Anatomically Based Modeling

Jane Wilhelms and Allen Van Gelder*

University of California, Santa Cruz

Abstract

We describe an improved, anatomically based approach to modeling and animating animals. Underlying muscles, bones, and generalized tissue are modeled as triangle meshes or ellipsoids. Muscles are deformable discretized cylinders lying between fixed origins and insertions on specific bones. Default rest muscle shapes can be used, or the rest muscle shape can be designed by the user with a small set of parameters. Muscles automatically change shape as the joints move. Skin is generated by voxelizing the underlying components, filtering, and extracting a polygonal isosurface. Isosurface skin vertices are associated with underlying components and move with them during joint motion. Skin motion is consistent with an elastic membrane model. All components are parameterized and can be reused on similar bodies with non-uniformly scaled parts. This parameterization allows a non-uniformly sampled skin to be extracted, maintaining more details at the head and extremities.

CR Categories and Subject Descriptors: I.3 [Computer Graphics]; I.3.5 [Computational Geometry and Object Modeling]; I.3.7 [Three-Dimensional Graphics and Realism] Animation, Virtual Reality.

Additional Keywords: Human and Animal Modeling, Anatomically-Based Modeling.

1 INTRODUCTION

Humans, and other animals, are among the most important and interesting objects simulated using computer graphics, but they are also the most difficult to realistically model and animate. In general, computer graphics has achieved greater realism by developing methods that *simulate* the real world, rather than using ad hoc methods. As a step in this direction, we present a modeling and animation approach that is more closely based on anatomical principles than previously described methods. This model consists of individual muscles, bones, and generalized tissues covered by an elastic skin. The components mimic actual components of the animal body. In real animals, muscles stretch across joints to cause motion, and skin movements caused by underlying muscles can occur across a wide area. For simulated animals (and humans) to appear realistic, these widespread skin effects must be seen, and what better way to simulate these effects than to actually model individual muscles?

*Computer Science Dept, Univ. of California, Santa Cruz, CA 95064, USA. E-mail: {wilhelms,avg}@cs.ucsc.edu

Our modeling approach involves the following steps: (1) specify a body hierarchy and rest position; (2) design individual muscles, bones, and generalized tissues; (3) voxelize components into a 3D grid; filter; extract triangle-mesh skin; (4) map skin vertices parametrically from world space into the coordinate system of the nearest underlying component.

Animation involves repetition of the following steps: (1) specify motion at joints; (2) reposition and deform underlying components according to new positions; (3) map skin vertices back to world space; (4) apply iterative relaxation algorithm, adjusting skin vertices to achieve equilibrium of the elastic membrane forces. Table 2 shows performance times for these steps.

A longer version of this paper is available as a technical report [20]. Examples from our work on anatomically based modeling can be found on our web site www.cse.ucsc.edu/~wilhelms/fauna.

2 BACKGROUND AND RELATED WORK

Blinn's seminal work on implicit surface modeling included a "blobby man" made by extracting a surface from around an articulated skeleton [1]. Magnenat-Thalmann and Thalmann developed joint-local deformations (JDL's) where procedures are associated with each joint to simulate natural changes [10].

Chadwick *et al.* presented a method for layered construction of flexible animated characters [2] using *free-form deformations* [12]. The free-form deformation approach was also used by Komatsu [8] for skin. Mark Henne used a layered approach [6], in which *implicit fields* simulated body tissue. Singh *et al.* also used implicit functions to simulate skin behavior [13]. Turner *et al.* used an elastic skin model for character animation [14]. Skin wrinkling has been explored by Wu *et al.* [21]. Gourret *et al.* [5] used a finite element method to model the hand during grasping. None of these methods attempted to model individual three-dimensional muscles.

Chen and Zeltzer presented a biomechanically based muscle model using a finite element method to realistically simulate a few individual muscles without an overlying skin [3]. Another example from biomechanics is the work of Delp and Loan [4].

Recent work by Scheepers *et al.* is most closely related to ours [11]. Their emphasis is on modeling musculature, using a variety of geometric primitives.

The most anatomically detailed simulations have been done for the human face. Skin is generally modeled as a surface mesh whose points must move as expression changes. Physical simulation has been integrated into facial modeling by Lee *et al.* [9]. Koch *et al.* described a system for simulating facial surgery using finite element models [7]. None of these models used individual muscles with a physical presence to deform overlying tissue, though some do mimic individual muscle actions.

3 THE BASIC MODEL

Our body model uses a standard hierarchy of rigid segments connected by joints and emanating from a single root. The body consists of four types of materials. Individual *bones* are rigidly embedded in segments. Individual *muscles* are attached to bones.

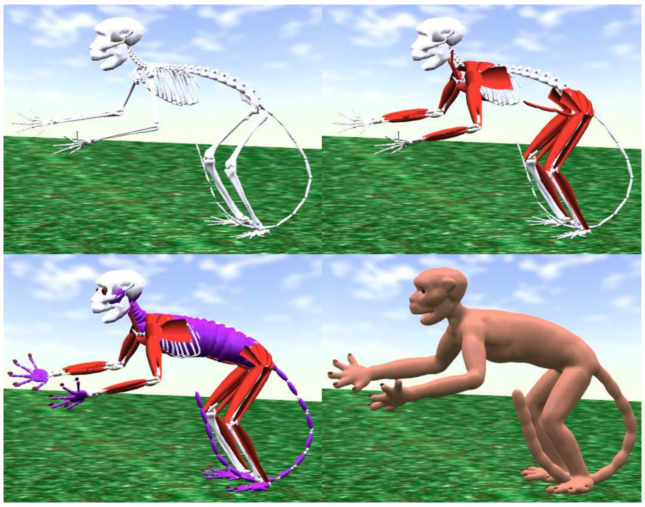

Figure 1: Anatomical components in rest posture: skeleton (white), muscles (red), generalized tissue (purple), and skin (lower right).

