

Real Time Muscle Deformations Using Mass-Spring Systems

Luciana Porcher Nedel

Daniel Thalmann

EPFL - Swiss Federal Institute of Technology
LIG - Computer Graphics Lab
CH 1015 - Lausanne - VD - Switzerland
(nedel, thalmann)@lig.di.epfl.ch

Abstract

In this paper we propose a method to simulate muscle deformation in real-time, still aiming at satisfying visual results; that is, we are not attempting perfect simulation, but building a useful tool for interactive applications. Muscles are represented at 2 levels: the action lines and the muscle shape. The action line represents the force produced by a muscle on the bones, while the muscle shapes used in the simulation consist of a surface based model fitted to the boundary of medical image data. The algorithm to model muscle shapes is described. To physically simulate deformations, we used a mass-spring system with a new kind of springs called "angular springs" which were developed to control the muscle volume during simulation. Results are presented as examples at the end of the paper.

1. Introduction

It's inherent to the human being trying to simulate real scenes as best as possible. History already reveals a great amount of interest on this subject. Earlier, traditional techniques like clay sculpturing, hand drawing and painting were used for this purposes. It is exactly the case for the representation of human appearance. With the advent of technology, still photographs and movies came, which are more like copying mechanisms limited to two dimensional reproduction. More recently, with computers, we can perform the tasks of synthesis and animation with greater creativity and precision both in two and three dimensions.

For years, modeling and animation of the human figure has been an important research goal in computer graphics. At first, humans have been represented as simple articulated bodies made of segments and joints with a kinematics based model to simulate them. More recently,

dynamic models have been used to improve the continuity and fluidity of the movement.

However, simple skeleton-based models are limited for human picture simulation. In fact, a human body is a collection of complex rigid and non-rigid components very difficult to model. Considering the complexity of the human structure and the fact that our eyes are specially sensitive to familiar things (e.g., our own image), people tend to be very exacting about human simulation. Consequently, researchers began to use human anatomy know-how to produce human models with more realistic behavior.

On the other hand, computer graphics now has a strong trend to develop real-time applications and there is a need for virtual human actors in many applications (e.g., the simulation of virtual environments, teleconference, multimedia, and interaction between real and virtual actors). Initially, the research on a realistic human model seems to be incompatible with real-time goals. In fact, it is still quite impossible today to have an application with a very accurate human simulation and high quality images, running in real time.

Our goal is to achieve a realistic simulation of human movements using an anatomical approach and executing in an acceptable time for interactive applications. The work described here was developed within the European ESPRIT project CHARM and presents a model to simulate muscle deformation in real-time, producing visually acceptable results. We have chosen a physically-based model because we believe we can obtain more fluid and realistic movements using the same principles used in nature to accomplish it.

Concerning the CHARM project (A Comprehensive Human Animation Resource Model), its goal was to develop a 3D human body model (more specifically the upper limb), allowing the physically-based movement and finite elements deformation. The model and simulation are validated with medical data. To simulate muscle motion, a biomechanical model that uses a finite elements

implementation technique [16] was developed. It produces a realistic result, but with a prohibitive computational cost to integrate an interactive tool. This limitation has motivated the development of a real-time model also based on physical principles, but not as accurate as the first one.

In this paper we propose an alternative method to simulate muscle motion in real-time. We are not attempting perfect simulation, but building a useful tool for interactive applications. In section 2, we review the main models of deformation which have been proposed in the past few years, some applications in muscle simulation and some fundamental concepts of muscle physiology that are very important to the development of this work.

In section 3 we present the work corresponding to muscle shapes production and in section 4 the physically based model developed. In section 5 we make some considerations about motion simulation. We finish by presenting our results, some conclusions and future work.

2. Background

2.1. Deformable models

Methods to represent non-rigid objects are an important research area in computer graphics. Along the years, a lot of different methods have already been proposed to simulate this kind of objects. Here, we divide this models in 3 categories: particle systems, geometric deformation methods and physically-based methods.

In [26], Reeves proposed for the first time in Computer Graphics a particle system to represent natural phenomena. His particles are sets of moving points under the influence of external forces and without interaction between them. More recently, some other works propose the use of oriented particles to simulate elastic surfaces [30] and inelastic deformations [34]. They use an attraction-repulsion force to model interactions between particles. Adding more particles to the system, we can simulate deformation and stretching.

Concerning geometric deformation models, Barr has introduced the deformation of solid objects by the manipulation of operators over abstract data [2]. Sederberg and Parry [27] introduced the concept of Free-Form Deformation (FFD) that provides global and local deformation. Afterwards, this technique was extended to animation. The AFFD (Animated Free-Form Deformation) was developed by Coquillart and Jancène [10]. More recently, Mocozet and Magnenat-Thalmann [19] proposed the Dirichlet Free-Form Deformation method (DFFD) to simulate hands deformation.

Implicit surfaces are another geometric technique to represent deformation. Implicit surfaces are typically

defined by starting with simple building block functions (called primitive) and by creating new implicit functions using the sum, min, or max of simpler functions. The final object is constructed by blending the primitives, and as primitives are moved and deformed the resulting blended surface changes shape. Some types of implicit surfaces like metaballs [21], blobs [4], convolution surfaces [5] or soft objects [39] have received increasing attention in Computer Graphics. Nowadays, an important trend in representation of deformable objects is the use of implicit surfaces mixed with particle systems, as we can see in [11].

