The Visual Computer

# Anatomic modeling of deformable human bodies

Luciana Porcher Nedel,
Daniel Thalmann

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

In this paper, we propose a method to simulate human bodies based on anatomy concepts. We believe that the closer our model is to reality, the better our results will be. Using this approach, we are developing our human representation. Our model is divided into three layers and presented in three steps: the concept of a rigid body from a real skeleton, the muscle design and deformation based on physical concepts, and skin generation. Muscles are represented at two 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 surface-based models. 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. Integration aspects and results are also presented.

**Key words:** Anatomic modeling – human body deformation – angular springs – action lines

*Present address and correspondence to*: L.P. Nedel, Computer Science Department, Federal University of Rio Grande do Sul (UFRGS), Brazil
e-mail: nedel@inf.ufrgs.br;
html: http://www.inf.ufrgs.br/∼nedel

## 1 Introduction

For many years, modeling and animation of the human bodies has been an important research goal in computer graphics. Initially, humans were represented as simple articulated bodies made of segments and joints with a kinematic model to simulate them. Since then, dynamic models have been used to improve the continuity and fluidity of the movement.
For instance, simple skeleton-based models are limited to human picture simulation when, in fact, a human body is a collection of complex rigid and non-rigid components that are 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 became very exacting about human simulation. Consequently, researchers began to use knowledge about human anatomy to produce human models with more realistic behavior.
In this paper, we describe our efforts in the human modeling aspects. Our first objective is to improve the complexity and the realism of motion. However, realism of motion needs to be improved, not only from the standpoint of joints as for robots, but also in relation to the deformation of bodies during the animation. As the final motion is directly related to the model complexity, we propose the modeling of a body based on anatomical concepts.
In the next sections, we describe our human model, concentrating on the aspects concerning the skeleton, the muscles, and the skin. As already stated, we intend to increase the realism of the motion improving body deformations and, for that reason, we focus this paper on deformable muscle modeling, our main contribution.

## 2 Background

There are two research directions in human modeling: the development of new motion algorithms and the realism of the body shape. The research on human animation started in the 1970s (Badler and Smoliar 1979) with the representation of humans by stick figures with only few articulations joining the segments animated by a kinematic model. More accurate models were developed, with bodies being composed of more joints and segments; there were improvements on some intricate regions of the body, like the shoulder (Magnenat-Thalmann and Thalmann 1991) and the spine (Monheit and Badler 1991).

The Visual Computer (2000) 16:306–321

To achieve more realistic movements, researchers introduced physically based models, and because humans do not act the same, they also introduced behavioral models (Badler 1992) to take into account the individuality of each character. More recently, the evolution of motion control models has made it possible to take into account the interactions among actors (Becheiraz and Thalmann 1996), actors with the environment, and actors with real humans (Balcisoy and Thalman 1997). In summary, the goal of all these applications is looking for realism of the motion.

Concerning the body modeling, important efforts have been made to represent and deform the human body shape. The models presented so far can be classified into four categories: stick figure models, surface models, volume models, and multilayered models.

Systems using stick figure models consist of a hierarchical set of rigid segments (limbs) connected at joints. These models, called articulated bodies, may be more or less complex, depending on the number of limbs and joints involved. Surface models are conceptually simple, containing a skeleton and an external shape, the skin (Magnenat-Thalmann and Thalmann 1987, Komatsu 1988, Forsey 1991). The main problem with the surface model is that it is hard to control the behavior of the surface deformation across joints.

The volume models approximate the structure and the shape of the body with a collection of elementary volume primitives: ellipsoids, spheres, cylinders, etc. An example of the use of these primitives are “metaballs”, which consider the volume as a potential function and produce good, natural results (Yoshimoto 1992). These models can generate better results than the surface models, but suffer from inadequate control mechanisms for a large number of primitives during the animation.

Finally, the multilayered models contain the skeleton layer, intermediate layers to represent the body volume (muscles, fat, bones, and so on), and the skin layer. A good example of layered construction and animation of deformable characters is presented by Chadwick et al. (1989). Other multilayered models were also developed, as the ones presented by Gascuel et al. (1991), Singh et al. (1995), Turner (1995) and Shen and Thalmann (1995), for example.