Generalized tissue gives shape to regions where detailed bones and muscles aren't used, and for features such as eyes and nails. An elastic overlying triangle-mesh *skin* is attached to underlying tissues with anchors, but adjusts in response to forces from neighboring skin vertices. The model is appropriate for any vertebrate; we illustrate it here with a monkey body model. Figure 1 shows the underlying components and the skin in the rest position of the body.

The monkey model has 85 segments, including all segments connected by major moving joints in a vertebrate body: skull, jaw, 44 vertebrae, pelvis, arms, legs, wrists, ankles, fingers, and toes. All joints are capable of three revolute degrees of freedom, but their range can be limited by a maximum and minimum angle. Also, one can designate "synergies", so that a requested rotation is distributed over a *series* of joints. Synergies help to control many degrees of freedom easily, particularly in the spine and hands. E.g., a 12 degree rotation of the thoracic region is interpreted as a one degree rotation at each thoracic vertebra. Each segment has a *basic segment size* based on the long diagonal of the segment bounding box in the rest position, which is used in parameterizing components.

The skeleton and generalized tissues are modeled as triangle meshes or ellipsoids. These components do not change shape during motion, but can change position relative to each other. Each has its own local coordinate system. When stored, the sizes and locations of bones and generalized tissues are parameterized by the basic segment size, so that the same components can be used in an individual where segments are different sizes.

The monkey skeleton consists of 88 individual triangle-mesh bones based on a human skeleton model from *Viewpoint DataLabs* and altered using the *SGI Alias/Wavefront* software to be more monkey-like. There are 68 ellipsoidal bones for the tail, hands, and feet. Generalized tissue is represented by 54 ellipsoids. Eyes and nails are also ellipsoids. (See Figure 1.)

4 MUSCLES

A typical skeletal muscle is an elastic, contractile material. In anatomical terminology it *originates* via tendons at fixed *origin* locations on one or more bones, passes across one or more joints, and *inserts* via tendons on fixed *insertion* locations on one or more other bones more distal to the body center. When the muscle contracts, and shortens, the bones to which the muscles are attached are pulled toward each other. The diameter and shape of the muscle changes depending on the relative positions of origins and insertions. In a real animal, muscle contraction causes joint motion. In our virtual animal, muscles change shape because of joint motion, to cause realistic skin deformations during animation. Initial work [19] used ellipsoids for muscles, which is adequate for those with a simple fusiform shape and a single origin and insertion. This model is insufficiently general for many muscles; e.g., only the lower arm muscles of the monkey model are ellipsoidal.

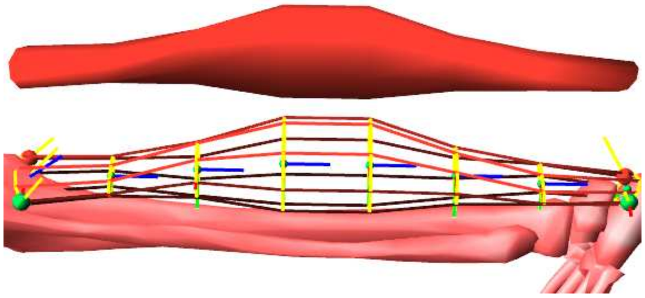

Figure 2: Typical default *deformed-cylinder* muscle, also illustrating anatomical terminology. The *proximal* direction is left, *distal* is right. The muscle is defined by two *origins* (red and green spheres at left) and two *insertions* (same at right). In the wireframe view below, eight yellow cross-sectional slices are connected by red edges to form a polygon mesh. The short blue and green lines are slice coordinate frame $Z-$ and $Y-$axes. The shaded polygon mesh is shown above.

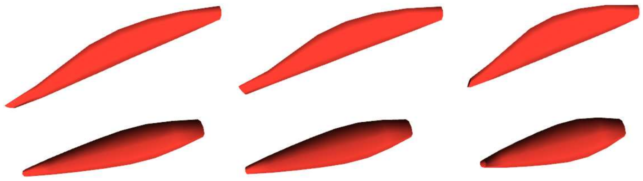

Figure 3: A (user-adjusted) deformed-cylinder muscle, seen from side and front at three different levels of contraction.

4.1 Deformed Cylinder Model for Muscles

Many muscles originate from or insert on more than one location. The various origins may be in different segments, and the same is true for insertions. To model such muscles, we have developed a muscle model based on a *deformed cylinder*. Our interests lie in producing relatively realistic animals that can be animated quickly and designed in a reasonable time frame. We have found this model to provide a good compromise between realism and speed.

The system generates, on request, a default deformed-cylinder muscle that is completely defined by two *origins* and two *insertions* (Figure 2). Origin and insertion points are described as three-dimensional locations on specific bones, parameterized by the size of the bounding box of the bone. In this way, the same muscle model can be used in different individuals. Origins must be proximal relative to insertions in the body hierarchy.

Each muscle is a discretized, deformed cylinder whose axis is a curve that proceeds from the midpoint of the origins to the midpoint of the insertions. Generally, the cylinder is discretized into 7 longitudinal *muscle sections* demarcated by 8 elliptical cross-sectional *slices*, as shown in Figure 2 (see also Figure 6).

The first elliptical slice lies between the two origins, and the last elliptical slice lies between the two insertions. The six intervening elliptical slices lie between and define the shape and longitudinal axis of the muscle. (There are no explicit tendons.) Each cross-sectional slice is then discretized into regularly spaced radial points, defining a planar polygon. Connecting the radial points between neighboring muscle cross-sections produces a polygon mesh, as shown in Figure 2.

There are approximately 80 major muscles that affect the shape of the arms, legs, and trunk (ignoring fingers and toes), most of which occur symmetrically on the right and left sides of the body [18]. The monkey model used here groups some of these muscles, divides a few, and ignores others. 40 deformed-cylinder muscles are used altogether.