Physically-based models are well suited to provide natural motion for flexible objects. One of the first physically-based models was developed by Weil and used to simulate cloth by interpolating the surfaces between catenary curves [36]. Barr, Terzopoulos, Platt and Fleisher have used discrete molecular components to model the elastic behavior of objects [22]. Terzopoulos and Fleischer have extended the model mixing flexible and rigid components [33] and to simulate also inelastic behavior [32].

Many other techniques have been proposed to improve physically-based deformations, as the methods based on constraints presented by Platt and Barr [23] and Metaxas and Terzopoulos [17]. Approaches based on discretising the object into a number of particles, whose connectivity is maintained through attached springs are named mass-spring systems. Miller used it to animate snakes and worms [18] and currently, a lot of variations of this kind of system is being presented by other authors.

2.2. Muscle simulation

Concerning the simulation of muscle deformation, we can identify three different approaches: biomechanical models, physically-based models and geometric models.

One of the most important works in the area of muscle simulation is the model presented by Chen and Zeltzer in [8]. To simulate muscle forces and visualize the dynamics of muscle contraction, they use a biomechanical model of muscle action and apply non-linear forces to a linear finite element mesh. The problem with this approach is that it is too expensive to be considered for interactive applications.

The use of simpler physically-based models has also been considered to simulate muscle deformation. Holton and Alexander have presented in [13] the creation of a soft cellular structure composed by a volumetric arrangement of mass-points and connectors that forms an approximate tetrahedral tessellation. The model was created to simulate realistic visco-elastic deformation of a large array of tetrahedral node-connector systems in real-time applications such as virtual surgery simulation. The

model is essentially linear and the priority in this case is speed.

In [6], Bro-Nielsen and Cotin also present a physically-based model to improve surgery simulation. They use a linear elastic deformation model applied over a 3D volumetric solid represented in a finite elements mesh. In this case, speed is also the most important goal and the elastic model only needs to be visually convincing.

Promayon et al. have presented a method of constraining physically-based deformable objects [25]. An object is described only by its contour that is composed by a set of mass points. The object is locally specified in terms of kinetic and dynamic (mass, position, speed) and physically (compressibility, elasticity, motor functioning). The goal is to simulate respiratory movements realistically in order to create an educational and medical tool. This method improves the control of objects volumes during evolution. For motion, it uses Euler integration method.

About the use of geometric methods, we can mention the use of implicit surfaces to simulate muscles, as the work of Wilhelms, who has presented in [37] a new approach to animal modeling and animation using simulated individual bones and muscles, soft tissues and skin. In the work mentioned above, muscles are made of combinations of three ellipsoids (two tendons and one muscle between). More recently, this work was extended and now, in her anatomically based model, muscles are represented by deformable discretized cylinders [38].

Turner has developed the LEMAN system [35] to construct and animate 3D characters based on the elastic surface layer model. Muscles are modeled as deformable implicit surfaces (currently spheres, cylinders and super-ellipsoids). The muscle layer has been represented by deformable geometrical solid surfaces which the skin may not penetrate. The deformation of muscles is simulated by animating parameters of the global deformations which can be tied to the joint angle values. To perform the movement, a numerical algorithm is used to solve the partial differential equation of motion for the elastic surface, the known simple Gauss-Seidel or relaxation technique.

Shen et al. have developed in the Body Builder [29], a system for interactive design of human bodies. To represent bones, muscles and fat tissues, they employ ellipsoids and ellipsoidal metaballs. This approach is based on contours deformation.

More recently, Scheepers et al. present in [28] another anatomy-based model of muscles. They consider the influence of the musculature on surface form and develop muscle models which react automatically to changes in the posture of the articulated skeleton. Muscle bellies are represented by ellipsoids and deformations are provided by scaling the three major axes of the ellipsoids, simulating compression and extension motion.

2.3. Muscle physiology

Muscles are a kind of soft tissues that possess contractile properties. The skeletal muscles are connected to the bones through tendons, which are composed of bundles of fibers. But there are also other kinds of muscles that, in fact, can be distinguished depending on their functionalities: skeletal muscles, smooth muscles, and the heart muscles. While skeletal muscles have their origins and insertions attached to bones and are responsible for the motion of the skeletal structure, smooth muscles act on organs and other soft tissues in order to accomplish physiological tasks. The heart is a particular kind of muscle that can be normally viewed as an organ itself with specific functions, that is, to pump blood and to irrigate the organism. However, all three kinds of muscles have, in spite of their different functions, approximately the same structural characteristics and mechanical properties [12].

Considering the action aspect of a muscle, it is important to remark that it is composed by the passive soft tissues which connect it to the bones and by muscular contractile fibers, arranged in a parallel fashion between both tendons of origin and insertion. Regarding muscle contraction, when a muscle fiber shortens, the thin and thick myofilaments do not themselves get shorter, they rather slide across each other [14].

3. Muscle shape

Most muscles are attached to bones at both ends. The attachment closer to the spine of the body is generally considered the *origin* while the other one is the *insertion*. Normally, the origin is the fixed point, and the insertion is where the muscle performs its action, but there is not an established law about it. Most muscles are simply attached to two bones across a joint, and each bone may function as either the fixed point or the moveable one [9].

To mechanically quantify the force that a muscle produces over a bone, we can represent muscles by lines, named *action lines*, with the same origin and insertion than a muscle. In fact, depending of its shape and complexity, a muscle can be represented by one or more action lines where each line can also be represented by a polyline.