Recently, multilayered techniques were applied to anatomically based models of humans and animals. Scheepers et al. (1997) have presented a model where muscles (represented by deformable ellipsoids) react automatically to changes in the posture of the articulated skeleton and influence on the surface form. Wilhelms and Van Gelder (1997) propose a multilayered, anatomically based model to simulate animals. In their model, muscles are represented by deformable discretized cylinders.

## 3 Skeleton design

The human body can be briefly defined as a composition of skeleton, muscles, fat, and skin. The skeleton is formed by bones (about 206 in total) attached to each other by joints, and constitutes the foundation of all human surface form. The base of a skeleton is the spinal column where the two pairs of limbs (arms and legs) and the head are connected.

We also use the term skeleton in computer animation to designate a stick figure representing the positions and orientations of the joints that build up the articulated figure. To represent our stick-made human model, we have used the basic skeleton structure proposed by Boulic and Renault (1991) and Boulic et al. (1994). The skeleton hierarchy is composed of articulated line segments whose movements are described at the joints level.

Regarding implementation aspects, a BODY data structure maintains some generic information, including a topological tree structure for a vertebrate body with predefined mobility. The hierarchical topology orients the propagation of motion from the root to the terminal nodes. Each joint of a body is represented by one or more nodes of the tree, depending on the joint’s degrees of freedom (DOFs). Joints have up to three DOFs and each DOF corresponds to one node of the tree. Each node position is defined locally in relation to its parent node in the hierarchy.

Our simplified model of the human skeleton contains 31 joints with 62 DOFs. We do not represent either the joints of the head or the facial animation.

As mentioned before, the main goal of this work is to produce a human model based on anatomic principles. At the skeleton level, we suggest the creation of a simplified body with the joint positions defined anatomically. To accomplish this, we propose the design of a new human template defining new positions and orientation for the joints. However, finding the location of the center of a joint from the knowledge of external markers is a very difficult task. Joints like the shoulder or the hip, for example, are particularly hidden within the flesh and complex bone structure.

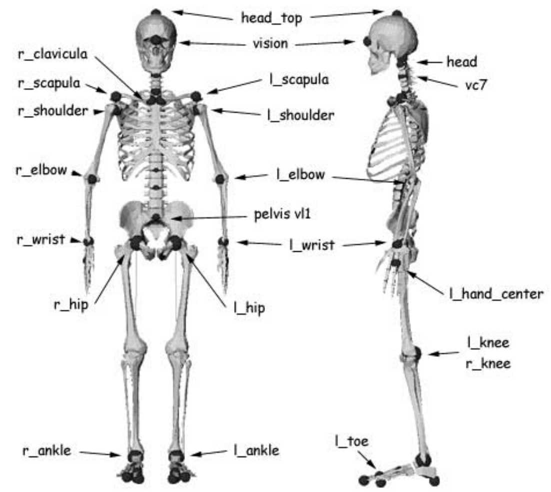
We propose to define the template from an analysis of a real 3D skeleton. From a realistic, reconstructed human skeleton, we have carefully determined the best position for each joint between the bones. The determination is based on anatomic concepts and motion observation. In Fig. 1, we can see the skeleton with bone reference points highlighted, representing the positions of the joints.

The higher part of the body hierarchy begins with the spine, which is a particularly interesting and complex structure. It consists of 25 bony components: 7 cervical, 12 thoracic and 5 lumbar vertebræ, and the sacrum with the coccyx.

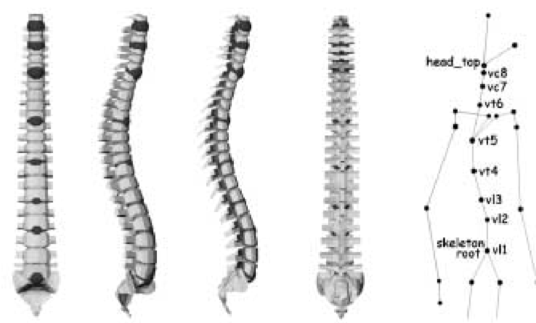
We have modeled the spine with eight compound vertebrædistributed in three groups that do not possess the same mobility: the three lumbar vertebræcannot have torsion; the three thoracic vertebræhave limited or no tilt; and the two cervical vertebræhave all three rotational mobilities (torsion, tilt, and roll). The name of the compound vertebra includes the first letter of the region it belongs to (l for lumbar, t for thoracic, and c for cervical). The first lumbar vertebra sits on the pelvic sacrum, while the last cervical vertebra (vc8) is under the atlas joint, corresponding to the little remaining distance to the head root (Fig. 2). The arms are attached to the spine through one of the three thoracic vertebræ(vt5).

