Efficient Muscle Shape Deformation

Amaury Aubel and Daniel Thalmann
Computer Graohics Lab

Swiss Federal Institue of Technology (EPFL)

CH 1015 Lausanne, Switzerland

Tel: +41 21 693 52 14 Fax: +41 21 693 53 28

e-mail: {Amaury.Aubel, Daniel.Thalmann}@epfl.ch

Abstract: In this paper we extend previous work [Aubel00] and propose a muscle model suitable for computer graphics based on physiological and anatomical considerations. Muscle motion and deformation is automatically derived from an action line that is deformed using a 1D mass-spring system. The resulting model is fast and can accommodate most superficial human muscles.

1.   INTRODUCTION

The basic function of the skeletal muscles is to generate movement. Upon contraction, the fibres that make up the muscle contract and slide across each other. As a result, the length of the whole muscle diminishes, so the bones to which the muscle is attached are pulled towards each other. A side effect is

that the muscle changes shape during contraction, which impacts the shape of the outer skin. This is well-known among painters and sculptors who study the anatomy of the human body to improve their work. Rather surprisingly, commercial modelling packages overwhelmingly ignore muscle modelling as an essential part of body modelling. The most widespread technique for skin deformation in the industry remains *skinning* which amounts to binding each skin vertex to one or more underlying bones. The displacement of a skin vertex during animation is then the result of a weighted combination of the displacements of the bones to which it is bound. The influence of muscles on the skin surface shape is not taken into consideration or restricted to simple geometric primitives that push the skin outwards. In this paper we detail a muscle model that is fast and yet realistic enough for a computer graphics use.

1.1   Related work

Existing muscle models can broadly be classified into two categories: purely geometric models and physically-based ones. We successively review these two approaches.

1.1.1   Geometric Deformations

Geometric models tend to use the ellipsoid as the basic building block. It is a natural choice because an ellipsoid approximates fairly well the appearance of a fusiform muscle. In addition, its analytic formulation lends itself well to inside/outside tests and volume preservation constraints. Thus, several researchers use a volume-preserving ellipsoid for representing a fusiform muscle [Scheepers97, Wilhelms97a]. Others approximate muscles by an implicit surface extracted from a set of ellipsoids [Turner93,Thalmann96]. Finally, multi-belly muscles e.g. the pectoral muscle can be represented by a set of ellipsoids positioned along two spline curves [Scheepers97]. In all these works, muscle flexing and bulging is simulated by binding the degrees of freedom (scaling and possibly translation and/or rotation) of each ellipsoid to the degrees of freedom of the underlying skeleton joints.

Despite its simplicity and attractiveness, the ellipsoid model cannot capture most muscle shapes. In more recent work [Wilhelms97b], Whilhelms et al. use a generalised cylinder made up of a certain number of cross-sections that consist in turn of a fixed number of vertices. Volume variation of the muscle during deformation is reduced by scaling each cross-section so as to preserve its area. Similarly, Scheepers and his colleagues

provide a general muscle model that consists of tubularly-shaped bicubic patches [Scheepers97]. Exact volume preservation remains possible as muscles shapes still have an analytic description. Interestingly, they also provide the user with scaling and tension parameters to simulate isometric contractions as well.

1.1.2   Simulation models

One of the first physically-based models is due to Chadwick et al. [Chadwick89]. The muscle is embedded in a FFD lattice [Sedeberg86]. Muscle deformation is achieved by simply deforming the embedding space. The FFD control points are moved by treating them as nodes interconnected by ideal hookean springs. Diagonal springs help to maintain the initial geometric configuration. One of the potential problems is that the FFD box may not approximate the muscle shape very tightly. The FFD control points have moreover no physical reality. As a consequence, the distribution of the muscle mass over the nodes is likely to be problematic. From a more bio-mechanics oriented point of view, Chen et al. simulate muscle contraction using the Finite Element Theory [Chen92]. However, their work only shows single muscles working in isolation.

