Eurographics/SIGGRAPH Symposium on Computer Animation (2003)
D. Breen, M. Lin (Editors)

Finite Volume Methods for the Simulation of Skeletal Muscle

J. Teran,[1] S. Blemker,[2] V. Ng Thow Hing,[3] and R. Fedkiw[4]

[1] Stanford University, jteran@stanford.edu
[2] Stanford University, ssblemker@stanford.edu
[3] Honda Research Institute USA, vng@honda-ri.com
[2] Stanford University, fedkiw@cs.stanford.edu

Abstract

Since it relies on a geometrical rather than a variational framework, many find the finite volume method (FVM) more intuitive than the finite element method (FEM). We show that the FVM allows one to interpret the stress inside a tetrahedron as a simple "multidimensional force" pushing on each face. Moreover, this interpretation leads to a heuristic method for calculating the force on each node, which is as simple to implement and comprehend as masses and springs. In the finite volume spirit, we also present a geometric rather than interpolating function definition of strain. We use the FVM and a quasi-incompressible, transversely isotropic, hyperelastic constitutive model to simulate contracting muscle tissue. B-spline solids are used to model fiber directions, and the muscle activation levels are derived from key frame animations.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation; I.3.5 [Computer Graphics]: Physically based modeling

1. Introduction

The pioneering work of Lasseter[18] on applying the principles of traditional animation to computer graphics emphasizes squash and stretch, timing, anticipation, follow through, arcs, and secondary action which all appeal to the use of physics based animation techniques. A variety of authors have worked to incorporate ideas such as these into their animations, e.g. Neff and Fiume[24] incorporated tension and relaxation into the animation of an articulated skeleton. Moreover, when considering the difficulties, such as collapsing elbows[19], associated with applying free form deformations[28] or related techniques to shape animation, one draws the conclusion that physics based simulation of muscle and fatty tissue should be the ultimate goal. Unfortunately, progress toward this goal has been rather slow due to the high cost of FEM and the poor quality of volumetric mass-spring models.

Significant effort has been placed into accelerating FEM calculations including recent approaches that precompute and cache various quantities[23], modal analysis[16], and approximations to local rotations[22]. In spite of significant effort into alternative (and related) methods for the robust simulation of deformable bodies, FVM has been largely ignored. Two aspects of FVM make it attractive. First, it has a firm basis in geometry as opposed to the FEM variational setting. This not only increases the level of intuition and the simplicity of implementation, but also increases the likelihood of aggressive (but robust) optimization and control. Second, there is a large community of researchers using these methods to model solid materials subject to very high deformations. For example, Caramana and Shashkov[5] used the FVM with subcell pressures to eliminate the artificial hour-glass type motions that can arise in materials with strongly diagonally dominant stress tensors, such as incompressible biological materials.

FEM has many attractive features that make it appealing to the engineering community, e.g. a solid theoretical framework for proving theorems and the ability to extend it to higher order accuracy. However, in graphics, visual realism is more important than the convergence rate, and thus the focus is on simulating a large number of cheap first order accurate elements rather than fewer more expensive higher order accurate elements. In fact, the same can be said for many engineering applications where quasi-static and other approximations may be made degrading the accuracy but allowing for the simulation of more elements.

We are particularly interested in the simulation of both active and passive muscle tissue. Biological tissues typically undergo large nonlinear deformations, and thus they create a stringent test for any simulation method. Moreover, they tend to have complex material models with quasi-incompressibility, transverse anisotropy, hyperelasticity and both active and passive components. In this paper, we use a state of the art constitutive model, B-spline solids to represent fiber directions, and derive muscle activations from key frame animation.

2. Related Work

Terzopoulos et al.[31, 30] simulated deformable materials including the effects of elasticity, viscoelasticity, plasticity and fracture. Although they mentioned that either finite differences or FEM could be used, they seemed to prefer a finite difference discretization. Subsequently, Gourret et al.[12] advocated FEM for simulating a human hand grasping a ball, and since then a number of authors have used the FEM to simulate volumetric deformable materials.

Chen and Zeltzer[6] used FEM, brick elements, and the constitutive model of Zajac[35] to simulate a few muscles including a human bicep. Due to computational limitations at the time, very few elements were used in the simulation. Wilhelms and Van Gelder[34] built an entire model of a monkey using deformed cylinders as muscle models. Their muscles were not simulated but instead deformed passively as the result of joint motions. Scheepers et al.[27] carried out similar work developing a number of different muscle models that change shape based on the positions of the joints. They emphasized that a plausible tendon model was needed to produce the characteristic bulging that results from muscle contraction. A recent trend is to use FEM to simulate muscle data from the visible human data set[36, 14].

In order to increase the computational efficiency, a number or authors have been investigating adaptive simulation. Debunne et al.[8] used a finite difference method with an octree for adaptive resolution. This was later improved using a finite volume integration technique to approximate the Laplacian and the gradient of the divergence operators.[10] We take a very different approach, using FVM to directly compute the stress based force on the nodes achieving a rather simple and intuitive method that trivially extends to arbitrary constitutive models. Debunne et al.[9] used FEM with a mutiresolution hierarchy of tetrahedral meshes, and Grinspun et al.[13] refined basis functions instead of elements.