Over the spine, the most difficult body part to model is the upper arm and the shoulder complex, which is also known to be the most complicated joint in the body. It is made up of three bones: the scapula, the clavicle, and the humerus. The joint itself is a ball-and-socket joint, the head of the humerus fitting into a socket in the outer edge of the scapula. The head of the humerus is much larger than this socket on the scapula. Because of this, the shoulder joint is very flexible, allowing the arm to point to many directions.

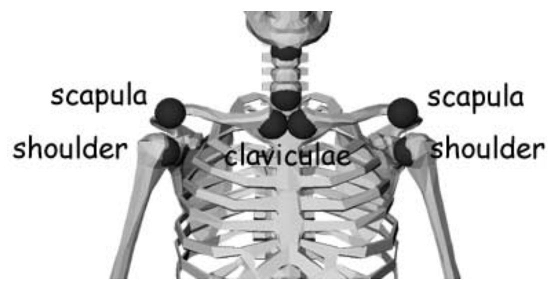
We have chosen to represent the shoulder by three joints, with seven DOFs: clavicle and scapula (abduction and rotation); and shoulder (flexion, abduction, and twisting). Strictly speaking, the arm structure is attached to the spine as a child of the fifth compound vertebra (vt5), while the two clavicle DOFs and the two scapula DOFs are not part of the arm structure; they only initiate its articulated chain. Figure 3 shows the bone reference points that define the center of rotation for the three joints.




1




2




3

**Fig. 1a,b.** Frontal (**a**) and lateral (**b**) views of the skeleton with the joint positions highlighted

**Fig. 2.** Several views of the spine with eight compound vertebræ

**Fig. 3.** Center of rotation of the clavicle, scapula, and shoulder articulations

Bones are linked to the joint reference systems and modeled by triangle meshes defined locally in relation to their parent joint. Our system is modeled with 73 bones, ignoring the hands and feet that are represented by whole sets of bones. Bones do not change shape during animation, but can change position in relation to each other.

Our main goal in representing bones is to allow muscle attachment definition (skeleton muscles are fixed on the bones by their extremity points, in general). Esthetic aspects were also considered. In fact, in some parts of the body, the skeleton contributes directly to the external appearance and should be considered during the skin-generation process. This is the case for the lower part of the legs, the elbow, and some ribs, if we consider a slim person. Furthermore, parts of bones that appear not to create any surface form in some postures, do so in other postures.

Another reason pushed us to introduce bone shapes in our model. The visualization of the skeleton during animation allows us to avoid forbidden postures, a tedious and abstract task when it is done on a stick figure. The definition of the limit angles for each joint also became more precise when bone representation between articulations was used.

## 4 Muscles

Among the anatomical systems that determine the skin shape, the musculature is the most complex. Three kinds of muscles can be identified in the human body: skeletal, cardiac, and smooth muscles. As our purposes involve the representation of the human body, and more precisely, the external appearance, we have modeled only skeletal muscles, which are referred to in this work simply as *muscles*. Skeletal muscles are arranged side by side and in layers on top of the bones and other muscles. Located throughout the entire body, the skeletal muscles make up 40% to 45% of the total body weight.

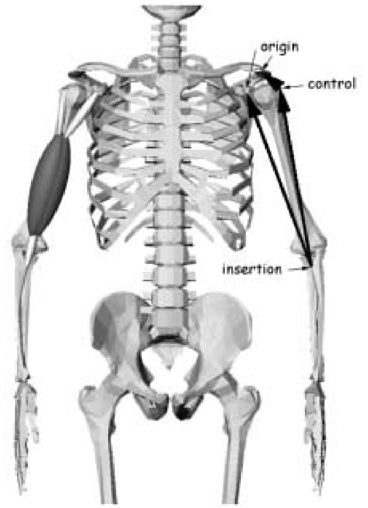
### *4.1 Action lines*

To mechanically quantify the force that a muscle produces over a bone, we represent muscles by lines, usually named *action lines*. Muscles are attached to bones at two points. The attachment that lies closer to the spine is generally considered the *origin*, while the other one is said to be the *insertion*. Normally, the origin is the fixed point, and the insertion is where the muscle performs its action, but there is no formal rule. 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 (Cody 1990).