Porcher-Nedel and Thalmann introduced the idea of abstracting muscles by an action line (a polyline in practice) representing the force produced by the muscle on the bones, and a surface mesh deformed by an equivalent mass-spring network [Porcher-Nedel98]. An elastic relaxation of the surface mesh is performed for each animation frame thus yielding a collection of static postures. In order to smooth out mesh discontinuities, they employ special springs termed *angular springs* that tend to restore the initial curvature of the surface at each vertex. If the mesh is somewhat coarse, angular springs also help to control volume deformation though not in an exact mathematical manner. However, angular springs cannot deal with local inversions of the curvature. Also, the authors do not explicit how they constrain the surface mesh to follow the action line when it consists of more than one segment.

Ng-Thow-Hing relies on the B-spline solid as the basic primitive for modeling individual muscles in animals and humans [Ng-Thow-Hing00]. The mathematical formulation of the B-spline solid can accommodate multiple shapes of muscles (fusiform, triangular, bipennate, etc.) and various sizes of attachments. Muscular deformation is achieved by embedding a mass-spring-damper network in the B-spline solid. In practice, the network does not coincide with the B-spline's control points but with spatial points of maximum influence since physical characteristics such as mass are best

specified at real locations of the muscle. Varying force magnitude in the network results in non-uniform physical effects. In contrast to most previous approaches that solve a sequence of static equilibrium problems only, inertially-induced oscillations can take place here thus enhancing the visual realism. Muscle-muscle and muscle-bone collision forces are also added as reaction constraints [Platt88]. Yet, trying to simulate every muscle-muscle and muscle-bone interaction seems unrealistic. For example, no solution is given as to how multiple collisions[1] between muscles are to be handled.

1.2   Overview

The remainder of this paper is organised as follows. Section 2 exposes some important considerations borrowed from the artistic anatomy, as well as physiological notions that guided us in developing a generic muscle model. In the following section we detail the muscle model. Lastly, section 4 presents our conclusions and possible future work.

2.   ARTISTIC ANATOMY

The muscle layer is the main contributing factor to the surface form. Muscles account for half of the total mass of the body and fill in almost completely the gap between the skeleton and the skin [Richer81]. Anatomists distinguish three types of muscles: skeletal muscles, smooth muscles and the heart. They have different functions but exhibit the same fundamental mechanical and constitutive properties [Maurel98]. We shall only consider the skeletal muscles because the other kinds barely influence the surface form.

Skeletal muscles produce the motion of the bones. Structurally, they consist of a contractile central part called *belly* and of tendinous extremities or *aponeurosis* that connect the belly to the bones. In constitutive description, the belly is made up of bundles of elastic contractile fibres. The bundles are wrapped into a single envelope called *fascia*. The belly's fibres are responsible for producing the contraction of the whole muscle. Tendons, which are hardly elastic, act as transmitters and help to move the weight away from the limbs' ends. In general, muscle tissues are several orders of magnitude more elastic than tendons [Fung81].

Upon *isotonic contraction*, the volume of the belly increases thus amplifying its influence on the shape of the skin, whereas the total length of

---

[1] A multiple collision occurs for example when a muscle A collides with a muscle B, which in turn collides with a muscle C within the same time step because of the displacement due to the first collision.

the muscle diminishes so that the bones to which the muscle is attached are pulled towards each other. Upon *isometric contraction*, the shape of the belly alters but the length of the muscle does not change, so no skeletal motion is produced. In a relaxed state, the belly undergoes the action of gravity and hangs somewhat loosely. Finally, muscles vary greatly in shape depending on their location: long fusiform muscles are found mainly in the limbs; short muscles appear around joints; large and flat muscles cover the back [Richer81].

3.   MUSCLE MODEL

We believe the real difficulty with muscles lies more with the animation than with the modelling. It is very complex to automatically derive the appropriate position and deformation of a muscle in any possible posture. Note that in our approach, as almost always the case in computer graphics, the motion of the skeleton induces the muscular deformations contrary to what occurs in reality.

Our approach consists of decomposing the muscle into two layers: an action line and a surface mesh. The action line, represented by a polyline with any number of vertices, is moved for each posture using a predefined behaviour and a simple physically-based simulation. It is then used as a skeleton for the surface mesh and the deformations are produced in a usual way [Sun99]. In order to avoid possible confusion, we shall use in the following the term "node" when referring to a vertex of the action line and "vertex" when speaking of a vertex of the muscle surface mesh.