The basic muscle definitions are provided to the program by an ascii file giving the muscle name, its origin and insertion bones, and the parameterized two origin and two insertion locations for each muscle. Origins and insertions can be interactively adjusted as needed for best effect, and the results saved.

4.2 Setting the Default Muscle Shape

The default muscle shape is automatically created based on the origins and insertions and can be altered interactively by the user. A *slice* coordinate frame is associated with each of the sliced segments of the muscle. The origins of these coordinate frames define the piecewise linear longitudinal axis of the muscle, and lie at the center of each slice ellipse. The X, Y, and Z axes of these slice frames are shown as red, green, and blue lines (respectively) in Figure 2. The XY planes of these slice coordinate frames define cross-sections of the muscle, and their Z-axis points along the longitudinal axis of the muscle, from muscle origins to muscle insertions. (We somewhat verbosely describe the (fixed) origin points of the muscle on bone as *muscle origins* and the origins of the slice coordinate frames as *coordinate frame origins* to prevent confusion in the different use of the word *origin*.)

The polyhedral vertices of the muscle's surface lie in the XY plane of each slice, arranged symmetrically around the slice coordinate frame origin, discretizing the ellipse. The number of vertices in each slice is under user control; the default number is 8. Figure 2 connects the muscle vertices within each slice with yellow lines, and muscle vertices between slices by reddish lines.

The locations of these frames and vertices are found as follows: First, all muscle origin and insertion locations are converted into the frame of the first muscle origin location's segment (i.e. the segment that contains the bone that the first muscle origin is attached to). The first muscle origin is just the one listed first in the muscle description file. This must be done anew each time any joint between origins and insertions changes.

Next, two end-slice coordinate frames are created. The midpoint of the two muscle origins becomes the origin of one end-slice frame, and the midpoint of the two insertions becomes the origin of the other. The lines between origins, and between insertions, define X-axes of end frames. The Z-axis of the origin slice coordinate frame is perpendicular to the X-axis and points outward from the origin bone. The Z-axis of the insertion slice frame is similarly perpendicular to the insertion frame X-axis, but points into the insertion bone. These Z-axes are found by averaging the outward normal vectors to the bones at the origin locations, and the inward normal vectors to the bones at the insertion locations. The Y-axes for these coordinate frames complete right-handed systems.

Now, the planes of the intermediate slices are calculated. Initially, they are arranged along a straight line between the origins of the two end coordinate frames described above, and equidistant from each other along this line. The intermediate slice frames have their Z-axes aligned along the line between end coordinate frame origins, and their X-axes are interpolated between the X-axes of the end frames. Once the thickness of the muscle slices is found, the origins of the intermediate frames are slightly displaced in the negative Y direction (outward from the underlying bone), to compensate for the greater thickness of the muscle near its center,

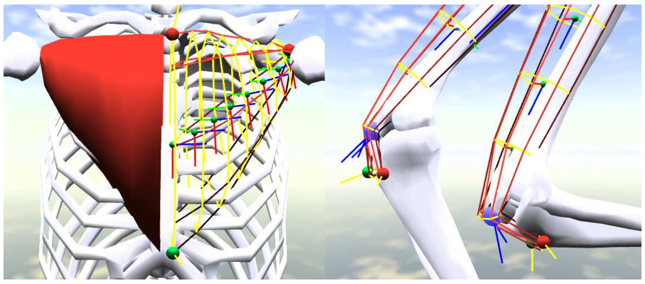

Figure 4: A. (at left) The right pectoralis major muscle shown shaded and the left in outline, illustrating a non-default muscle shape set interactively by the user. B. (at right) The quadriceps femoris muscles illustrating the use of a pivot. The right lower leg is in the rest position, and the left lower leg is flexed showing the muscle bending around its pivot.

creating a curved path from origin to insertions.

Finally, the vertices around each slice frame, which define the actual polygon mesh muscle surface, are calculated. These points are radially located around the coordinate frame origin of the slice in its XY-plane. The two end coordinate frames are constrained so that the width of the muscle (in X) is the distance between muscle origin points, and muscle insertion points, respectively. The vertex locations for the intermediate slices are scaled in X and Y to produce a fusiform shape, larger in the middle slices than in the end slices, and larger across (in X) than in thickness (in Y).

4.3 Non-Default Muscle Shapes

The user can interactively alter the size of a muscle, and the orientation and location of slice coordinate frames, and the locations of origins and insertions. Figure 4 shows two non-default muscle shapes, which illustrate the topics of this section.

First, the interface provides a facility to change the scale factors defining the X-width and Y-thickness of a muscle or any individual muscle slice. The cross-section of the muscle is always elliptical. Muscles such as the gastrocnemius in the leg, the biceps in the arm, and the sterno-cleido-mastoid in the neck are well represented by slightly scaling the default muscle shape (Figure 3).

Other muscle modification parameters alter location and orientation of the muscle slice coordinate frames. Any intermediate slice can be interactively translated from the default position. The orientation of the slice frames can also be rotated, in X, Y, or Z for intermediate frames, or in X alone for end frames. Figure 4.A. shows the pectoralis major muscles in the chest. Note that the shape of the muscle is more triangular than fusiform, the muscle is quite flat, and the slice coordinate frames are in an arc around the ribs, not in a straight line from origin to insertion.

Another modification method is to designate a *pivot* coordinate frame. In this case, the path of the muscle is not from muscle origins to insertions, but from origins to pivot, then from pivot to insertions. Pivots are helpful in modeling muscles such as the quadriceps femoris, which runs across the front of the thigh and bends over the knee (Figure 4.B). The origins of pivot coordinate frames are shown as blue spheres in outline figures. Generally, pivots can reduce the problem of interpenetration of the muscle with the material beneath it. (See Section 7 for further discussion of this problem.)