In our model, a muscle is represented at two levels: the action line and the muscle shape. To simulate the muscle forces, muscles can be represented by one or more action lines, basically defined by an *origin* and an *insertion* point, as already described. These points represent the links between the muscles and the bones, and sometimes they also correspond to the extremities of the muscle shape. However, depending on the shape, position, and complexity of the muscle, simple action lines are not enough to represent the force produced over the skeleton. For this reason, we have decided to represent the actions of the muscles, not simply by a straight line, but by using polylines. To accomplish this, we have developed the concept of *control* points, whose objective is to guide the line, avoiding the intersection with the bones. An example of this kind of action line is shown in Fig. 4. The *biceps brachii* is represented by two action lines (one for each of the two muscle bellies), but in one of them we need to use a control point. If we try to represent this action line by a straight line, the bone is intercepted.

Other examples of intersection between bones and action lines can be perceived only while we animate the skeleton. This is exactly the case, for example, of the *triceps brachii*, which are represented by three action lines (one for each muscle belly) attaching the upper arm to the upper part of the lower arm. Figure 5 shows the definition of the action line that represents the *triceps middle head*. In Fig. 5a, we can see the skeleton of the arm with the defined action line when the muscle is in its rest position. Figure 5b shows the arm contraction and the resultant action line, as if it were defined by a simple straight line. We can see in the example that if the action line is not correct, the muscle can contract during an extension action. Finally, Fig. 5c shows the correct simulation of the triceps action line, for which some control points are used.

Moreover, Fig. 5 also shows an example of the muscle shape simulation and reaction to the arm motion. In these examples, we can verify that the muscle belly is not attached to the bones by the origin and insertion of the action lines. It has its own origin and insertion points, simulating, in this way, the presence of tendons. In order to simulate the tendons, we assume that the action line is, in fact, a thread link-




4

5a 5b 5c

**Fig. 4.** The *biceps brachii* muscle and corresponding action lines

**Fig. 5a–c.** The *triceps brachii middle head* action representation: **a** bones of the arm and triceps action line with the muscle in its rest position; **b** action line and muscle reaction during the arm contraction with a straight action line; **c** correct simulation of the action line using additional control points

ing two bones and crossing some pulley, here represented by the control points. Tendons stretch just about 8% of their original length, but, for the sake of simplicity, we define the tendons by defining certain distances between the muscle boundaries and the action lines extremities. During the animation, these distances are kept constant.

## *4.2 Shape design*

To simulate muscle deformations, we use a set of mass points linked by springs and organized so that a correspondence between the action line and the muscle shape is ensured. We consider, for all the cases, that each muscle belly involves completely an action line or a part of an action line. Each point that composes the surface of a muscle is arranged between two neighbors in the horizontal direction and two others in the vertical direction, considering the action line of a muscle as the up reference. In this sense, we can also say that the extremities of the muscle shape have the same position as the action lines extremities or, if we are using tendons, that they are placed on the action line.

Taking into account the limitations imposed by our deformation model, which was designed specifically to simulate fusiform muscles, we developed a resampling method whose goal is the generation of simple and uniform muscle surfaces. From a muscle form composed of triangles, this method is able to generate another muscle shape designed to achieve our objectives.

To satisfy the needs of our deformation model, each point in the model surface should be placed according to the action line position. To this end, we have developed a muscle editor to help the user 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 the given dense, irregular, triangle meshes into regular grids (Fig. 6 shows an example of a resampled triceps), the user needs to define the number of slices perpendicular to the action line on the new muscle shape, as well as the number of points in each slice. Excepting the extremities, every slice has the same number of points. The algorithm goes through all the action line and, at each position of a new muscle slice, designs an imaginary circle. Lines are drawn in a star-shaped manner with the origin on the intersection between the action line and the slice. For each line, we compute the outermost intersection point with the initial muscle. Each resulting point of these intersections will be a point over the new muscle surface.