3.1   Action line

First of all, the user specifies a default behaviour for each node of the action line: the node is mapped to a specific bone and its motion is defined with respect to a given number of joints. Then, a 1D mass-spring-damper system is constructed from the polyline. It is used for automatically determining new positions of the nodes. Currently, all nodes are given an equal mass. The user may choose at any time to deactivate the dynamic behaviour of a node, in which case the predefined behaviour takes over. An elastic relaxation is performed for each posture. The physical simulation can be advanced rapidly by relying on an implicit integration scheme since it yields an easily invertible tridiagonal[2] matrix [Kass93]. We add attractive and repulsive implicit force fields (currently ellipsoids and ellipsoidal

---

[2] A nine-diagonal matrix in fact as each force vector has three entries.

metabolls) to constrain the action line. Repulsive force fields prevent gross interpenetration while attractive fields help to refine the trajectories of the action line.
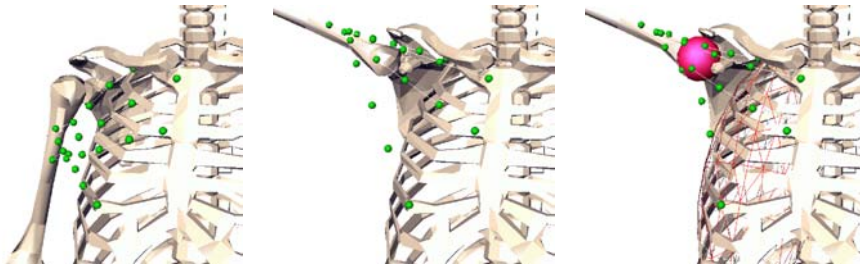

Figure 1. Action lines of the pectoral muscle during shoulder abduction. Right picture shows the use of two attractive force fields (solid and wireframe ellipsoids)

In practice, non-dynamic vertices correspond to the insertion and origin of the tendons. The action line can wrap itself around joints providing that the number and location of vertices is well chosen. Nearly rigid portions of the muscle such as tendons can easily be simulated because the stiffness of each spring is under user's control. The implicit integration easily handles these stiff segments. Analogously, increasing the number of vertices and fine-tuning the stiffness of the created springs can roughly approximate non-linear elasticity.

3.2   Local frames

The positions of the action line nodes provide information as to how the surface mesh will expand or shrink over time. Yet, the orientation of the mesh cannot be inferred from these positions only. A local frame need to be constructed for each node of the action line. This is an involved operation. We start by computing the Z axis at each node as depicted in Figure 2: Z is set to the normal of the bisecting plane for every in-between node (V_1, V_2) and to the tangent for the end nodes (V_0, V_3). We then proceed to compute the X axis. Note that the Y axis are ultimately found by completing the right-handed co-ordinate systems.
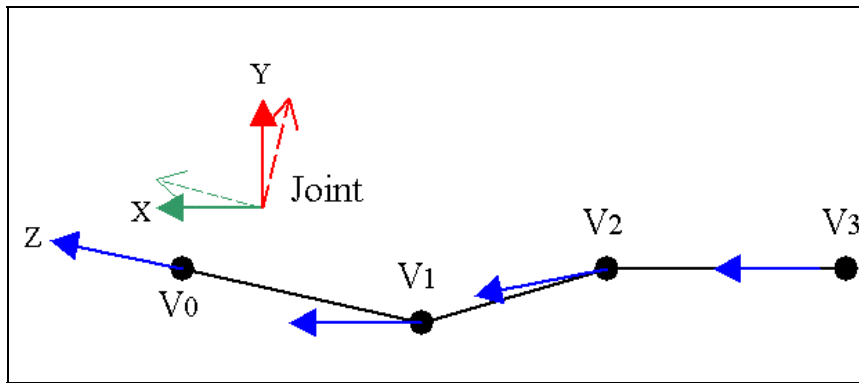
Figure 2. Z axis is set using the bisecting plane. The joint frame is initially rotated so as to align one of its axis (here the X axis) with axis Z of end node V0.