In our approach, we represent a muscle on 2 levels: the action line and the muscle shape. The first step of this work is to choose the action lines that best represent each muscle. Muscles are composed by one or more action lines and each one is defined by an origin point, an insertion point and optionally by one or more guide points. The origin and insertion points provide the connection of the muscle with the bones, while the guide points are used where the muscle action line could not be represented by only one segment (e.g. the Biceps

Brachii). As shown in Figure 1, this muscle is represented by two action lines and one of them needs to use a guide point. If we try to represent this action line by a straight line, we will intercept the bone. To avoid the collision, we design the action line as a polyline which the middle points are named guides.

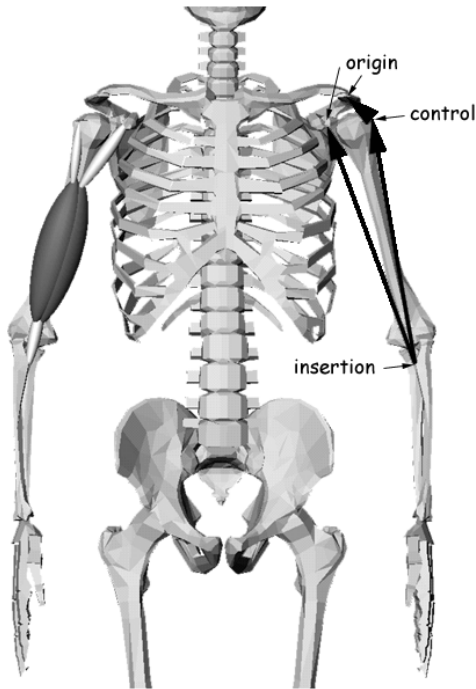


Figure 1. The Biceps Brachii muscle and corresponding action line

However, we cannot forget that over each group of one or more action lines we have a muscle shape, and all the work could be much simpler if all the muscles were ideal fusiform muscles, as shown in Figure 2. Our approach to simulating muscle deformations is based on the use of mass points connected by springs, with mass points organized in a way to assure a correspondence between the action line and the muscle shape. That is, each point in the model is placed between two neighbors in the horizontal and two others in the vertical sense, considering the action line of a muscle as the up reference. But, unfortunately for our purposes, real muscles don't have this ideal shape.

The reconstructed muscles that we are using in the simulation, consist of a surface based model fitted to the boundary of the medical image data, as one can see in Figure 3.a. This model [3] was developed in the context of the CHARM project and uses imaging data from the Visible Human Dataset (VHD) [1]. The process of surface reconstruction had several steps: data preprocessing, labeling and interpretation, contour fitting, reconstruction and postprocessing.

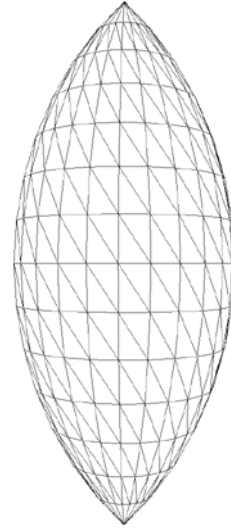


Figure 2. Ideal reconstructed muscle

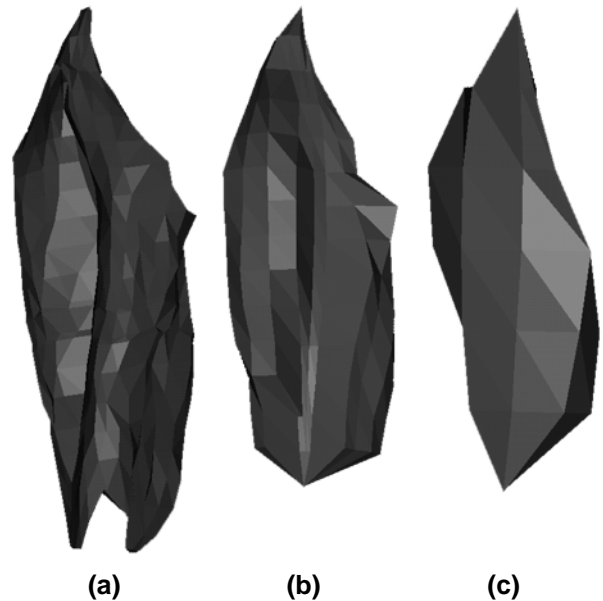


Figure 3. (a) reconstructed Brachialis (557 points); (b) resampled Brachialis with 262 points; (c) with 42 points

As explained above, we need each point in the model surface placed accordingly with the action line position. In this sense, the reconstructed model does not match our needs. To accomplish it, we have resampled this dense irregular triangle mesh into a regular grid. To this end, we have developed a muscle editor where the user can connect the reconstructed muscle to its respective action line, as well as define the number of points desired on the new muscle after the resampling process.

To resample muscles, we have used a simple algorithm conceived exclusively for fusiform muscles. The user can define the number of slices perpendicular to the action

line on the new muscle shape, as well as the number of points on each slice. Excepting the extremities, every slice has the same number of points. The algorithm goes through all the action lines and, at each position of a new muscle slice performs the design of an imaginary circle, draws lines with the origin on the intersection between the action line and the slice; and calculates the intersection between this line and the initial muscle. Each result point of these intersections will be a point of the new muscle.

In Figure 3 we can see the Brachialis muscle reconstructed (Figure 3.a) with two examples of resampled muscles obtained by our editor (Figure 3.b and 3.c). We can also notice that the lower part of the muscle is apparently not well resampled, but this is due to the action line. As it is defined from the muscle top until the intersection between the two lower extremities, the extreme lower parts were not considered.