Note that there is a loss of quality involved in the resampling process. Here we have the usual compromise: presentation quality versus speed. Muscles have a very complicated form that is hard to reproduce. Furthermore, many muscles are covered

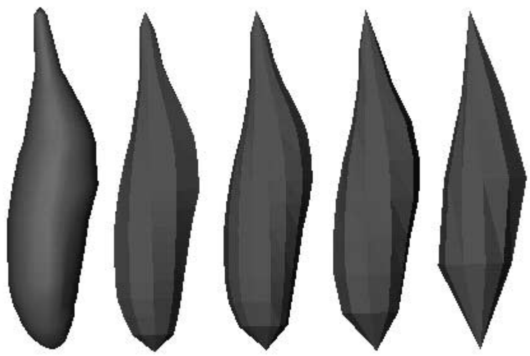


**Fig. 6.** Model of the internal part of the triceps (with 2160 points) and the resampled models with 197, 182, 82, and 32 points

by others, and all of them are covered by the skin. Then, to represent a global animation, we do not really need a high degree of accuracy in muscle shape representation.

Continuous level-of-detail and multiresolution techniques (Certain et al. 1996, Lindstrom et al. 1996) have recently been studied to solve the conflicting requirements of presentation quality and animation speed. In future work, we intend to study the integration of some multiresolution methods in muscle presentation. To this end, we are considering two kinds of multiresolution: the spatial one, which aims at automatically changing the degree of discretization of a muscle in the function of its position in relation to the others; and the temporal one, which changes the resolution according to the position of the camera, during the visualization process.

### 4.3 Deformation model

The problem of simulating deformable objects is that it can be seen as a continuous system in space and time. The first step to solve this system is to discretize the continuous equations in material coordinates, which results in a large system of simultaneous ordinary differential equations. The shape of a body is determined by the euclidean distances between nearby points, while the evolution of the object in the scene depends on the forces acting on its points. The second step is the time discretization that generates the evolution of the objects in time.

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 directly considering their volumetric characteristics. The surface is composed of a set of particles with point mass $m$. Their behavior is determined by their interaction with the other particles that define the muscle surface. In connection to the geometric structure already presented, 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 the 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 forces: elasticity, curvature, and constraint force. Then, the resultant force in each particle $i$ can be calculated as:

$$f_{\text{results}}\left(x_i, x_i^0, x_i^1, x_i^2, x_i^3\right) = \left[f_{\text{elasticity}} + f_{\text{curvature}} + f_{\text{constraints}}\right]\left(x_i, x_i^0, x_i^1, x_i^2, x_i^3\right),$$

where $x_i$ is the vector position of the particle $i$ and $(x_i^0, x_i^1, x_i^2, x_i^3)$ the positions of the particles that compose its neighborhood.

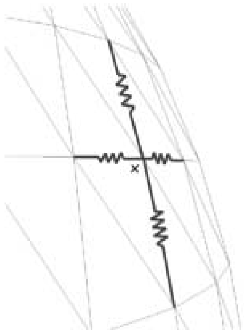
We present in detail the components involved on the resultant force application on a particle, as well as the aspects involved in its definition and calculus. We would like to emphasize that we have used a vectorial representation of the forces described here.

*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 consider the connection between each particle and its four neighbors, with the use of springs, as one can see in Fig. 7.

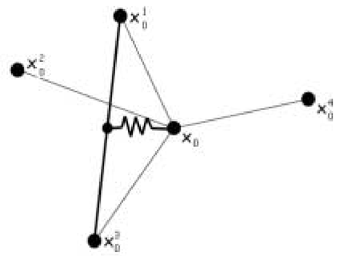
Considering the force produced by a linear spring (also known as Hooke's spring) over a mass point as

$$f_{\text{spring}_{ij}}(x_i, x_j) = -k_{s_{ij}}\left[(x_j - x_i) - (x_{j_0} - x_{i_0})\right]u_{ji}\ ,$$

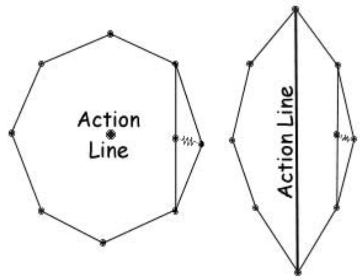
where $k_{s_{ij}}$ is the coefficient that indicates the degree of elasticity of a spring, $x_i$ is position of the spring's oscillating extremity, and $x_j$ is the position of the fixed extremity. $x_{i_0}$ and $x_{j_0}$ are the positions of extremities $i$ and $j$ when the spring is in the rest state and $u_{ji}$, the unit vector from $j$ to $i$. Further, we can