3. Geometric Calculation of Strain

A deformable object is characterized by a time dependent map ϕ from undeformed material coordinates $\mathbf{X}$ to deformed spatial coordinates $\mathbf{x}$. We use a tetrahedron mesh and assume that the deformation is piecewise linear, which implies $\phi(\mathbf{X}) = \mathbf{F}\mathbf{X} + \mathbf{b}$ in each tetrahedron. The Green strain is defined as $\mathbf{G} = 1/2\left(\mathbf{F}^T\mathbf{F} - \mathbf{I}\right)$.

For simplicity, consider two spatial dimensions where each element is a triangle. Figure 1 depicts a mapping ϕ from a triangle in material coordinates to the resulting triangle in spatial coordinates. We define edge vectors for each triangle as $\mathbf{d}_{m_1} = \mathbf{X}_1 - \mathbf{X}_0$, $\mathbf{d}_{m_2} = \mathbf{X}_2 - \mathbf{X}_0$, $\mathbf{d}_{s_1} = \mathbf{x}_1 - \mathbf{x}_0$ and $\mathbf{d}_{s_2} = \mathbf{x}_2 - \mathbf{x}_0$. Note that $\mathbf{d}_{s_1} = \left(\mathbf{F}\mathbf{X}_1 + \mathbf{b}\right) - \left(\mathbf{F}\mathbf{X}_0 + \mathbf{b}\right) = \mathbf{F}\mathbf{d}_{m_1}$ and likewise $\mathbf{d}_{s_2} = \mathbf{F}\mathbf{d}_{m_2}$ so that $\mathbf{F}$ maps the edges of the triangle in material coordinates to the edges of the triangle in spatial coordinates. Thus, if we construct 2×2 matrices $\mathbf{D}_m$ with columns $\mathbf{d}_{m_1}$ and $\mathbf{d}_{m_2}$, and $\mathbf{D}_s$ with columns $\mathbf{d}_{s_1}$ and $\mathbf{d}_{s_2}$, then $\mathbf{D}_s = \mathbf{F}\mathbf{D}_m$ or $\mathbf{F} = \mathbf{D}_s\mathbf{D}_m^{-1}$. Using this definition of $\mathbf{F}$ the Green strain is $\mathbf{G} = 1/2\left(\mathbf{D}_m^{-T}\mathbf{D}_s^T\mathbf{D}_s\mathbf{D}_m^{-1} - \mathbf{I}\right)$, which can be rewritten to obtain

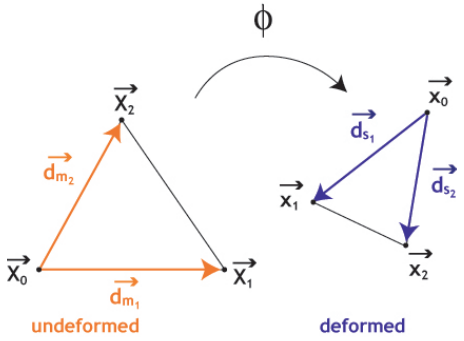

Figure 1: ***Undeformed and deformed triangle edges.***

$$\mathbf{D}_m^T\mathbf{G}\mathbf{D}_m = 1/2\left(\mathbf{D}_s^T\mathbf{D}_s - \mathbf{D}_m^T\mathbf{D}_m\right) =$$

$$\frac{1}{2}\left[\begin{pmatrix} \mathbf{d}_{s_1}\cdot\mathbf{d}_{s_1} & \mathbf{d}_{s_1}\cdot\mathbf{d}_{s_2} \\ \mathbf{d}_{s_1}\cdot\mathbf{d}_{s_2} & \mathbf{d}_{s_2}\cdot\mathbf{d}_{s_2} \end{pmatrix} - \begin{pmatrix} \mathbf{d}_{m_1}\cdot\mathbf{d}_{m_1} & \mathbf{d}_{m_1}\cdot\mathbf{d}_{m_2} \\ \mathbf{d}_{m_1}\cdot\mathbf{d}_{m_2} & \mathbf{d}_{m_2}\cdot\mathbf{d}_{m_2} \end{pmatrix}\right].$$

in order to emphasize that we are simply measuring the change in the dot products of each edge with itself and the other edge.

In three spatial dimensions, $\mathbf{D}_m$ and $\mathbf{D}_s$ are 3×3 matrices with columns equal to the edge vectors of the tetrahedra, and $\mathbf{D}_m^T\mathbf{G}\mathbf{D}_m$ is a measure of the difference between the dot products of each edge with itself and the other two edges. Note that $\mathbf{D}_m^{-1}$ can be be precomputed and stored for efficiency.

4. Finite Volume Method

FVM provides a simple and geometrically intuitive way of integrating the equations of motion, with an interpretation that rivals the simplicity of mass-spring systems. However, unlike masses and springs, an arbitrary constitutive model can be incorporated into FVM.