The X axis is first computed for the non-dynamic nodes[3]. We take, in a rest posture, the local frame of the joint (X and Y solid arrows in Figure 2) to which the non-dynamic node is bound and rotate it so as to bring its X, Y, or Z axis in alignment with the node's Z axis. The selected axis is the one that leads to the minimal rotation. In Figure 2 for instance, X yields the smallest rotation. The resulting frame (dashed arrows in Figure 2) is then expressed and saved in the joint's co-ordinate system. During any subsequent animation, this frame is transformed by the joint's current co-ordinate system, then rotated again so as to be aligned with the node's new Z axis. This rotation is usually quite small because we initially chose the smallest rotation in the rest posture. Thus, the local frame of every non-dynamic node is smoothly updated as the action line moves and deforms itself.

---

[3] Non-dynamic nodes have their motion driven by a predefined behaviour (cf. Section 3.1). They usually (but not necessarily) correspond to the end nodes (V0 and V3 in fig. 2).
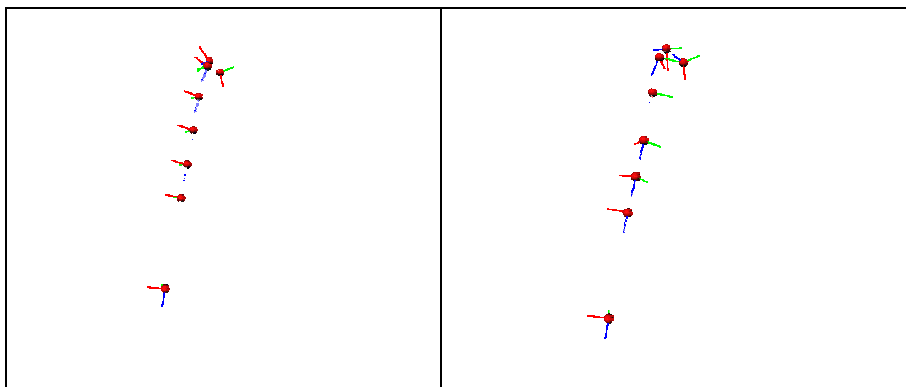
Figure 3. Left: propagating first node's orientation upwards; Right: Resulting frame interpolation for in-between nodes.

There remains to compute the X axis for the dynamic nodes in between. It is no use interpolating the two end orientations directly using the commonplace spherical linear interpolation because the two end orientations may be quite different from each other. As spherical linear interpolation picks the shortest path on the quaternion unit sphere, the frame orientation at a node may flip from one animation frame to the next. Besides, direct interpolation does not guarantee the interpolated frames will be aligned with the Z axis already computed. Our method consists in propagating the X axis direction of each end frame to the in-between nodes. As Z axis are readily computed using the method described above, we already have, for each node V_i, a plane P_i normal to Z_i in which the remaining X_i and Y_i axis must lie. Starting from axis X_0 at end node V_0, we estimate the axis X_1 in the plane P_1 by sampling the trigonometric circle and finding the minimal deviation from X_0. As we sample the entire circle, we do not get stuck into local minima. We then iterate the process to compute X_{i+1} by minimising its deviation from X_i. We thus propagate the orientation of each end frame to the other end node. Note that Figure 3. only show the upwards propagation but the inverse downwards propagation is also carried out. Finally, we perform a linear interpolation of the two X axis computed at each node using a ratio that is related to the distance from the in-between node to the two end nodes along the polyline (Figure 3 right).

3.3   Muscle Mesh

We automatically map each surface vertex to the two closest delimiting planes that pass trough an action line's node as in [Sun99]. Vertices positions are later found by linear interpolation of the position and orientation of the enclosing local frames. Isotonic contraction is simulated by

scaling each surface vertex orthogonally to the action line. The scaling factor is individually computed based on the action line. We compute the elongation – defined as the current length divided by the initial length – for every segment of the action line (it is computed anyway when evaluating the spring's elastic force). We interpolate these discreet measurements with a cubic spline curve. Thus we obtain a smooth, individual elongation value for each muscle vertex that we use as the scaling factor squared root: $scaling = \sqrt{elongation}$. Though this empirical formula does not ensure volume preservation, we experimentally measured for various muscle shapes a maximal volume variation of 6% when the muscles shorten by 30%, which corresponds to the maximal physiological compression rate [Richer81].
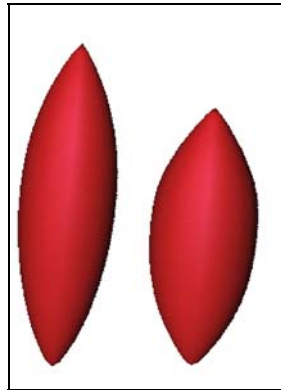