As explained before, our method admits only fusiform muscles, and the muscle represented on Figure 3.a does not corresponds exactly to this, because of the two lower ends. To solve this problem, we ignore this two extremities, considering that the muscle action line is defined from the uppermost point on the muscle to the lowest end point, if we consider only the fusiform volume. On the other hand, the resampled method permits also the reconstruction considering one of the two extremities as the lowest end.

Note that there is a loss of quality involved in the resampling process. Here we have the same usual dilemma: presentation quality vs speed. As we are looking for real-time deformations our choice is speed. Muscles have a very complicated form which is hard to reproduce. Furthermore, a lot of muscles are covered by others and all of them are covered by the skin. Then, to represent a global animation, we really do not need a high degree of accuracy in muscle form representation.

Fortunately, continuous level of detail and multiresolution techniques [7][15] have recently been studied to solve the conflicting requirements of representation quality and animation speed. In future work we intend to study the possibility of the development of some methods to implement multiresolution in muscle presentation. For this, we are considering two kinds of multiresolution: the spatial one, that aims at changing automatically the degree of discretization of a muscle in function of its position in relation to the others; and the temporal one, that changes the resolution according to the position of the camera, during the visualization process.

4. Physical model

In order to physically simulate the deformation of a muscle, we use mechanical laws of particles. The motion of a particle is defined by its nature and by the position of

other objects and particles in its neighborhood. In our specific case, we have decided to consider only the representation of muscle surfaces, in order to reduce calculations. In fact, we believe we can simulate muscle deformation without considering directly their volume characteristics. The surface is composed by a set of particles with mass density m . Their behavior is determined by their interaction with the other particles that define the muscle surface. In a correspondence with the geometric structure presented before, each point of the mesh corresponds to a particle in the physical model.

The physical model presented here, is based on the application of forces over all mass points that compose the mesh, generating new positions for them. Adding all the applied forces, we obtain a resultant force for each particle on the deformable mesh. For the sake of simplicity, we have considered three different forces: elasticity force, curvature force and constraint force. Then, the resultant force in each particle x_i can be calculated as:

$$f_{result}(x_i) = f_{elasticity}(x_i) + f_{curvature}(x_i) + f_{constraints}(x_i)$$

Further, we present in detail the components involved on the resultant force calculus over a particle, as well as the aspects involved in its definition and calculus. We would like to stress that we have used a vectorial representation of the forces described here.

4.1. Elasticity

To simulate elastic effects between the particles on the muscle mesh, we have used some concepts from the Theory of Elasticity, specifically concerning linear springs. We suppose the connection between each particle and its four neighbors with the use of springs, as you can see in Figure 4.

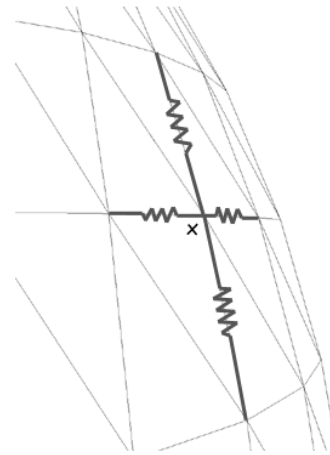


Figure 4. Elastic model of the muscle surface

Considering that the force produced by a linear spring (also known as Hooke's spring) over a mass point is

$$f_{spring}(x_i) = -ks(x_i - x_{rest})$$

where ks is the coefficient that indicates the degree of elasticity of a spring, x_i is the spring's oscillating extremity and x_{rest} is the rest position of this extremity. In extension, we can define the elasticity force as the sum of all forces exerted over the point x_i by the springs connecting itself to its four neighbors. Now we can represent the elasticity force over one particle, as follows:

$$f_{elasticity} = \sum_{j=0}^3 f_{spring_j}(x_i)$$

We have designed two classes of springs: vertical and horizontal ones (both defined in relation to the muscle action line), where the horizontal springs are perpendicular to the action line. The difference between the two is related to the parameters specification of their elasticity coefficients. Furthermore, we have two different degrees of elasticity: in width and height. We consider that the muscle height is defined in the same direction of its action line. Figure 5 shows the minimal elasticity component of a muscle, composed by a spring and a damping module.

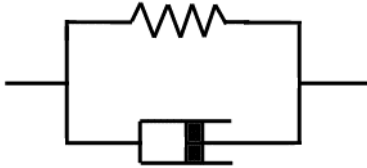


Figure 5. Minimal elasticity component

4.2. Curvature and torsion

The force that determines the degree of bending and twisting of a muscle surface was named curvature and torsion force. As the elasticity force, this force is also calculated for each mass point over the surface, in function of its four neighbors.

The physical simulation of these effects was designed by employing another kind of linear spring, we have created and named **angular springs**. The difference between these springs and the other ones used to simulate elasticity is the way they are attached. Consider a mass point x_0 with a neighborhood formed by x_1, x_2, x_3, x_4 , as shown in Figure 6. Each mass point x_0 has two

correspondent angular springs, one in the angle defined by the vectors $\overline{x_0x_1}$ and $\overline{x_0x_3}$, and the other one in the angle between $\overline{x_0x_2}$ and $\overline{x_0x_4}$.

