.456512    -1371.192871
         80.442795    310.574524    -1371.159912
         80.397705    310.567078    -1371.056030
         79.996841    311.340149    -1370.792603
         77.604012    313.330780    -1367.156006
         76.160690    314.694305    -1365.090576
         75.893753    314.786743    -1364.583008
         73.981018    314.789856    -1360.426880
         73.556396    314.307434    -1359.124146
         72.729324    314.410004    -1357.406738
         72.187256    314.385010    -1356.208496
         71.904449    313.526398    -1354.918213
         71.788612    312.744598    -1354.051514
         71.211349    312.575867    -1352.663696
         70.802460    312.150055    -1351.439697
         70.411034    311.624390    -1350.175293
         70.647758    311.178314    -1350.338989
         70.039772    307.382019    -1346.031006
         69.983765    307.150848    -1345.727417
         70.099861    306.480530    -1345.452515
         69.759659    305.793671    -1344.172607
         68.715866    302.445282    -1339.269409
         68.693306    302.203583    -1339.030151
         68.888695    300.994110    -1338.503662
         69.143524    299.506683    -1337.887695
         69.049980    299.305878    -1337.526367
         69.495232    297.015869    -1336.692993
         69.514267    296.921448    -1336.660156


offset
         83.907822    305.664246    -1360.062500
matrix
         -0.566979    0.712023      0.778038       0.000000
         0.166097     0.965770      -0.687002      0.000000
         -1.023733    -0.235020     -0.545445      0.000000
         0.000000     0.000000      0.000000       1.000000
centroid
         83.907661    305.664032    -1360.062378
points 79
         81.250740    285.710205    -1346.477783
         77.075821    288.654114    -1339.910400
```

75.767380	292.994659	-1339.324829
73.556427	299.627747	-1338.033203
73.566216	301.358093	-1338.797119
72.649704	302.900177	-1337.741455
72.646042	303.708649	-1338.082886
72.793015	304.306702	-1338.616455
73.053856	304.862335	-1339.345459
73.120049	304.959442	-1339.511475
73.111336	305.052917	-1339.535400
73.419121	305.834473	-1340.449829
74.821861	310.761505	-1345.205566
74.822800	310.770844	-1345.211304
74.831383	310.772980	-1345.228394
74.846359	310.816071	-1345.275024
76.100784	312.454956	-1348.335571
76.189247	312.503906	-1348.522705
76.318535	313.126617	-1349.033691
77.020294	312.952240	-1350.275635
76.666084	313.738617	-1349.949707
79.014389	315.368164	-1355.059326
79.110268	316.223053	-1355.607666
79.565376	316.481049	-1356.572998
80.028023	319.419373	-1358.707397
80.552124	319.655609	-1359.792847
81.357689	320.565186	-1361.696655
82.347916	321.631531	-1364.014648
84.077515	321.408356	-1367.164795
84.157249	321.377258	-1367.301025
86.045113	322.284729	-1371.235352
86.225174	322.276367	-1371.569702
86.449249	321.942932	-1371.846558
87.137482	321.879272	-1373.110840
88.159126	322.352722	-1375.232422
89.442390	322.537689	-1377.720581
89.683678	322.273743	-1378.059692
89.943596	321.881744	-1378.378662
89.617004	321.109039	-1377.432739
91.509483	321.281219	-1381.058838
91.810425	321.436737	-1381.690674
91.992691	321.532928	-1382.074219
92.349068	321.224640	-1382.610352
92.527992	321.265015	-1382.963501
93.547104	320.168884	-1384.403931
94.583115	319.839050	-1386.206299
96.021332	317.817932	-1388.034790
97.251884	317.587311	-1390.244995
97.348007	317.549286	-1390.409058
97.411743	317.365509	-1390.449463
97.193947	314.987976	-1389.016357

97.126740	314.970032	-1388.882446
97.088890	314.956970	-1388.805786
97.054520	314.826660	-1388.685059
96.209129	312.253662	-1385.989746
95.834198	311.220428	-1384.840820
93.901215	308.660553	-1380.109863
93.555923	308.299286	-1379.306152
93.334877	307.910339	-1378.723755
93.186119	306.283905	-1377.743652
92.155052	306.058502	-1375.711426
91.108063	303.363892	-1372.585327
87.797394	297.691040	-1363.927246
87.747269	297.695374	-1363.835083
87.707741	297.481964	-1363.668945
87.477348	295.466431	-1362.368042
87.109047	295.292267	-1361.601807
87.156174	294.831970	-1361.491943
87.141052	294.573364	-1361.352051
86.912148	293.860870	-1360.615479
86.676094	292.822845	-1359.725098
84.866951	287.761749	-1354.148926
84.691063	287.379913	-1353.654175
84.478821	287.125458	-1353.146240
84.318954	287.180298	-1352.869873
83.586296	287.678131	-1351.709229
83.041298	288.315948	-1350.961182
80.555595	289.187225	-1346.671265
81.250740	285.710205	-1346.477783

APPENDIX D

Modeling Function Location and Construction

The muscle modeling function and its associated header file are located on the Silicon Graphics Indigo named Armus in the Engineering Teaching Center room 3.110 on the University of Texas at Austin campus. This workstation is listed on the internet as armus.me.utexas.edu. The paths to the modeling function and its associated header file are /usr/people/garner/DAVE/VIA/TEST/domuscles.c, and /usr/people/garner/DAVE/VIA/TEST/domuscles.h, respectively.