Figure 4. Compression of a fusiform muscle with near-exact volume preservation

4. CONCLUSION

We presented a two-layered muscle model suitable for computer graphics applications. An action line is used for driving the motion and deformation of the outer layer. As the model makes use of a 1D mass-spring system, muscle deformation can be performed in real-time. Figure 5 shows the deformation of the brachial muscle as an example.
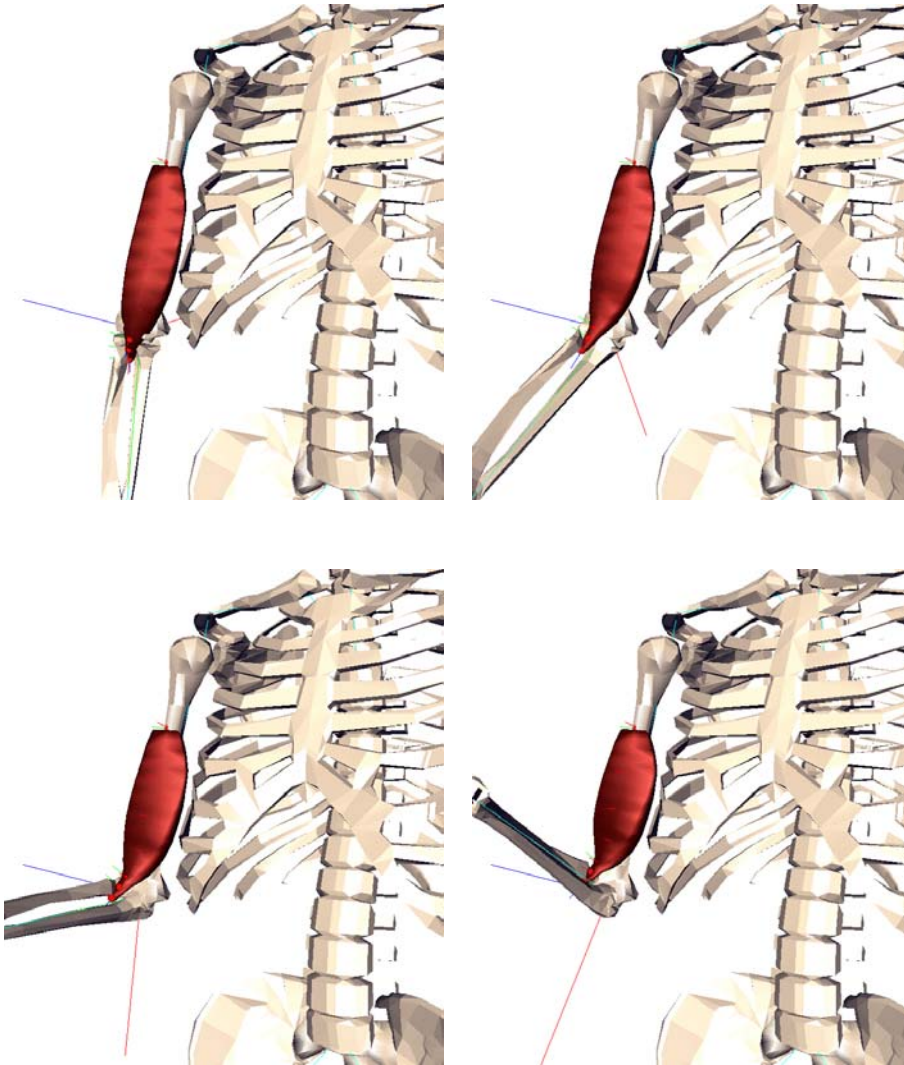
Figure 5. deformation of the *brachialis* muscle (volume variation remains under 3%)

We plan to extend our 1D muscle model to large flat muscles as those in the back for example. Our idea is to use a 2D "action grid", essentially a surface, with nodes interconnected by ideal hookean springs. Then, a surface mesh would be wrapped around the action grid using the same method as that described in this paper.