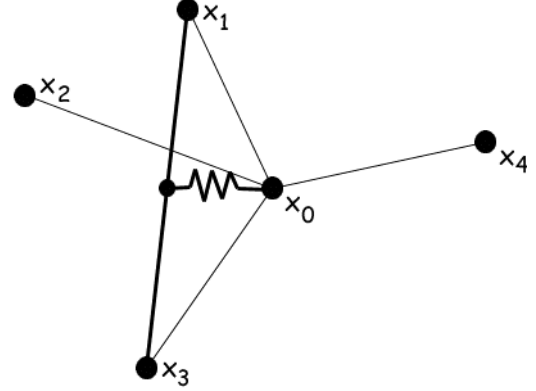


Figure 6. Angular spring between the line segments $\overline{x_0x_1}$ and $\overline{x_0x_3}$

The implementation of this kind of springs is done (consider the spring in Figure 6, placed in the angle between $\overline{x_0x_1}$ and $\overline{x_0x_3}$) by attaching a spring linking the point x_0 with the mid point of the line defined by the vector $\overline{x_1x_3}$, when all the points are in initial position, that is, with all springs in rest position.

As for the springs presented in Section 4.1, we can define the degree of curvature of a muscle in two dimensions. In order to implement this, we allow the specification of curvature coefficients in the two orientations (horizontal and vertical), considering the muscle action line as the up reference. We can say that the horizontal springs are the ones perpendicular to the action line (defined between a mass point and its neighbors over the same muscle slice - see Figure 7.a) and the vertical springs, the springs defined between each mass point and its neighbors on the upper and lower slices (see Figure 7.b).

As we have simulated muscles only by their surfaces, we do not have the representation of a volume, but a surface designing a muscle shape. Like any surface assembled by points linked to their neighbors by springs, it can twist and bend in any direction, changing completely the expected final shape. With the use of angular springs, we succeeded in avoiding this kind of behavior, as shown in Figure 8, where we compare a muscle compression with and without the use of angular springs.

Finally, we have realized that increasing the angular springs coefficients, we are able to control the muscle volume during deformation, that is, we do not need a

post-processing step for this purpose. Obviously, this is not a method that mathematically guarantees the volume preservation. But it allows an efficient and quick way to provide different degrees of deformation to the same initial muscle shape while preserving the homogeneity and guaranteeing some kind of volume control in processing time. In Figure 12 we show an example of muscle compression with different curvature coefficients.

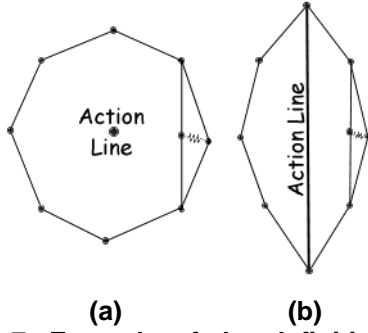


Figure 7. Example of the definition of an horizontal angular spring (a) and a vertical one (b)

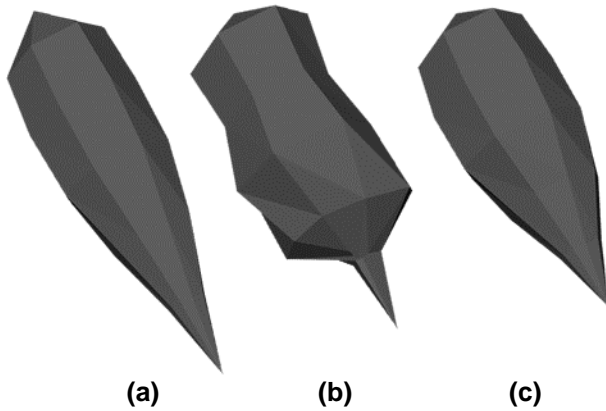


Figure 8. (a) Muscle in the rest position; (b) compressed muscle without angular springs; (c) compressed muscle with angular springs

4.3. Geometric constraints

The geometric constraints were developed to improve the response to the different behaviors not formally explicit in our internal force model, and that can be applied to some specific points or regions of muscles. An example of this situation could be the movement of the points that attach the muscle to the bones. The motion of those points is a function of the bones motion, i. e., if you move your arm, the muscles attached at the corresponding bones also move. Different behaviors can also result from a collision between a muscle and another internal structure, as organs, bones, fat, etc.

The implementation of most part of the constraints forces can be done by using inverse dynamics. We specify the constraints actuating on a point and calculate the force necessary to compel the point to obey these constraints. The methodology used to satisfy constraints, allows the inclusion of new ones, without the need to modify the physical model. At this time, we are using the geometric constraints only to move the attachment points of muscles.

5. Motion simulation

As already mentioned, the lowest level in our muscle simulation system is the action line. We simulate at first the action line motion and after, the muscle deformation. In this way, the muscle compression or extension depends directly on the action line. However, the final deformation depends also on other parameters.

The model used to deform muscles is based on an oriented particle system where the interactions between particles are modeled by springs. The movement simulation is done by applying motion equations over each particle of the model.

We have used the Lagrange's equations of motion as in [31], using a system of second-order ordinary differential equations:

$$m_i \ddot{x}_i + \gamma_i \dot{x}_i + f_{result}(x_i) = f_{extern_i}$$

where $x_i, \dot{x}_i, \ddot{x}_i$ are respectively, the positions, velocities and accelerations, m_i, γ_i are the nodal mass and the damping factor, $f_{result}(x_i)$ is the result force over the node i and f_{extern_i} is an external force.

We use a fourth order Runge-Kutta method [24] to integrate the ordinary differential equations. We have chosen this method because it is more stable than Euler's method while retaining an acceptable time of execution.