The modeling function is made up of several subfunctions. These subfunctions' names and descriptions are listed below in alphabetical order.

angleorder - Sorts the vertices for one latitudinal plane in order from lowest angle on that plane to highest.

bulgeslice - Applies a bulge factor to a given slice.

cartesianize - Calculates three-dimensional cartesian values for vertices based on their slice number and angle and radius within the plane.

computeareas - Computes the area for all of the planes of each muscle.

computerelaxedlength - Solves for the length between each vertex and the center of its latitudinal plane for the static muscle data. This length is commonly referred to as the radius.

computeslicedepths - Solves for the depth of each plane in each muscle.

computevolumes - Solves for the total volume of muscle, opposed to tendon, in each muscle.

copyoriginaldata - Copies originally read static muscle data into backup variables to free the original variables for dynamic data.

dostraightmuscles - Applies dynamics to muscles by computing the needed bulging and rotations for latitudinal planes.

drawmuscles - Draws muscles that are composed of an equal number of vertices on each latitudinal plane with all vertices sorted in order of increasing angle on their plane.

initialparametrization - Computes the parametric length value (zero at the origin, one at the insertion) for each slice in each muscle.

latspline - Fits cardinal spline functions to the vertices defining latitudinal planes and interpolates new vertices to define the planes. *latspline* is a prerequisite for *longspline.*

longspline - Fits cardinal spline functions to corresponding vertices in the latitudinal planes interpolates new vertices to define new latitudinal planes. *latspline* is a prerequisite for *longspline*.

radialize - Calculates an eulerian angle for each verticex in each muscle.

readmuscles - Reads in all of the data for all of the muscles defined in

return_bone_number - Returns a predefined number representing each bone in the model: one for clavicle, two for scapula, etc.

rotateslice - Rotates all of the vertices in a given slice for given α, β, and γ-values.

solveoffsets - Solves for the centers of slices by averaging the locations of the vertices on that plane. (NOTE: This method has less computations than solving by tesselation and held less than 0.1 percent error compared to solving by tesselation.)

translateslice - Translates all of the vertices in a given slice for given x, y, and z-values.

BIBLIOGRAPHY

[1] Pandy, M. G. and Zajac, F. E. Optimal Muscular Coordination Strategies for Jumping. *Journal of Biomechanics*. 24:1-10. 1991.

[2] Pandy, M.G., et al. An Optimal Control Model for Maximum-Height Human Jumping. *Journal of Biomechanics*. 23:1185-1198. 1990.

[3] http://www.nlm.nih.gov/publications/factsheets/visible_human.html. The Visible Human Project Internet Site. January, 1997.

[4] http://www.crd.ge.com/esl/cgsp/projects/vm. Marching Through the Visible Man. January, 1997.

[5] Banvard, Richard A. Office of High Performance Computing and Communications, Lister Hill National Center for Biomedical Communications. Electronic mail correspondence with the author. May 23, 1997.

[6] Tiede, Ulf, et al. Visualizing the Visible Human.. *IEEE Computer Graphics & Applications*: v.16 n.1. 1996.

[7] http://www.nlm.nih.gov/research/visible/image/thorax_fresh.jpg. JPEG image 512x315 pixels. February, 1997.

[8] http:// www.nlm.nih.gov/research/visible/image/thorax.jpg. JPEG image 786x459 pixels. February, 1997.

[9] Netter, Frank H. Atlas of Human Anatomy. Summit, NJ: Ciba-Geigy Corporation, 1989.

[10]Barret, C.P., et al. Primer of Sectional Anatomy with MRI and CT Correlation. 2nd ed. Baltimore: Williams & Wilkins, 1994.

[11]Farrar, Roger P. *Applied Human Anatomy* course supplement, The University of Texas at Austin, spring 1997.

[12]Ganong, Willian F. Review of Medical Physiology. 17th ed. Norwalk, CT: Appleton & Lange, 1995.

[13]http://www.ncar.ucar.edu/METASCI/VHP.html. The Visible Human Project. February 1997.

[14]http://www.ncar.ucar.edu/METASCI/knee.gif. GIF image 517x517 pixels. February 1997.

[15]http://www.ncar.ucar.edu/METASCI/surgery.gif. GIF image 632x479 pixels. February 1997.