In the deformed configuration, consider dividing up the continuum into a number of discrete regions each surrounding a particular node. Figure 2 depicts two nodes each surrounded by a region. Suppose that we wish to determine the force on the node $\mathbf{x}_i$ surrounded by the region Ω. Ignoring body forces for brevity, the force can be calculated as

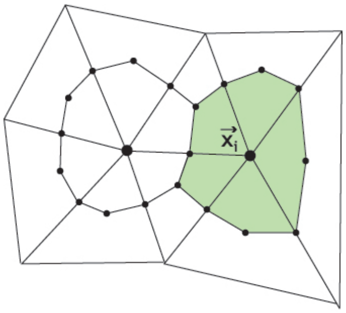

Figure 2: ***FVM integration regions.***

$$\mathbf{f}_i = \frac{D}{Dt}\int_\Omega \rho\mathbf{v}d\mathbf{x} = \oint_{\partial\Omega} \mathbf{t}dS = \oint_{\partial\Omega} \sigma\mathbf{n}dS$$

where ρ is the density, $\mathbf{v}$ is the velocity, and $\mathbf{t}$ is the surface traction on $\partial\Omega$. The last equality comes from the definition of the Cauchy stress $\sigma\mathbf{n} = \mathbf{t}$.

Evaluation of the boundary integral requires integrating over the two segments interior to each incident triangle. Figure 3 (left) depicts one of these incident triangles along with interior segments labeled $\partial\Omega_1$ and $\partial\Omega_2$. Since σ is constant in each triangle and the integral of the local unit normal over any closed region is identically zero (from the divergence theorem), we have

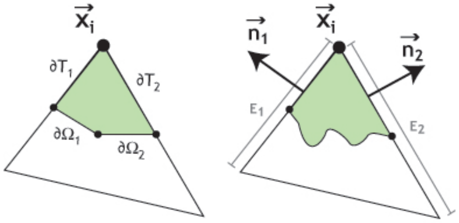

Figure 3: ***Integration over a triangle.***

$$\oint_{\partial\Omega_1} \sigma\mathbf{n}dS + \oint_{\partial\Omega_2} \sigma\mathbf{n}dS + \oint_{\partial T_1} \sigma\mathbf{n}dS + \oint_{\partial T_2} \sigma\mathbf{n}dS = 0$$

where ∂T_1 and ∂T_2 are depicted in the figure. More importantly, we have

$$\oint_{\partial\Omega_1} \sigma\mathbf{n}dS + \oint_{\partial\Omega_2} \sigma\mathbf{n}dS = -\oint_{\partial T_1} \sigma\mathbf{n}dS - \oint_{\partial T_2} \sigma\mathbf{n}dS$$

indicating that the integral of $\sigma\mathbf{n}$ over $\partial\Omega_1$ and $\partial\Omega_2$ can be replaced by the integral of $-\sigma\mathbf{n}$ over ∂T_1 and ∂T_2. That is, for each triangle, we can integrate over the portions of its edges incident to $\mathbf{x}_i$ instead of the two interior edges $\partial\Omega_1$ and $\partial\Omega_2$. Moreover, even if $\partial\Omega_1$ and $\partial\Omega_2$ are replaced by an arbitrary path inside the triangle, see figure 3 (right), we can replace the integral over this region with the integral over ∂T_1 and ∂T_2.

We choose an arbitrary path inside the triangles that connects the *midpoints* of the two edges incident on $\mathbf{x}_i$. Then the surface integrals are simply equal to $-\sigma\mathbf{n}_1 e_1/2$ and $-\sigma\mathbf{n}_2 e_2/2$ where e_1 and e_2 are the edge lengths of the triangles. Thus, the force on node $\mathbf{x}_i$ is updated via

$$\mathbf{f}_i += -\tfrac{1}{2}\sigma\left(e_1\mathbf{n}_1 + e_2\mathbf{n}_2\right).$$

In three spatial dimensions, given an *arbitrary* stress σ, regardless of the method in which it was obtained, we obtain the FVM force on the nodes in the following fashion. Loop through each tetrahedron interpreting $-\sigma$ as the outward pushing " multidimensional force". For each face, multiply by the outward unit normal to calculate the traction on that face. Then multiply by the area to find the force on that face, and simply redistribute one third of that force to each of the incident nodes. Thus, each tetrahedron will have three faces that contribute to the force on each of its nodes, e.g. the force on node $\mathbf{x}_i$ is updated via

$$\mathbf{f}_i += -\tfrac{1}{3}\sigma\left(a_1\mathbf{n}_1 + a_2\mathbf{n}_2 + a_3\mathbf{n}_3\right).$$

Note that the cross product of two edges is twice the area of a face times the normal, so we can simply add one sixth of $-\sigma$ times the cross product to each of the three nodes.

4.1. Piola-Kirchhoff Stress

Often, application of a constitutive model will result in a second Piola-Kirchoff stress, $\mathbf{S}$, which can be converted to a Cauchy stress via $\sigma = J^{-1}\mathbf{FSF}^T$ where $J = det(\mathbf{F})$. Using this equality and the identity $a\mathbf{n} = J\mathbf{F}^{-T}A\mathbf{N}$, we can write

$$\mathbf{f}_i += -\tfrac{1}{3}\mathbf{P}\left(A_1\mathbf{N}_1 + A_2\mathbf{N}_2 + A_3\mathbf{N}_3\right)$$

where $\mathbf{P} = \mathbf{FS}$ is the first Piola-Kirchhoff stress tensor, the A_i are the areas of the undeformed tetrahedron faces incident to $\mathbf{X}_i$ and the $\mathbf{N}_i$ are the normals to those undeformed faces.