The pivot frame is defined relative to a specific segment, usually that of the segment containing the bone to which the first muscle origin is attached. Its location and orientation are under user control. A specific slice uses the pivot frame as its slice frame, and other frames arrange themselves on the line to the pivot frame, or from the pivot frame, depending on their position relative to the

2 origin locations parameterized on bones (x,y,z)
2 insertion locations parameterized on bones (x,y,z)
x,y scale of each muscle shape slice
x,y,z translate of slice coordinate frame from default
x,y,z rotation of slice coordinate frame from default
whether pivot is on, off, or variable
which muscle shape slice acts as the pivot
location of the origin of the pivot coordinate frame
orientation of the pivot coordinate frame

Table 1: Muscle parameters controllable by the user.

pivot slice. The pivot frame can be permanently in effect (always used), or it can be used intermittently depending on whether the joint angles make it natural to use it. In Figure 4.B, the pivot frame is the seventh frame, as the bend occurs near the very end of the muscle. The right leg is in its rest position, and the left lower leg is flexed, showing the muscle bending around its pivot.

Table 1 summarizes the user-controllable parameters for muscle design. Nearly all the muscles in the monkey model shown in Figure 1 used some non-default parameters.

4.4 Muscle Animation

Muscle animation involves the automatic recalculation of the muscle shape whenever a joint lying between muscle origins and insertions moves. The muscle origins and insertions are converted to the frame of the first origin, given the new joint positions. A default shape is found, and automatically adjusted using the user-specified non-default muscle parameters described in the previous section.

Finally, the muscle width and thickness are scaled to maintain approximately constant muscle volume during joint changes. The *rest length* of the muscle is the distance from the midpoint of the muscle origins, through the pivot, if present, to the midpoint of the insertions, while the muscle is in the resting position, as designed. A *present length* is similarly calculated for when the muscle is repositioned due to joint changes. The width and thickness of the internal slices of the muscle (not the end slices) are scaled by $\sqrt{\textit{rest length} \,/\, \textit{present length}}$. This increases the cross-sectional area if the muscle shortens, and decreases it if it lengthens. Volume is preserved exactly in regions between parallel slices, and is changed as a second order effect in regions between two nonparallel scaled slices. However, end slices do not change shape, so regions involving an end slice will normally vary in volume. In any case, exact volume preservation of muscles is not biologically justified. Isometric deformations provide a case in point. Future work should address a more flexible and more biologically sophisticated method for adjusting muscle volume.

The new muscle shape is stored and reused for display and skin adjustment purposes until another joint change necessitating recalculation occurs. Figure 3 shows a muscle from the front and side at three levels of contraction.

5 SKIN

The skin is an elastic triangle-mesh surface that is attached to underlying components but can move relative to them; i.e., a separate, controllably loose layer over underlying components. The initial creation of the surface is based on fairly standard implicit surface techniques, and is summarized below. The novel contribution of this paper is the methodology for skin deformation in response to deformation of the underlying tissue, described in subsequent subsections.

The region around the animal in the rest position is voxelized to create a three-dimensional discrete grid of points. Values at points define an artificial *density function* that is positive if the point is inside any body part and zero otherwise. (In addition, certain artificial components may have a *negative* density to create a repulsion effect, discussed below.) Then the initial density function is filtered with a Gaussian kernel, whose width is under user control. The user chooses a threshold, and an isosurface of the filtered density is extracted as a triangulated surface mesh.

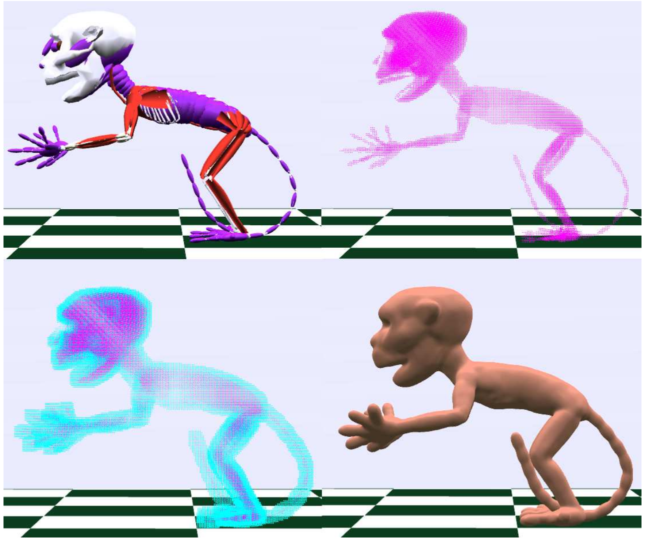

Figure 5: Voxelization and skin extraction illustrated: underlying parts with extremities enlarged (upper left); initial "density" function (upper right); after filtering (lower left); extracted skin (lower right).

Figure 5 shows these steps. Upper left shows the underlying components; the head and extremities are enlarged for better skin detail in those areas, as described in Section 5.3. Upper right is a voxel grid whose maximum resolution is 200, showing interior grid points in magenta. Grid points outside the body are not shown. The purple spheres near the eyes contribute negative values to the field, making deep orbits for the eyes. Lower left is the voxel grid after filtering, showing the grid points with positive field values in cyan. The Gaussian filter kernel had a standard deviation of about 3 voxel separations. Lower right is the extracted skin.

5.1 Anchoring Skin

After surface extraction, in a second stage called *anchoring*, each vertex in the triangle-mesh skin is associated with the closest underlying body component (muscle, bone, or generalized tissue).

The *anchor* of a particular skin vertex is the nearest point on its underlying component. More important for animation is the *virtual anchor*, which is the initial position of a skin vertex relative to its underlying component. This is the position of the vertex when the skin was extracted in the animal's rest position. The anchors and virtual anchors are stored parameterized in the local space of the component. If shape changes occur in the underlying component, they are transmitted through the anchors and virtual anchors, to affect the skin vertices correspondingly. Each skin vertex is considered to be connected to its virtual anchor by a spring of rest length zero, and a specified spring stiffness. (See Section 5.2.)