[16]Scheimann, Thomas, et al. Segmentation of the Visible Human for High Quality Volume Based Visualization. *Visualization in Biomedical Computing*. Lecture Notes in Computer Science: 1131, 13-22. 1996.

[17]Schiemann, Thomas, et al. The Visible Human within the VOXEL-MAN Framework. *Proc. Visible Human Conference*, Bethesda, MD. 1996.

[18]Schiemann, Thomas, et al. Generation of 3D Anatomical Atlases Using the Visible Human. *Proc. Computer Applications in Radiology*, Denver, CO. 1996.

[19]http://www.uke.uni-hamburg.de/Institutes/IMDM/IDV/VisibleHuman.html. Work with the VISIBLE HUMAN data set. February, 1997

[20]http://www.uke.uni-hamburg.de/Institutes/IMDM/IDV/VOXEL-MAN.html. VOXEL-MAN interactice 3D anatomical atlas. February, 1997

[21]http://www.uke.uni-hamburg.de/Institutes/IMDM/IDV/Articles/vbc96/f6_C08_C.jpg. JPEG image 501x369 pixels. February, 1997.

[22]http://www.crd.ge.com/esl/cgsp/projects/vm/slides/m1.jpg. JPEG image 640x480 pixels. February 1997.

[23]Wood, J.E., et al. Quantitation of Human Shoulder Anatomy for Prosthetic Arm Control - I. Surface Modelling. *Journal of Biomechanics.* 22:273-292. 1989.

[24]Ng Thow Hing, Victor. "A Biomechanical Musculotendon Model for Animating Articulated Objects". Master's Thesis, University of Toronto. 1994.

[25]Howard, Caleb. C.O.R.E. Digital Pictures. Electronic mail correspondence with the author. March 12, 1996 to April 17, 1996.

[26]Mack, Kevin. Digital Domain. Electronic mail correspondence with the author. March 12, 1996 to April 17, 1996.

[27]Muscling in on More Realistic Animations. *Computer Graphics World.* pp. 14-5, October 1995.

[28]Wood, Sam M. "Computer Graphics Modeling of the Musculo-Skeletal System of the Human Arm". Master's Thesis, The University of Texas at Austin. 1990.

[29]Marin-Risinger, Nena M. "Development of a Three-Dimensional Computer Graphics Surface Model of the Human Musculature of the Elbow Joint Complex". Master's Thesis, The University of Texas at Austin. 1995.

[30]Garner, Brian A. *cryo_edit* computer program for IRIX 4.0 OS. The University of Texas at Austin. 1996.

[31]Garner, Brian A. *cryo_recon* computer program for IRIX 4.0 OS. The University of Texas at Austin. 1996.

[32]Lorensen, W.E. and Cline, H.E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics.* v. 21, n. 4. July 1987.

[33]Garner, Brian A. *smoothbone* computer program for IRIX 4.0 OS. The University of Texas at Austin. 1996.

[34]Levine, D., et al. A Realistic, Dynamic Computer-Based Graphical Muscle Model. *16th Southern Biomedical Engineering Conference Proceedings.* Biloxi, MS. April 1997.

[35]Garner, Brian A. *fitpoint* computer program for IRIX 4.0 OS. The University of Texas at Austin. 1996.

[36]Levine, David A. *normalizer* computer program for IRIX 4.0 OS. The University of Texas at Austin. 1996.

[37]Levine, David A. *musclecad* computer program for IRIX 4.0 OS. The University of Texas at Austin. 1996.

[38]Schroeder, William J., et al. Decimation of Triangle Meshes. Computer Graphics. v. 26, n. 2. July 1992.

[39]Garner, Brian A. *selectplane* computer program for IRIX 4.0 OS. The University of Texas at Austin. 1996.

[40]Schneck, Daniel J. Mechanics of Muscle: Second Edition. New York University Press. New York, 1992.

[41]Hatze, H. Myocybernetic Control Models of Skeletal Muscle: Characteristics and Applications. University of South Africa Press. Pretoria, 1981.

[42]Crawford, Richard H. *Computational Geometry* class notes, The University of Texas at Austin. October 2, 1996.

[43]Rogers, David F., Adams, Alan J. Mathematical Elements for Computer Graphics. McGraw-Hill Publishing Company. New York, 1990.

VITA

David Asher Levine was born on [redacted] in [redacted], son of Sara Rae Seidman Levine and Richard Carl Levine, brother of Naomi Rena and Earl Isaac. After completing his work at Richardson High School, Richardson, Texas in 1991, he entered the University of Texas at Austin. He received a Bachelor of Science degree in Mechanical Engineering in the summer of 1995. Upon graduation, he immediately began his work at the University of Texas at Austin Graduate School. Mr. Levine will begin law school in the fall of 1997 at The George Washington University and plans to pursue a career as an attorney.

Permanent address: 7950 Woodstone Lane
Dallas, TX 75248

This thesis was typed by the author.