Since the A_i and $\mathbf{N}_i$ do not change during the computation, we can precompute and store these quantities. Then the force contribution to each node can be computed as $\mathbf{g}_i = P\mathbf{b}_i$, where the $\mathbf{b}_i$ are precomputed and the force on each node is updated with $\mathbf{f}_i += \mathbf{g}_i$. Moreover, we can save 9 multiplications by computing $\mathbf{g}_0 = -(\mathbf{g}_1 + \mathbf{g}_2 + \mathbf{g}_3)$ instead of $\mathbf{g}_0 = P\mathbf{b}_0$. We can compactly express the computation of the other $\mathbf{g}_i$ as $\mathbf{G} = \mathbf{PB}_m$ where $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$ and $\mathbf{B}_m = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$. Thus, given an *arbitrary* stress $\mathbf{S}$ in a tetrahedron, the force contribution to all four nodes can be computed with two matrix multiplications and 6 additions for a total of 54 multiplications and 42 additions. A similar expression can be obtained for the Cauchy stress, $\mathbf{G} = \sigma\mathbf{B}_s$ where $\mathbf{B}_s$ is computed using deformed (instead of undeformed) quantities. Unfortunately, $\mathbf{B}_s$ cannot be precomputed since it depends on the deformed configuration.

4.2. Comparison with FEM

Using constant strain tetrahedra, linear basis functions N_i, etc., an Eulerian FEM derivation[2] leads to a force contribution of

$$\mathbf{g}_i = \int_{tet} \sigma \nabla N_i^T d\mathbf{v}.$$

A few straightforward calculations lead to

$$\mathbf{G} = \int_{tet} \sigma \mathbf{D}_s^{-T} d\mathbf{v} = \sigma \mathbf{D}_s^{-T} v = \sigma \hat{\mathbf{B}}_s$$

using our compact notation. Here, v is the volume of the deformed tetrahedron and $\hat{\mathbf{B}}_s = v\mathbf{D}_s^{-T}$.

Now consider $\mathbf{D}_s^T \mathbf{B}_s$ from the FVM formulation. Since the rows of $\mathbf{D}_s^T$ are edge vectors and the columns of $\mathbf{B}_s$ are each the sum of three cross-products of edges divided by 6, we obtain a number of terms that are triple products of edges divided by 6. Each of these terms is equal to either 0 or $\pm v$, and the final result is $\mathbf{D}_s^T \mathbf{B}_s = vI$. That is, $\mathbf{B}_s = v\mathbf{D}_s^{-T} = \hat{\mathbf{B}}_s$, and in this case of constant strain tetrahedra, linear basis functions, etc., FVM and FEM are identical methods. Although this equivalence is not true in general, this simple case is used by a number of authors[26, 23] and in this case FVM provides an intuitive geometric interpretation of FEM.

$\mathbf{D}_s^{-T}$ is the cofactor matrix of $\mathbf{D}_s^T$ divided by the determinant, and since $\mathbf{D}_s^T$ is a matrix of edge vectors, its determinant is a triple product equal to $6v$. That is, $\hat{\mathbf{B}}_s = v\mathbf{D}_s^{-T}$ computes the volume twice even though it cancels out resulting in a cofactor matrix times $1/6$. Thus, $\mathbf{B}_s$ can be computed with 27 multiplications and 21 additions, for a total of 54 multiplications and 45 additions to compute the force contributions using the Cauchy stress. Although, this is 3 more additions than the second Piola-Kirchhoff stress case, we do not need to store the 9 numbers in $\mathbf{B}_m$ at each node.

Muller et al.[23] point out that a typical FEM calculation such as in O'Brien and Hodgins[26] requires about 288 multiplications. Instead, they use QR-factorization, loop unrolling, and the precomputation and storage of 45 numbers per tetrahedron to reduce the amount of calculation to a level close to our 54 multiplications. However, in the second Piola-Kirchhoff stress case that they consider, we only need to store 9 numbers per tetrahedron (as opposed to 45). Moreover, in the Cauchy stress case that they do not consider, it is not clear that their optimizations could be applied without an expensive calculation to transform back to a second Piola-Kirchhoff stress. On the other hand, using the geometric intuition we gained from FVM that led to the cancellation of v (that other authors have not noted[26, 23]), we once again need only 54 multiplications and this time do not need to precompute and and store any extra information at all.

4.3. Time Stepping

When using a mixed explicit/implicit approach to time integration[3, 4] treating the elastic stress explicitly and the damping stress implicitly, one only needs the forces (i.e. no force Jacobians) for implementation because most damping models are linear (including Rayleigh damping[15]). This time stepping scheme is attractive as it does not suffer from the artificial numerical viscosity created by fully implicit time integration of both position and velocity, while still allowing one to disregard the strict time step restriction imposed by the damping forces. The time step restriction for explicit time integration of the damping forces is proportional to the minimum tetrahedron edge length squared, whereas explicit time integration of the position only (with implicit integration of the damping) results in a much less stringent restriction proportional to the minimum edge length to the first power[4].

4.4. Example

In order to illustrate the FVM technique, we simulate a bouncing torus using simple isotropic linear elasticity to calculate the stress. The results are shown in figure 4.

5. Constitutive Model for Muscle

Muscle tissue has a highly complex material behavior—it is a nonlinear, incompressible, anisotropic, hyperelastic material. This section details the constitutive model used to calculate the stress exerted by a volume element given a measure of the material deformation. We use a state-of-the-art constitutive model that includes a hyperelastic component, a quasi-incompressible component, and a transversely isotopic component.