Anchoring refers to the process of finding the nearest underlying component of each skin vertex, converting that skin vertex to a parameterized local location relative to the component, and storing this local position of the skin vertex as its virtual anchor. Anchoring skin to ellipsoids was previously described [19]. To anchor skin to triangle-mesh bones, skin vertices are converted into the coordinate

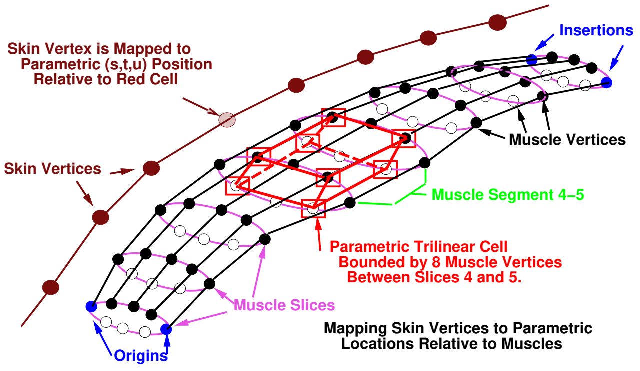

Figure 6: Illustration of mapping skin vertices to parametric trilinear functions over muscle segments. The lightest skin vertex lies between muscle slices 4 and 5, and is mapped into an (s, t, u) coordinate system defined by the eight muscle vertices shown in red.

system of the bone and scaled by the size of its bounding box in each dimension. Anchoring points to deformed-cylinder muscles is a more interesting problem, described next.

The skin vertex is associated with a section of the muscle that lies between two muscle slices. In Figure 6, the lighter skin vertex lies nearest to the muscle segment between slices 4 and 5, and will be mapped to a parametric location in this region. One cannot simply map the skin vertices into the frame of single slice, because bumps appear in the skin when the muscles change shape, due to the abrupt changes in mapping from one slice frame to the next.

Rather, we define a *parametric trilinear transformation* (more precisely named "tri-affine") over the space between the planes of the two slices in rest position, and assign parameters to the virtual anchor in accordance with the inverse of this transformation. A parametric trilinear transformation is a three-vector of independent trilinear functions, each of which maps parameter-space (s, t, u) into physical-space (x, y, z). The parametric trilinear transformation maps a unit cube into a warped cell such that edges in the original cell remain as straight lines. Parametric trilinear transformations are defined on adjacent cubes so that they map each shared corner of their cubes to the same point. This ensures that they combine to make a $C0$ continuous transformation. Related ideas are found in free-form deformations of computer modeling [12].

The parametric trilinear transformation for a muscle segment is found by taking four corresponding muscle vertices from each of the slice planes that bound the segment. The four from the "proximal" slice will be the image of a parametric unit cube's $u = 0$ face, while the four from the "distal" slice will be the image of the same unit cube's $u = 1$ face. Joining corresponding vertices of the two slices makes the warped image of the unit cube. In Figure 6 such a warped cell is shown in red. These eight vertices provide the 24 unknowns necessary to specify the coefficients of the parametric trilinear transformation. Each skin vertex associated with a particular muscle segment is then mapped inversely to a parametric position (s, t, u) within the domain of the parametric trilinear transformation defining that segment. The inverse mapping has no closed form, and requires 3D Newton-Raphson iteration [15]. The (s, t, u) parameterized position is the virtual anchor for that skin vertex. This expensive operation needs to be done only once per vertex, during anchoring.

When the body is moved, new world space positions are calculated for the slices (see Section 4.4). Then for each adjacent pair of slices a *new* parametric trilinear transformation is defined. (This process is not expensive, compared to calculating general inverse mappings.) Virtual anchor points associated with this segment are mapped from (s, t, u) to world space, using the new parametric trilinear transformation, in the *forward* direction. Each virtual anchor provides an initial skin position for its corresponding skin vertex, for this body configuration.

Figure 7: This figure illustrates the concepts of anchors, virtual anchors, and elastic relaxations, on a shoulder and raised left arm. It corresponds to the center third of Figure 8. Skin vertices are connected by brown edges to form a triangle mesh. In the left image, skin vertices coincide with their virtual anchors, as no elastic relaxation has been done. In the right image, after elastic relaxation, the virtual anchor positions are unchanged, while the skin vertices have been redistributed more uniformly. In both images yellow lines connect skin vertices to their muscle anchors. In the right image red lines connect the skin vertices to their virtual anchors, showing the displacement necessary to equalize spring forces.

Figure 7 illustrates these concepts on the left shoulder and arm of the monkey. The arm is raised away from the rest position, with consequent deformation of nearby muscles. Virtual anchor points have been remapped from parameter space to world space, taking into account both the rigid motion and deformation of their respective muscles.

5.2 The Elastic Model

To simulate an elastic membrane each edge of the skin triangle mesh is considered to be a spring with a certain rest length and stiffness. Together with other forces and constraints in the system, these springs are brought into equilibrium by means of a series of relaxation operations. The initial skin positions, from which relaxation commences, are provided by the positions of the virtual anchor, as described in Section 5.1. Relaxation operations continue iteratively until a user-defined convergence tolerance is reached, or a user-defined maximum number of iterations has occurred.

The spring stiffness coefficient (stiffness, for short), for edges between skin vertices is denoted as k_e. This value is automatically calculated as

$$k_e(v, v_j) \quad = \quad \frac{a_1 + a_2}{len^2} \qquad (1)$$

where *len* is the length of the edge between skin vertices v and v_j, and a_1 and a_2 are the areas of the two triangles sharing the edge, all calculated once in the rest position. This formula provides a more accurate model of uniformly elastic skin than would uniform stiffness for all springs [17]. The formula used is simplified from the general case by assuming the Poisson coefficient for skin is 0. (A more complicated formula with the more realistic Poisson coefficient of 0.25 did not produce observable differences.)