6. Some results

To illustrate the applications of the methods described here, we have implemented a system to animate reconstructed muscles using a SGI Impact and Open Inventor for the interface.

Concerning the specification, the user can decide the following: the elasticity and curvature constants, the compression or extension length of the action line, the time step used on the simulation, the total mass of the muscle and the damping factor.

As an example of our method, we have simulated the motion of the Brachialis muscle. The Brachialis has the origin at the front of the lower half of the shaft of the

humerus and the insertion at the front of the coronoid process of the ulna. It is responsible for the flexion of the forearm and can be modeled by only one action line, between the ulna and the humerus. We assume all muscles designed are in a state of rest before the simulation.

In Figure 9 we can see a compression process of the Brachialis muscle. We have used a muscle composed by 50 mass-points and a compression rate of 30%.

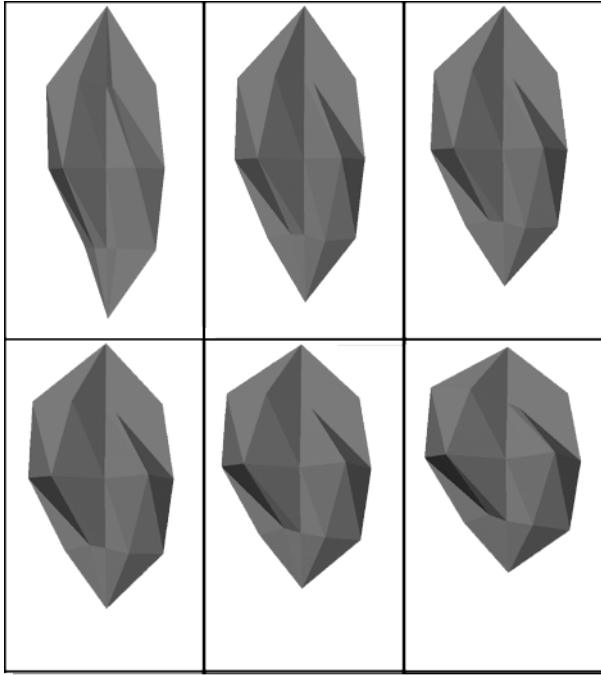


Figure 9. Muscle compression sequence

In Figure 10 we have used the same parameters of the example shown in Figure 9, but with a different movement. In this case, we have simulated an extension rate of 30%.

In Figure 11, we have another example of compression. We have used exactly the same parameters of the other two examples, but this time we have simulated the deformation of the Brachialis with bones motion. Note that the compression rate is much less significant than in the other examples. On the Figure 9 and 10 we have used more compression and extension rates than in reality. Exaggerating the motion, we can show the model's behavior better.

In Figure 12, we present an example of muscle compression using different curvature coefficients. Figure 12.a presents the muscle on its rest state and Figure 12.b, 12.c and 12.d shows the same muscle compressed at 20%. The difference between the three images comes from the curvature coefficient. We have used the same elasticity coefficients on the three examples, but with different

curvature coefficients. From Figure 12.b to 12.c, we have increased the coefficient.

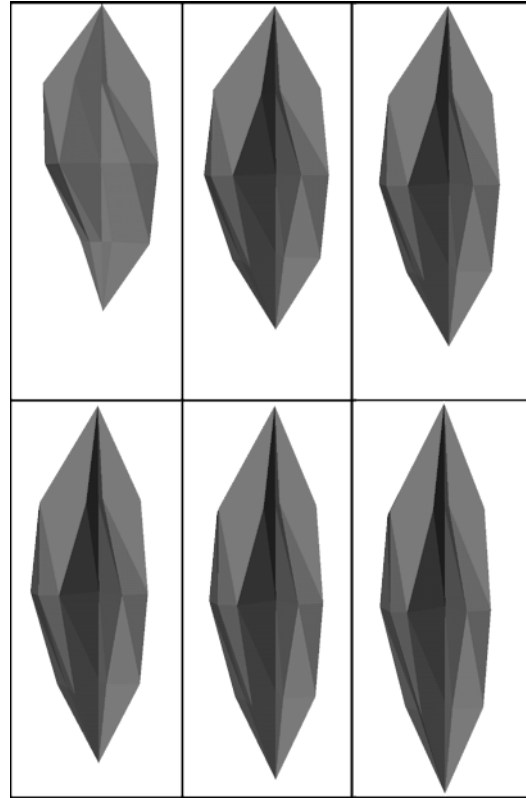


Figure 10. Muscle extension sequence

To conclude, we can state that, not only motion equations are important to achieve a realistic deformation but also the relationship between all the deformation parameters. As we have said before and can be verified on Figure 12, by varying the curvature coefficients we can change the volume of a muscle. The problem is to discover a set of parameters that achieves our needs.

Regarding the system performance, we have made some tests on a SGI Impact workstation, with a MIPS R10000 Processor. The time per calculated image depends of some simulation parameters. In fact, we would like to demonstrate simply how our system works. Just to illustrate, for example, for a simulation of a muscle with 82 mass points, we have reached the performance of 16 images calculated on each second. Using a much more simple muscle with only 17 mass points, the performance was 84 images/sec.

Animations can be seen at:
http://ligwww.epfl.ch/~nedel/muscle_def.html

7. Discussion and future work

This paper has proposed a simple physically-based algorithm to simulate muscles efficiently. As a particular

contribution we have developed the concept of angular springs to control the volume of muscles during the deformation process. The deformation looks natural and the movement continuous, as we can see on the animations examples. About the simulation speed, we consider acceptable to be used in real time applications with only one muscle, but of course, it depends of which resolution we want.