Muscle is a hyperelastic material meaning that it is a soft elastic material that undergoes large deformations. To model a hyperelastic material, a scalar function $W(\mathbf{G})$ is defined to represent the strain energy at each point in the tissue as a function of the strain. The second Piola Kirchhoff stress in a hyperelastic material is related to the strain by $\mathbf{S} = \partial W / \partial \mathbf{G}$. The combined structure of connective tissue, water, and fibers in muscle can be modeled as a fiber-reinforced composite with a strain energy that has the form

$$W(I_1, I_2, \lambda, \mathbf{a}_o, \alpha) = F_1(I_1, I_2) + U(J) + F_2(\lambda, \alpha)$$

where I_1 and I_2 are deviatoric isotropic invariants of the strain, λ is a strain invariant associated with transverse isotropy (it equals the deviatoric stretch along the fiber direction), $\mathbf{a}_o$ is the fiber direction, and α represents the level of activation in the tissue. F_1 is the isotropic term, $U(J)$ is the term associated with incompressibility, and F_2 represents the active and passive muscle fiber response. This strain energy function is based on Weiss et al.[33], with some modifications to represent active and passive muscle properties.

A reasonable form for F_1 is a Mooney-Rivlin rubber-like model[21] because of muscle's soft, nonlinear behavior. We use $F_1(I_1, I_2) = AI_1 + BI_2$ where A and B are material constants, $I_1 = tr(\mathbf{C})$ and $I_2 = 1/2((tr(\mathbf{C}))^2 - tr((\mathbf{C})^2))$, where $\mathbf{C} = J^{-2/3}\mathbf{F}^T\mathbf{F}$.

The incompressibility term, $U(J)$, assumes the strain energy function takes an uncoupled form by separating the dilitational (volumetric) and deviatoric (non-volumetric) responses[29, 33]. $U(J) = Kln(J)^2$ represents the dilitational response where J is the relative volume ($= 1$ for strict incompressibility) and the degree of incompressibility can be controlled by the magnitude of the bulk "volumetric" modulus K.

The muscle fiber term F_2 must take into account the muscle fiber direction $\mathbf{a}_o$, the deviatoric stretch in the along-fiber direction λ, the nonlinear stress-stretch relationship in muscle, and the activation level. The tension produced in a fiber is directed along the vector tangent to the fiber direction (therefore, muscle is much stiffer in the fiber direction than in the plane perpendicular to the fiber direction). The relationship between the stress in the muscle and the fiber stretch has been established using single-fiber experiments and then normalized to represent any muscle fiber[35]. We define the first derivative of F_2 with respect to λ such that a non-linear stress-stretch relationship in the fiber direction follows these known behaviors for muscle fibers and can be scaled by the activation level (see figure 5). Passive muscle fibers resist tension and follow an exponential relationship, where the slack length occurs when the fiber stretch, λ, is equal to one. The stretch in the fiber direction is calculated by $\lambda = \sqrt{(\mathbf{a}_o\mathbf{C}\mathbf{a}_o)}$, and the muscle term F_2 is a function of the stretch and the activation level (see figure 5): $F_2(\lambda, \alpha) = \alpha F_{active}(\lambda) + F_{passive}(\lambda)$. A similar passive behavior is used to model the material response in the tendons by using stiffer material constants and a zero activation level.

Based on the final form for W, the stress in the tissue is:

$$\sigma = p\mathbf{I} + \tfrac{2}{J}\left(\left(W_1 + I_1 W_2\right)\mathbf{B} - W_2\mathbf{B}^2 + W_\lambda \lambda^2 \mathbf{a}\otimes\mathbf{a}\right)$$

$$-\tfrac{2}{3J}\left(W_1 I_1 + 2W_2 I_2 + W_\lambda \lambda^2\right)\mathbf{I}$$

where $\mathbf{B} = J^{-2/3}\mathbf{F}\mathbf{F}^T$, $W_i = \partial W/\partial I_i$, $\mathbf{a} = J^{-1/3}\mathbf{F}\mathbf{a}_o/\lambda$, and $p = K\partial U/\partial J$. The model describes the stress-strain relationship of the material for a given fiber direction and activation level.

6. B-spline Fiber Representation

Muscle tissue fiber arrangements vary in complexity from being relatively parallel and uniform to exhibiting several distinct regions of fiber directions. In Ng-Thow-Hing and Fiume[25], volumetric B-spline solid models were used to successfully capture detailed fiber architecture of actual muscle specimens. In our work, we use B-spline solids to solve the problem of assigning fiber directions to individual tetrahedrons of our muscle simulation meshes by querying the B-spline solid's local fiber direction at a spatial point corresponding to the centroid of a tetrahedron. As the fibers often vary in direction within a B-spline solid, this permits the modeling of a nonuniform distribution of fibers within a single muscle.

B-spline solids have a volumetric domain and a compact representation of control points, $\mathbf{q}_{ijk}$, weighted by B-spline basis functions $B^u(u), B^v(v), B^w(w)$:

$$\mathbf{F}(u,v,w) = \sum_i \sum_j \sum_k B_i^u(u) B_j^v(v) B_k^w(w)\mathbf{q}_{ijk}$$

where $\mathbf{F}$ is a volumetric vector function mapping the material coordinates (u,v,w) to their corresponding spatial coordinates. Taking the partial derivatives of $\mathbf{F}$ with respect to each of the three material coordinates $\partial\mathbf{F}/\partial u$, $\partial\mathbf{F}/\partial v$, $\partial\mathbf{F}/\partial w$ produces three directional vectors. In this manner, a B-spline solid has an implicit fiber field defined in its domain in each of its material coordinate directions.