The spring stiffness for the edge between the skin vertex and its virtual anchor is denoted as k_a. This "edge" has zero rest length. For each skin triangle, the force resulting from pulling the skin away from the underlying material is assumed to be proportional to the area of the triangle (as is standard for "body forces" in elasticity), and this area is distributed equally to the three skin vertices incident upon the triangle. Consequently,

$$k_a(v) \quad = \quad C_a \sum_i \frac{a_i}{3} \qquad (2)$$

where a_i denotes the area of the i-th triangle incident upon the skin vertex v, and the sum is over all such triangles. The coefficient C_a is a proportionality constant to control the relative strengths of skin-skin springs (k_e's) and skin-virtual-anchor springs (k_a's). For the monkey model, the anchor spring stiffness is generally scaled by $C_a = 0.10$, which allows the skin to slide readily over the underlying parts.

If constant spring stiffness coefficients are used instead of the geometrically determined values in Eqs. 1–2), then we observed that the equilibrium position of the skin appears irregularly stretched.

To find the change in position to be applied to a vertex due to the influence of a connected edge, the elastic force vector is calculated. First, the vector $w_j = v_j - v$ is defined, where v is the vertex being analyzed, and v_j is the vertex at the far end of the edge. The length of w_j is the present length of the edge. The rest length of the edge is subtracted from the present length, giving the *length excess*, which may be negative. Now the spring stiffness, $k_e(v, v_j)$ may be modified, based on the *length excess*, providing a form of nonlinear spring (see below). The *length excess* is multiplied by the (possibly modified) spring stiffness for the edge to give the scalar value of elastic force due to this edge. The direction of the elastic force is just the direction of w_j. Thus the elastic force is toward v_j if the *length excess* is positive, and away from v_j if it is negative.

For each vertex v, the sum of the elastic forces due to each of its edges to other skin vertices v_j, and due to the edge to its virtual anchor, defines the *net elastic force* acting on this vertex. This net force is divided by the sum of the spring stiffness coefficients that contributed to the net force, giving the *elastic relaxation vector* for v. (This denominator is conservatively large, as various forces tend to cancel each other; however it stabilizes the relaxation.) All skin vertex positions are translated by their relaxation vectors in one round of relaxation. The relaxations are iterated until the maximum relaxation vector is below the user-specified threshold, or the user-specified maximum number of iterations occurs.

Figures 7 and 8 show the effect of elastic relaxation on the skin. The smooth redistribution seen in the right images is important to achieve a natural appearance for fur [16] or skin with markings.

A few other parameters can be applied to the skin. First, we want the skin to *pull* from a stretched position toward its rest state more strongly than it *pushes* back when it is compressed. Therefore, a user-controlled scale factor can be applied to scale down the spring force if the present length is less than the rest length. Typically this scale factor is 0.1.

Second, the skin can appear more smooth if we adjust the model so that the skin is slightly stretched in its extracted configuration. I.e., the *rest length* of the edge is taken to be some percentage of the measured *default length* of the edge when extracted. Suppose the user has chosen 90% for this parameter, and 0.1 for the above "pushing back" parameter. Then k_e, the spring stiffness will be modified by the following continuous function of *length excess*, discussed above. If *length excess* is positive, k_e is used as is (100%). If *length excess* is more negative than $-0.1\times$(rest length), then $0.1k_e$ is used (10%). If *length excess* is between these bounds, the multiplier for k_e is interpolated between 10% and 100%.

Finally, a collision influence can be applied to prevent the skin from sinking into underlying components. Each skin vertex is prohibited from penetrating a sphere whose center is somewhat below the anchor point in the underlying component. The surface of this sphere is tangent to the tangent plane of the skin at the virtual anchor point. I.e., if the calculated new position of the skin vertex penetrates such a sphere, a repelling force is activated to displace the skin vertex outward toward the sphere surface.

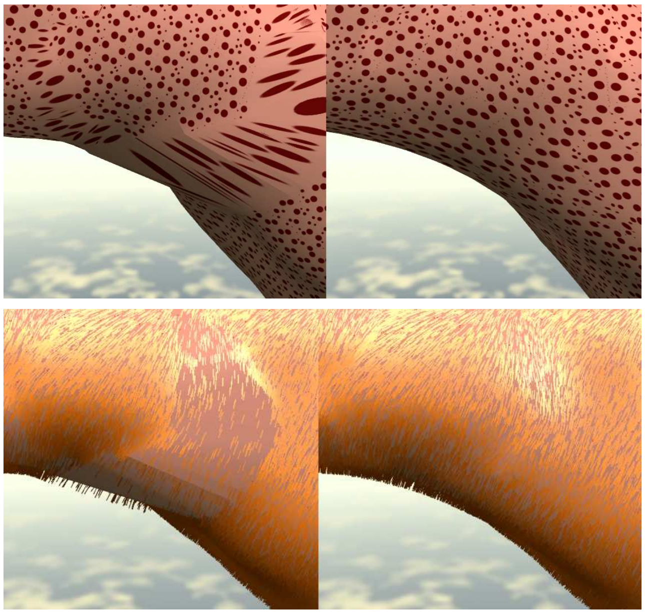

Figure 8: This figure illustrates the effect on skin of elastic relaxation. At left, no elastic relaxation has occurred, and skin vertices coincide with their virtual anchors. At right, 30 relaxations have occurred, the skin has largely stabilized, and skin vertices are more uniformly distributed (see wireframe detail view in Figure 7). Red dots appearing in the top images are produced by texture maps on the triangles of the skin surface, which vary in size and shape, even in the rest position. Dots are proportional in area to their triangles, and are circular in the rest position. Before relaxation (left) unequal distortions of the red dots are apparent, while after relaxation (right) they exhibit smoothly varying deformations. Bottom images show the effect on fur, which is attached to skin triangles. Before relaxation, the unequal skin stretching produces gaps and irregularities; after relaxation, fur and skin appear normal.