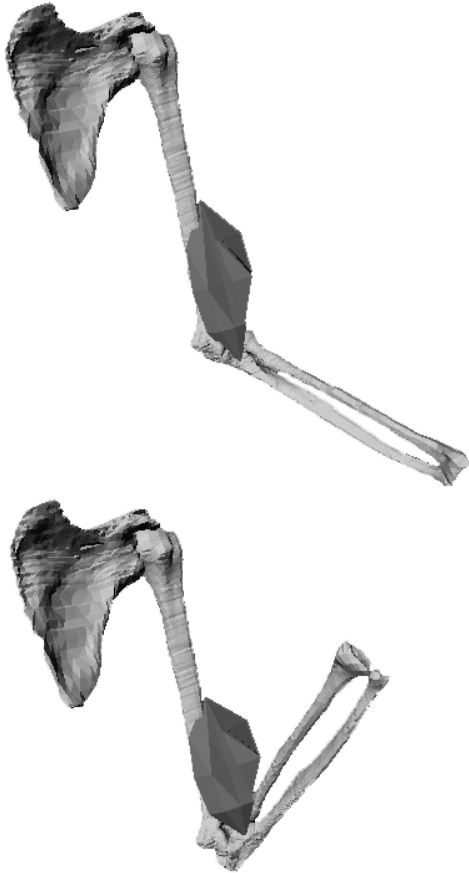


Figure 11. Bones motion with muscle contraction

In future work we intend to:

- investigate an automatic specification of the muscles' action lines motion. To provide this, we intend to use motion capture [20] to acquire the global movement of the arm, calculate the motion of action-lines and the deformation of the muscles;
- provide some automatic way to specify the elasticity and curvature coefficients, and fundamentally the relation between this parameters;
- study the possibility of using multiresolution techniques to discretize muscles;
- model muscles with more than one action line;
- improve the system performance to allow the use of a system with several muscles working in parallel.

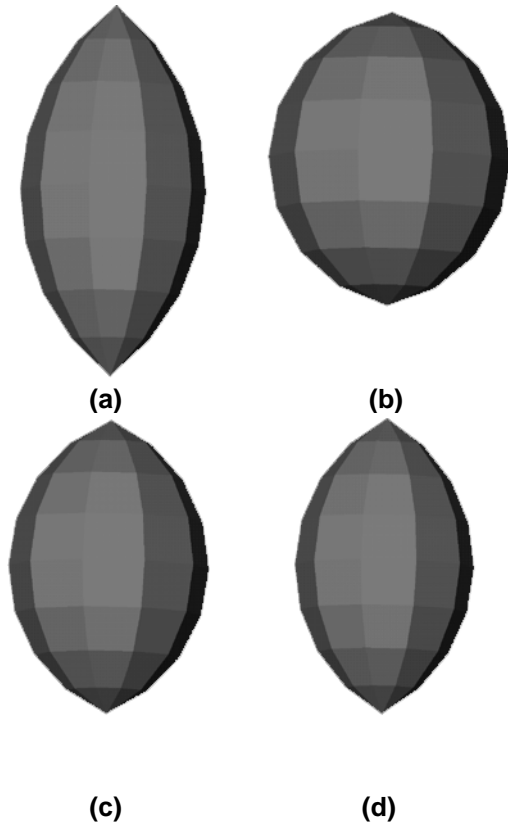


Figure 12. (a) muscle in the rest state; (b) 20% compression and curvature coefficient equal to 1.0; (c) the same compression rate, but with curvature of 5.0; (d) curvature of 10.0

Acknowledgments

We would like to thank Walter Maurel and Pascal Fua for reading the paper and having made useful suggestions and Rafael Bordini and Joaquim Esmerado for English review. We also thank MiraLab, at Geneva for the muscles reconstruction. This work was supported by CAPES-Brazil and by the Swiss National Foundation for Scientific Research.

References

- [1] Ackermann, M. J. The Visible Human Project, <http://www.nlm.nih.gov>, 1995
- [2] Barr, A. H. "Global and Local Deformations of Solid Primitives", *Computer Graphics*, v. 18, n. 3, 1984.
- [3] Beylot, P., Gingins, P., Kalra, P., Thalmann, N., Maurel, W., Thalmann, D. and Fasel, J. "3D Interactive Topological Modeling using Visible Human Dataset", *EUROGRAPHICS'96*, v. 15, n. 3, p. C-34-C-44, 1996.