In Ng-Thow-Hing and Fiume[25], one of these parameters always coincided with the local tangent of the muscle fiber located at the spatial position corresponding to the material coordinates. The inverse problem of finding the material coordinates for a given spatial point can be solved using numerical root-finding techniques to create a fiber query function

$$\mathbf{X}(\mathbf{x}) = \frac{\partial\mathbf{F}(\mathbf{F}^{-1}(\mathbf{x}))/\partial m}{\left\|\partial\mathbf{F}(\mathbf{F}^{-1}(\mathbf{x}))/\partial m\right\|}$$

with $m = \{u,v,w\}$ depending on the parameter chosen and the fiber directions normalized. The function $\mathbf{X}$ describes an operation that first inversely maps the spatial points back to their corresponding material coordinates (u,v,w) and then computes the normalized fiber direction at that point.

7. Specifying Activation Levels

We use a variation of an established biomechanics analysis technique known as *muscle force distribution*[7] to find the activations of redundant sets of muscles about each joint that they span. Muscle force distribution requires a static inverse dynamics analysis first be performed on the articulated skeleton to estimate the required joint torques to achieve a static pose. A simple line segment model of muscle is used to efficiently compute muscle moment arms[11], and a quasi-static approach is used to solve the inverse dynamics by taking each key pose of an animation and assuming it is in static equilibrium (velocities and accelerations are zero). This is a sufficient assumption for relatively slow-moving motions, but less adequate if limbs are undergoing high accelerations. A constrained least-squares optimization[1] is performed to minimize the sum of activations squared while enforcing equality constraints that balance the joint torques computed from the inverse dynamics analysis with the sum of muscle torques generated from the muscle forces about that joint. See figure 6.

Although accurate muscle force models can be used in this technique, in practice we used a simpler linear muscle force model: $F_m = a_m F_m^{max}$ where the maximum muscle force F_m^{max} for muscle m is scaled linearly by its activation a_m. This simpler model requires only a single parameter, F_m^{max}, be specified for each muscle corresponding intuitively to the inherent strength of the muscle, making it easier for an animator to find a suitable set of muscle force parameters to generate enough torque in the joints.

8. Simulating Skeletal Muscle

We built tetrahedral meshes (using Molino et al.[20]) of the biceps and triceps muscles from visible human data set[32]. This mesh was combined with our muscle constitutive model including the B-spline fiber directions and muscle activations. FVM proved to be both efficient and robust on a wide variety of both isometric and non-isometric simulations. We report on a sample of these below.

8.1. Isometric Contraction

Figure 7 shows the results of an isometric contraction of both the biceps brachii and the triceps brachii. The figure on the left is the relaxed state, while the figure on the right demonstrates the bulging of muscle bellies under contraction. The muscle bellies bulge due to stretch in the tendons which have a passive constitutive model. Multiple heads of both the triceps and the biceps muscles were included.

8.2. Contraction during Arm Movement

Elbow and shoulder angles were prescribed for a general arm motion, and the activation levels for the biceps and triceps were calculated (based on section 7). The muscle deformations during this motion were significant showing ballistic follow through motion of the muscle. This motion is exaggerated here because there are no surrounding tissues (fascia, other muscles, skin, etc.) to constrain the motion. See figure 8.

9. Conclusions and Future Work

Biomechanical simulations of human movement currently assume that muscles contract uniformly by representing muscles geometrically as a single line of action from origin to insertion (e.g. Delp and Loan[11]). This limiting assumption is largely a result of the computational intensity required by FEM techniques to model nonlinear three-dimensional soft materials. Previous three-dimensional FEM muscle models have been limited—they have been used to simulate a single muscle in isolation with highly simplified geometries (e.g. Johansson[17]). In light of this, the geometrically intuitive FVM formulation should provide valuable insight in the face of aggressive optimization, approximation and control currently being sought after in both the graphics and biomechanics communities. Already, FVM has motivated our reduction of the multiplications needed to put the force on the nodes from 288 (see Muller et al.[23]) to 54 per tetrahedron. Moreover, we obtain this reduction while only storing 9 numbers per tetrahedron (as opposed to 45 in Muller et al.[23]) in the second Piola-Kirchoff stress case, and without any storage requirements whatsoever in the Cauchy stress case.

10. Acknowledgements

Research supported in part by an ONR YIP award and a PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, a Sloan Research Fellowship, ONR N00014-03-1-0071, ONR N00014-02-1-0720, NSF ITR-0121288, NSF ACI-0205671, NIH HD38962 and by Honda Research Institute USA. In addition, J. T. and S. B. were each supported in part by a National Science Foundation Graduate Research Fellowship.