We are also considering covering the muscles with a deformable skin mesh. Our idea is to deform the skin mesh using a two-stage process. A

skinning process would be used for roughly positioning each skin vertex. We plan to decouple the degrees of freedom of a 3-DOF joint (e.g. the shoulder) into a swing motion on the one hand and a twist motion on the other. This should prove useful because the skin does not completely follow the skeleton when twisting one's limbs as pointed out by Molet et al. [Molet96]. In the second stage the skin vertices would be pushed and attracted by the underlying muscles.

ACKNOWLEDGMENTS

REFERENCES

[Aubel00] A. Aubel, D. Thalmann, "Realistic Deformation of Human Body Shapes", Proc. Computer Animation and Simulation 2000, Interlaken, August 2000, pp. 125-135.

[Chadwick89] J. Chadwick, D. Haumann, R. Parent, "Layered construction for deformable animated characters", Computer Graphics (SIGGRAPH '89 Proceedings), pp.243-252.

[Chen92] D. Chen, D. Zeltzer, "Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method", Computer Graphics (SIGGRAPH '92 Proceedings), pp.89-98.

[Fung81] Y.C. Fung "Biomechanics: Mechanical Properties of Living Tissues", Springer-Verlag, 1981.

[Kass93] M. Kass, "Introduction to Continuum Dynamics for Computer Graphics", SIGGRAPH Course Notes 60, 1993.

[Maurel98] W. Maurel, Y. Wu, N. Magnenat Thalmann, D. Thalmann, "Biomechanical Models for Soft Tissue Simulation", Springer-Verlag, Berlin/Heidelberg 1998.

[Molet96] T. Molet, R. Boulic, D.Thalmann, "A Real-Time Anatomical Converter for Human Motion Capture", Proc. 7h Eurographics Workshop on Animation and Simulation, Springer-Verlag, Wien, September 1996.

[Ng-Thow-Hing00] V. Ng-Thow-Hing, "Anatomically-Based Models for Physical and Geometric Reconstruction of Humans and Other Animals", Ph.D. Thesis, Department of Computer Science, University of Toronto, 2000.

[Platt88] J.Platt, A. Barr, "Constraint Methods for Flexible Models", Computer Graphics (SIGGRAPH '88 Proceedings), pp. 279-288.

[Porcher-Nedel98] L. Porcher-Nedel, D.Thalmann, "Real Time Muscle Deformations Using Mass-Spring Systems", Proc. CGI '98, IEEE Computer Society Press, 1998.

[Richer81] P. Richer, "Artistic Anatomy", Watson-Gutpill Publications, New York, 1981, Translated by Robert Beverly Hale.

[Scheepers97] F. Scheepers, R. Parent, W. Carlson, S. May, "Anatomy-Based Modeling of the Human Musculature", Computer Graphics (SIGGRAPH '97 Proceedings), pp. 163-172.

[Sederberg86] T. Sederberg, S. Parry, "Free-From Deformation of Solid Geometric Models", Computer Graphics (SIGGRAPH '86 Proceedings), pp.151-160.

[Sun99] W. Sun, A. Hilton, R. Smith, J. Illingworth, "Layered Animation of Captured Data", Animation and Simulation '99 (10th Eurographics Workshop Proceedings), Milano, 1999, pp.145-154.

[Thalmann96] D. Thalmann, J.Shen, E. Chauvineau, "Fast Realistic Human Body Deformations for Animation and VR Applications", Computer Graphics International'96, Pohang, Korea, June, 1996.

[Turner93] R. Turner, D. Thalmann, "The Elastic Surface Layer Model for Animated Character Construction", Proc. Computer Graphics International '93, Lausanne, Switzerland, Springer-Verlag, Tokyo, pp. 399-412.

[Wilhelms97a] J. Wilhelms, "Animals with Anatomy", IEEE Computer Graphics And Applications, Vol. 17, No. 3, May 1997, pp.22-30.

[Wilhelms97b] J. Wilhelms, A. Van Gelder, "Anatomically Based Modeling", Computer Graphics (SIGGRAPH '97 Proceedings), pp. 173-180.