- [4] Blinn, J. F. "A Generalization of Algebraic Surface Drawing", ACM Transactions on Graphics, v. 1, n. 3, p. 235-256, 1982.
- [5] Bloomenthal, J. and Shoemaker, K. "Convolution Surfaces", Computer Graphics, v. 25, n. 4, p. 251-256, 1991.
- [6] Bro-Nielsen, M. and Cotin, S. "Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation", EUROGRAPHICS'96, v. 15, n. 3, p. C-57-C-66, 1996.
- [7] Certain, A., Popovic, J., DeRose, T., Duchamp, T., Salesin, D. and Stuetzle, W. "Interactive Multiresolution Surface Viewing", Computer Graphics, New Orleans, ACM SIGGRAPH, 1996.
- [8] Chen, D. T. and Zeltzer, D. "Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method", Computer Graphics, v. 26, n. 2, p. 89-98, 1992.
- [9] Cody, M. D. J. "Visualizing Muscles - A New Ecorché Approach to Surface Anatomy", University Press of Kansas, 1990. 231 p.
- [10] Coquillard, S. and Jancène, P. "Animated Free-Form Deformation: An Interactive Animation Technique", Computer Graphics, v. 25, n. 4, p. 23-26, 1991.
- [11] Desbrun, M. and Gascuel, M. -P. "Animating Soft Substances with Implicit Surfaces", Computer Graphics, Los Angeles, California, ACM SIGGRAPH, 1995.
- [12] Fung, Y. C. "Bioviscoelastic Solids", Biomechanics: Mechanical Properties of Living Tissues, 1993.
- [13] Holton, M. and Alexander, S. "Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials", Computer Graphics: Developments in Virtual Environments, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd, p. 449-460, 1995.
- [14] Huxley, A. F. and Niedergerke, R. "Structural changes in muscle during contraction", Nature, v. 173, p. 971-973, 1954.
- [15] Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L. F., Faust, N. and Turner, G. A. "Real-Time, Continuous Level of Detail Rendering of Height Fields", SIGGRAPH'96, New Orleans, ACM SIGGRAPH, 1996.
- [16] Maurel, W., Thalmann, D., Hoffmeyer, P., Beylot, P., Gingins, P., Kalra, P. and Magnenat-Thalmann, N. "A Biomechanical Musculoskeletal Model of Human Upper Limb for Dynamic Simulation", Computer Animation and Simulation'96, p. 121-136, 1996.
- [17] Metaxas, D. and Terzopoulos, D. "Dynamic Deformation of Solid Primitives with Constraints", Computer Graphics, v. 26, n. 2, p. 309-312, 1992.
- [18] Miller, G. "The motion of snakes and worms", Computer Graphics, v. 22, n. 4, p. 169-177, 1988.
- [19] Moccozet, L. and Magnenat-Thalmann, N. "Dirichlet Free-Form Deformations and their Application to Hand Simulation", In: Computer Animation'97, p. 93-102, Geneva, Switzerland, June, 1997.
- [20] Molet, T., Boulic, R. and Thalmann, D. "A Real Time Anatomical Converter for Human Motion Capture", Computer Animation and Simulation'96, 1996.
- [21] Nishimura, H., Hirai, M., Kawai, T. *et al.* "Object modeling by distribution function and a method of image generation", Transactions of IECE J68-D, n. 4, p. 718-725, 1985.
- [22] Platt, J., Terzopoulos, D., *et al.* "Elastically Deformable Models", SIGGRAPH'88 Course Notes #27: Developments in Physically-based Modeling, p. K1-K25, 1988.
- [23] Platt, J. C. and Barr, A. H. "Constraints Methods for Flexible Models", Computer Graphics, v. 22, n. 4, p. 279-288, 1988.
- [24] Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. "Numerical recipes in C - The Art of Scientific Computing - Second Edition", Cambridge University Press, 1992. 994 p.
- [25] Promayon, E., Baconnier, P. and Puech, C. "Physically-Based Deformations Constrained in Displacements and Volume", EUROGRAPHICS'96, v. 15, n. 3, p. C-155-C-164, 1996.
- [26] Reeves, W. T. "Particle systems: a technique for modeling a class of fuzzy objects", ACM Transactions and Graphics, v. 2, n. 2, 1983.
- [27] Sederberg, T. W. and Parry, S. R. "Free-Form Deformation of Solid Geometric Models", Computer Graphics, v. 20, n. 4, p. 151-160, 1986.
- [28] Scheepers, F., Parent, R. E., Carlson, W. E. and May, S. F. "Anatomy-Based Modeling of the Human Musculature", Computer Graphics Proceedings, Annual Conference Series, 1997.
- [29] Shen, J. and Thalmann, D. "Interactive Shape Design using Metaballs and Splines", In: Implicit Surfaces'95, Grenoble, France, April, 1995.
- [30] Szelinski, R. and Tonnesen, D. "Surface Modeling with Oriented Particle Systems", Computer Graphics, v. 26, n. 2, p. 185-194, 1992.
- [31] Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K. "Elastically Deformable Models", Computer Graphics, v. 21, n. 4, p. 205-214, 1987.
- [32] Terzopoulos, D. and Fleischer, K. "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture", Computer Graphics, v. 22, n. 4, p. 269-278, 1988.
- [33] Terzopoulos, D. and Witkin, A. "Physically Based Models with Rigid and Deformable Components", IEEE Computer Graphics and Applications, p. 41-51, 1988.
- [34] Terzopoulos, D., Platt, J., *et al.* "Heating and melting deformable models (from goop to glob)", Computer Interface'89, Ontario, 1989.
- [35] Turner, R. "LEMAN: A System for Constructing and Animating Layered Elastic Characters", Computer Graphics: Developments in Virtual Environments, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd, p. 185-203, 1995.
- [36] Weil, J. "The Synthesis of Cloth Objects", Computer Graphics, v. 22, n. 4, p. 269-278, 1988.
- [37] Wilhelms, J. "Modeling Animals with Bones, Muscles, and Skin", Santa Cruz, CA, Baskin Center for Computer Engineering and Information Sciences, University of California, 1994.
- [38] Wilhelms, J. and Van Gelder, A. "Anatomically Based Modeling", Computer Graphics Proceedings, Annual Conference Series, 1997.
- [39] Wyvill, G., McPheeters, C., *et al.* "Data Structure for Soft Objects", The Visual Computer, v. 2, n. 4, 1986.