References

1. AEM Design. Fsqp. www.aemdesign.com, 2000.
2. J. Bonet and R. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, Cambridge, 1997.
3. R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21:594–603, 2002.
4. R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *ACM Symp. Comp. Anim.*, 2003.
5. E. Caramana and M. Shashkov. Elimination of artificial grid distortion and hourglass-type motions by means of lagrangian subzonal masses and pressures. *J. Comput. Phys.*, (142):521–561, 1998.
6. D. Chen and D. Zeltzer. Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. *Comput. Graph. (SIGGRAPH Proc.)*, pages 89–98, 1992.
7. R. Crowninshield. Use of optimization techniques to predict muscle forces. *Trans. of the ASME*, 100:88–92, may 1978.
8. G. Debunne, M. Desbrun, A. Barr, and M-P. Cani. Interactive multiresolution animation of deformable models. In *Proc. of the Eurographics Wrkshp on Comput. Anim. and Sim. 1999*. Springer Verlag, 1999.
9. G. Debunne, M. Desbrun, M. Cani, and A. Barr. Dynamic real-time deformations using space & time adaptive sampling. *Comput. Graph. (SIGGRAPH Proc.)*, 20, 2001.
10. G. Debunne, M. Desbrun, M.-P. Cani, and A. Barr.

Adaptive simulation of soft bodies in real-time. In *Comput. Anim. 2000, Philadelphia, USA*, pages 133–144, May 2000.

11. S. Delp and J. Loan. A graphics-based software system to develop and analyze models of musculoskeletal structures. *Comput. Biol. Med.*, 25(1):21–34, 1995.

12. J.-P. Gourret, N. Magnenat-Thalmann, and D. Thalmann. Simulation of object and human skin deformations in a grasping task. *Comput. Graph. (SIGGRAPH Proc.)*, pages 21–30, 1989.

13. E. Grinspun, P. Krysl, and P. Schröder. CHARMS: A simple framework for adaptive simulation. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21:281–290, 2002.

14. G. Hirota, S. Fisher, A. State, C. Lee, and H. Fuchs. An implicit finite element method for elastic solids in contact. In *Comput. Anim.*, 2001.

15. T. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Prentice Hall, 1987.

16. D. James and D. Pai. DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21, 2002.

17. T. Johansson, P. Meier, and R. Blickhan. A finite-element model for the mechanical analysis of skeletal muscles. *J. Theor. Biol.*, 206:131–149, 2000.

18. R. Lasseter. Principles of traditional animation applied to 3D computer animation. *Comput. Graph. (SIGGRAPH Proc.)*, pages 35–44, 1987.

19. J. Lewis, M. Cordner, and N. Fong. Pose space deformations: A unified approach to shape interpolation a nd skeleton-driven deformation. *Comput. Graph. (SIGGRAPH Proc.)*, pages 165–172, 2000.

20. N. Molino, R. Bridson, J. Teran, and R. Fedkiw. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra, (in review).

21. M. Mooney. A theory of large elastic deformation. *J. Appl. Phys.*, 11:582–592, 1940.

22. M. Muller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proc. of the ACM SIGGRAPH Symp. on Comput. Anim.*, pages 49–54. ACM Press, 2002.

23. M. Muller, L. McMillan, J. Dorsey, and R. Jagnow. Real-time simulation of deformation and fracture of stiff materials. In *Comput. Anim. and Sim. '01*, Proc. Eurographics Workshop, pages 99–111. Eurographics Assoc., 2001.

24. M. Neff and E. Fiume. Modeling tension and relaxation for computer animation. In *Proc. ACM SIGGRAPH Symp. on Comput. Anim.*, pages 77–80, 2002.

25. V. Ng-Thow-Hing and E. Fiume. Application-specific muscle representations. In W. Sturzlinger and M. McCool, editors, *Proc. of Gr. Inter. 2002*, pages 107–115. Canadian Information Processing Society, 2002.

26. J. O'Brien and J. Hodgins. Graphical modeling and animation of brittle fracture. *Comput. Graph. (SIGGRAPH Proc.)*, pages 137–146, 1999.

27. F. Scheepers, R. Parent, W. Carlson, and S. May. Anatomy-based modeling of the human musculature. *Comput. Graph. (SIGGRAPH Proc.)*, pages 163–172, 1997.

28. T. Sederberg and S. Parry. Free-form deformations of solid geometric models. *Comput. Graph. (SIGGRAPH Proc.)*, pages 151–160, 1986.

29. J. Simo and R. Taylor. Quasi-incompressible finite elasticity in principal stretches: continuum basis and numerical examples. *Comput. Meth. in Appl. Mech. and Eng.*, 51:273–310, 1991.

30. D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *Comput. Graph. (SIGGRAPH Proc.)*, pages 269–278, 1988.

31. D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Comput. Graph. (Proc. SIGGRAPH 87)*, 21(4):205–214, 1987.

32. U.S. National Library of Medicine. The visible human project, 1994. http://www.nlm.nih.gov/research/visible/.

33. J. Weiss, B. Maker, and S. Govindjee. Finite-element impementation of incompressible, transversely isotropic hyperelasticity. *Comput. Meth. in Appl. Mech. and Eng.*, 135:107–128, 1996.

34. J. Wilhelms and A. Van Gelder. Anatomically based modeling. *Comput. Graph. (SIGGRAPH Proc.)*, pages 173–180, 1997.

35. F. Zajac. Muscle and tendon: Properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomed. Eng.*, 17(4):359–411, 1989.

36. Q. Zhu, Y. Chen, and A. Kaufman. Real-time biomechanically-based muscle volume deformation using fem. *Comput. Graph. Forum*, 190(3):275–284, 1998.