This approach only handles collisions between the skin and its nearby underlining tissues. A second collision type occurs when a joint is bent so that skin on either side of the joint meet. In this case interpenetration can occur, but it is hidden within the folds of the skin. A third collision type occurs when skin from one part of the body presses on another. This would require modeling the influence of external collision forces on the skin, which we do not do.

5.3 Non-Uniform Scaling and Re-usability

Because all components are parameterized, they can be re-used to a large extent in different individuals. The basic reference for all components is ultimately the basic segment size (see Section 3). If a model with larger or smaller segments is used, components compensate automatically for the change in size. It is also possible to scale the basic size by applying an extra scale factor to each segment when the body is read in. In this way, we created the model shown in Figure 5, with extra large extremities. By voxelizing such a model, more detail is preserved in the hands and feet. The skin is then saved, and can be re-used on a model with more normal-sized segments. All the monkeys shown in this paper used such scaling. Because of the similarity between vertebrates, many of the components of one model can be used for other species as well. It is also possible to scale the thickness of the skin, by moving the virtual anchors further from their underlying components.

Figure 9 shows the effects of scaling. These three monkeys were made from the identical underlying components and skin as all the other monkeys in the paper but in the left image the arms are

Figure 9: Three monkeys were made from the identical underlying components and skin, but with long arms (left), thick skin (middle), and long legs (right).

lengthened, in the right image the legs are lengthened, and in the center the skin thickness is increased to create a plump monkey.

6 RESULTS AND DISCUSSION

The monkey model contains 85 body segments, 156 bones, 52 muscles, and 54 generalized components. The skin was voxelized over a grid whose maximum dimension was 225. Internal points were assigned 200 as density, outside points were 0. It was filtered by a Gaussian kernel with standard deviation 3.2. The isosurface threshold was 35. There are about 75,000 skin vertices and about 150,000 skin triangles. Table 2 gives times for various steps.

Figure 10 shows a selection of monkey images in various positions. The "mohawk" fur [16] is a whimsical addition to make the monkey seem less bald without obscuring the skin deformations that we wish to emphasize.

The parameterized muscle shape model is fast and easy to change, and provides a good approximation to the shape of most muscles (but see below). The monkey model could be improved by making more muscles. There can be problems in certain extreme positions. However, as the images and animations show, the body looks good over a wide range of motions.

The most time-consuming part of creating the model was to model individual muscles. Although each change is easy, a tedious trial-and-error process was needed to customize difficult muscles. Knowledge of anatomy is helpful, although students with no background in anatomy were able to perform many modeling tasks with light supervision of a trained biologist, and appropriate texts on anatomy (e.g., [18]).

An alternative to adjusting a permanent skin in response to underlying tissue deformations would be to re-extract skin over new positions. However, this would raise questions about consistency from one position to the next. For example, how would a texture be associated with the skin that appears to shift realistically?

The current skin model produces a smooth surface that is visually close to equilibrium after about 30 iterations; after this, changes from one iteration to the next are barely noticeable. Even after 5 or 10 iterations, acceptable convergence is found in many cases.

Steps Done Once to Create Skin	
Voxelization (Max. Dim. 225)	185
Filtering (3.2 std.dev. kernel)	160
Isosurface Extraction	16
Anchoring	645
Steps Required if Joints Moved	
Redraw of Underlying Components	Less than 1
Skin Repositioning - 0 relaxations	3
Skin Repositioning - 10 relaxations	9
Skin Repositioning - 30 relaxations	25

Table 2: Elapsed times in seconds for various steps, on an SGI with four 150-MHz R4400 processors, using shared-memory multiprocessing.

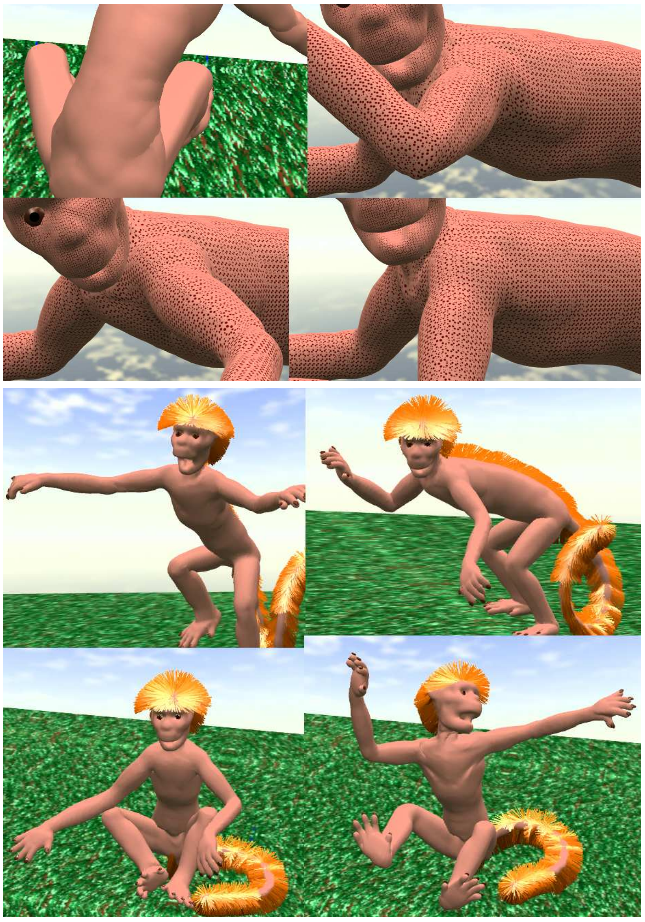

Figure 10: A selection of monkey images. Notice the effect of individual muscles, and the ability of the model to simulate both stretching and folds.

7 FUTURE WORK

This remains a very approximate model when compared to actual animal anatomy. It is a reasonable compromise between detailed realism and acceptable modeling and animation speed, and is a considerable step forward from ellipsoidal models.