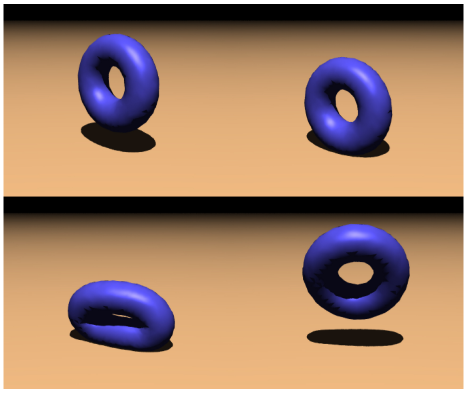

Figure 4: *Deformable torus simulated with FVM.*

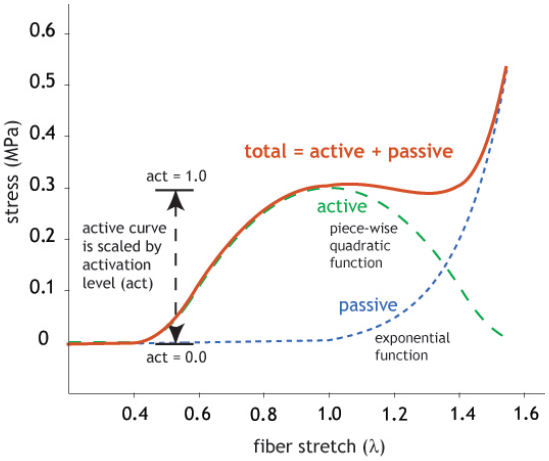

Figure 5: *Muscle fiber active and passive behavior.*

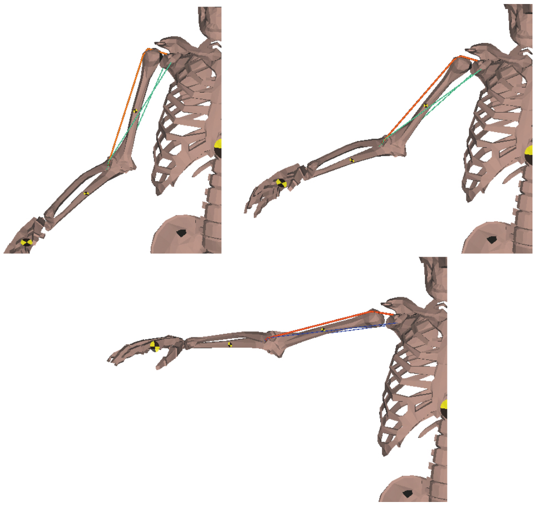

Figure 6: *Activations computed from three different poses. The line segment muscles are color-coded along the spectrum of little activation (blue) to full activation (red).*

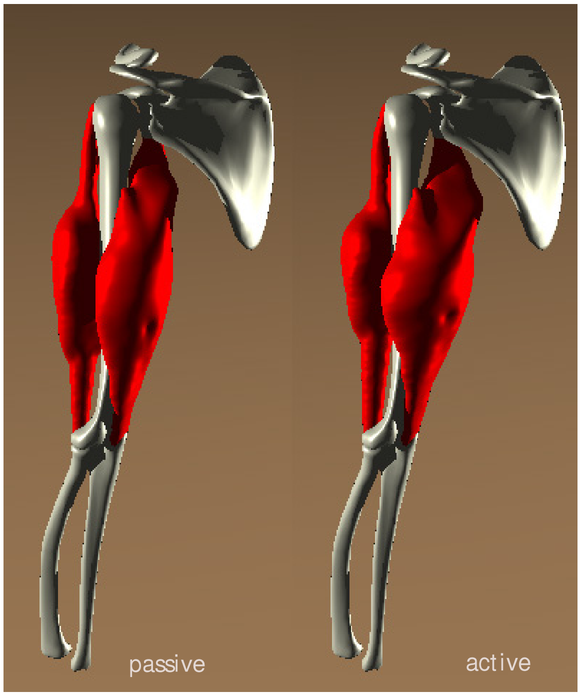

Figure 7: *Simulation of isometric contraction. A posterior (from behind) view of the upper arm shows contraction of the triceps muscle and the partially occluded biceps muscle from passive (left) to full activation (right).*

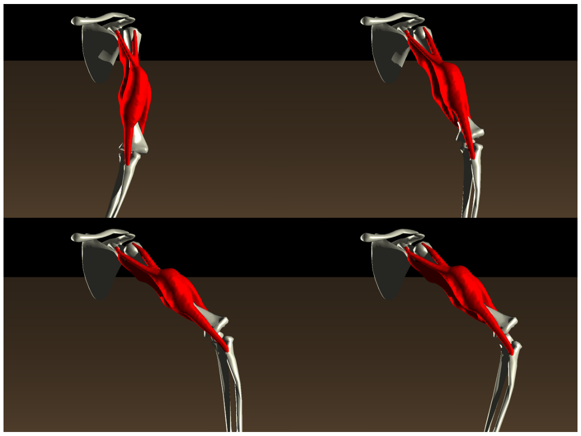

Figure 8: *Muscle contraction with skeletal motion. Inverse dynamics calculations were used with the motion sequence to compute muscle activations. These activations influence the amount of tension in the muscle during the animation and hence cause the muscle to deform in a realistic manner.*