The major next step we envision is underlying components that interact with each other, and do not interpenetrate. This will allow more detailed, space-filling components to be created, which reposition themselves based on influences from neighboring components. Generalized tissue should also be extended to accommodate shape changes, as muscles do now. While these additions will be more expensive, the minimal computational cost of adjusting the present underlying components suggest it will still be feasible in a fairly interactive system.

It would be a relatively simple matter to have muscles pull on their bones and implement a real physically based simulation of joint motion. This could be useful for educational purposes or for research in biomechanics. However, the control problems of physically simulating general realistic motion using contraction of individual muscles is really beyond present capabilities.

It would be desirable to have more detail in the face than provided by the non-uniform scaling. Because the skin is initially a separate polygonal layer that is then connected to the underlying components, it would be possible to connect a triangle-mesh skin from elsewhere to underlying components. For example, a digitized head model or a model extracted from CT scans could be connected to the underlying parts. This would require appropriate positioning and scaling.

The skin and underlying components should also react to outside forces, such as gravity, and detect and respond to all collision types.

8 CONCLUSIONS

This paper describes a new and improved modeling and animation approach for animals and humans that is based on actual three-dimensional representations of individual body components such as bones, muscles, and miscellaneous tissue, covered by a skin. We believe this is the most natural approach to use for creating realistic animals and humans. The scheme is a good compromise between realism and complexity, and can be displayed and animated interactively. We believe the approach can be extended to produce much greater realism at an acceptable computational cost.

Acknowledgments

List processing software by Yumi Tsuji was used in this software package. Marlon Veal helped with programming. Research supported by a gift from Research and Development Laboratories, and by NSF Grant CDA-9115268.

References

[1] James F. Blinn. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics*, 1(3):235–256, July 1982.

[2] John E. Chadwick, David R. Haumann, and Richard E. Parent. Layered Construction for Deformable Animated Characters. In *Computer Graphics (SIGGRAPH 89 Conference Proceedings)*, volume 23 of *Annual Conference Series*, pages 242–252. Addison Wesley, August 1989.

[3] David T. Chen and David Zeltzer. Pump It Up: Computer Animation Based Model of Muscle Using the Finite Element Method. In *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, volume 26, pages 89–98. Addison Wesley, July 1992.

[4] Scott L. Delp and J. Peter Loan. A Graphics-based Software System to Develop and Analyze Models of Musculoskeletal Structures. *Computers in Biology and Medicine*, 25(1):21–34, 1995.

[5] Jean-Paul Gourret, Nadia Magnenat-Thalmann, and Daniel Thalmann. Simulation of Object and Human Skin Deformations in a Grasping Task. In *Computer Graphics (SIGGRAPH 89 Conference Proceedings)*, volume 23, pages 21–30. Addison Wesley, July 1989.

[6] Mark Henne. A Constraint-Based Skin Model for Human Figure Animation. Master's thesis, University of California, Santa Cruz, Santa Cruz, CA 95064, June 1990.

[7] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Buerin, G. Fankhauser, and Y. I. H. Parish. Simulating Facial Surgery Using Finite Element Models. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 421–428. ACM SIGGRAPH, Addison Wesley, August 1996.

[8] K. Komatsu. Human Skin Model Capable of Natural Shape Variation. *The Visual Computer*, 4(3):265–271, 1988.

[9] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 55–62. ACM SIGGRAPH, Addison Wesley, 1995.

[10] Nadia Magnenat-Thalmann and Daniel Thalmann. Human Body Deformations Using Joint-Dependent Local Operators and Finite Element Theory. In N. Badler, B. Barsky, and D. Zeltzer, editors, *Making Them Move*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1991.

[11] Ferdi Scheepers, Richard E. Parent, Wayne E. Carlson, and Stephen F. May. Anatomy-based modeling of the human musculature. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, August 1997.

[12] Thomas W. Sederberg and Scott R. Parry. Free-form deformations of solid geometric objects. In *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, volume 20, pages 151–160. Addison Wesley, August 1986.

[13] Karansher Singh, Jun Ohya, and Richard Parent. Human figure synthesis and animation for virtual space teleconferencing. In *Proceedings of the Virtual Reality Annual International Symposium '95*, pages 118–126, Research Triangle Park, N.C., March 1995. IEEE Computer Society Press.

[14] R. Turner and D. Thalmann. The Elastic Surface Layer Model for Animated Character Construction. In N. M. Thalmann and D. Thalmann, editors, *Proceedings of Computer Graphics International '93*, pages 399–412, Lausanne, Switzerland, June 1993. Springer-Verlag.

[15] Allen Van Gelder and Jane Wilhelms. Interactive Animated Visualization of Flow Fields. In *1992 Workshop on Volume Visualization*, pages 47–54, Boston, Mass., October 1992. ACM.

[16] Allen Van Gelder and Jane Wilhelms. An Interactive Fur Modeling Technique. In *Proceedings of Graphics Interface*, May 1997.

[17] Allen Van Gelder and Jane Wilhelms. Simulation of Elastic Membranes and Soft Tissue with Triangulated Spring Meshes. Technical Report UCSC-CRL-97-12, CS Dept., University of California, 225 A.S., Santa Cruz, CA 95064, January 1997.

[18] Marvalee H. Wake, editor. *Hyman's Comparative Vertebrate Anatomy*. University of Chicago Press, Chicago, Illinois, third edition edition, 1979.

[19] Jane Wilhelms. Animals with Anatomy. *IEEE Computer Graphics and Applications*, 17(3):22–30, May 1997.

[20] Jane Wilhelms and Allen Van Gelder. Anatomically Based Modeling. Technical Report UCSC-CRL-97-10, CS Dept., University of California, 225 A.S., Santa Cruz, CA 95064, April 1997.

[21] Yin Wu, P. Kalra, and N. M. Thalmann. Simulation of Static and Dynamic Wrinkles of Skin. In *Proceedings of Computer Animation '96*, pages 90–97, Geneva, Switzerland, June 3–4 1996. IEEE Computer Society Press.