**7**



**8**




**9a** **9b**

**Fig. 7.** Elastic model of the muscle surface
**Fig. 8.** Angular spring between the line segments $\boldsymbol{x_0 x_0^1}$ and $\boldsymbol{x_0 x_0^3}$
**Fig. 9a,b.** Example of the definition of a horizontal angular spring (**a**) and a vertical one (**b**)

define the elasticity force as the sum of all forces exerted over the point $x_i$ by the springs connecting it to its four neighbors. Now we can represent the elasticity force over one particle, as follows:

$$f_{\text{elasticity}}\left(x_i, x_i^0, x_i^1, x_i^2, x_i^3\right) = \sum_{j=0}^{3} f_{\text{spring}_{ij}}(x_i, x_j) .$$

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 given by the specification of their elasticity coefficients. Furthermore, we have two degrees of elasticity: in width and height (always relating the muscle action line). We assume that the muscle height is defined in the same direction as its action line.

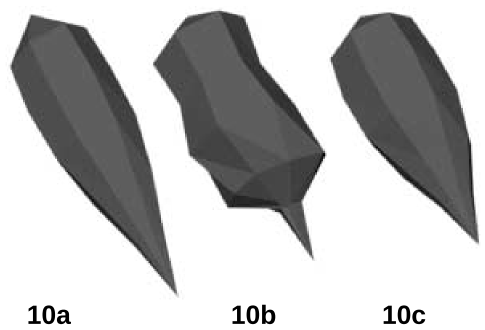
*Curvature and torsion*. The force that determines the degree of bending and twisting of a muscle surface was named the curvature and torsion force. As the elasticity force, this force is also calculated for each mass point on the surface as a function of its four neighbors.

The physical simulation of these effects was designed by employing another kind of linear spring we have created and named the *angular spring*. The difference between these springs and the ones used to simulate elasticity is the way they are attached. Consider a mass point $x_0$ with a neighborhood formed by $x_0^1, x_0^2, x_0^3, x_0^4$, as shown in Fig. 8. Each mass point $x_0$ has two corresponding angular springs, one in the angle defined by the vectors $\boldsymbol{x_0 x_0^1}$ and $\boldsymbol{x_0 x_0^3}$, and the other one in the angle between $\boldsymbol{x_0 x_0^2}$ and $\boldsymbol{x_0 x_0^4}$.

This kind of spring (consider the spring in Fig. 8, placed in the angle between $\boldsymbol{x_0 x_0^1}$ and $\boldsymbol{x_0 x_0^3}$) is implemented by attaching a spring linking the point $x_0$ with the midpoint of the line segment defined by the vector $\boldsymbol{x_0^1 x_0^3}$, when all the points are in their initial positions (that is, with all springs in rest position).

As for the springs used to simulate the elasticity, 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 Fig. 9a) and the vertical springs, i.e., the springs defined between each mass point and their neighbors on the upper and lower slices (see Fig. 9b).

As we simulate 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, completely changing the expected final shape. With the use of angular springs, we succeeded in avoiding this kind of behavior, as shown in Fig. 10, where we compare



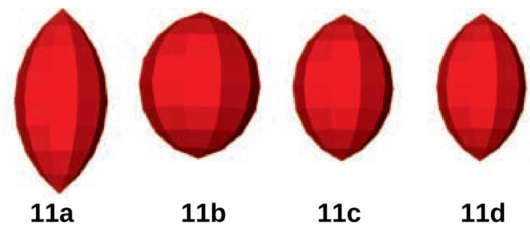




**Fig. 10. a** Muscle in the rest position; **b** compressed muscle without angular springs; **c** compressed muscle with angular springs

**Fig. 11. a** Muscle in the rest state; **b** 20% compression and curvature coefficient equal to 1.0; **c** the same compression rate, but with a curvature of 5.0; **d** curvature of 10.0

a muscle compression with and without the use of angular springs.

Finally, we realized that by increasing the angular spring-coefficients, we were 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, preserving the homogeneity and guaranteeing some kind of volume control during the processing phase itself. In Fig. 11, we show an example of muscle compression with various curvature coefficients.

*Geometric constraints*. The geometric constraints were defined to improve the response to the various conditions not formally explicit in our internal force model, and can be applied to some specific points (local constraints), regions of muscles (zonal constraints) or to all the muscle (global constraints). An example of local constraint is the movement of the points that attach the muscle to the bones. The movement of these points is a function of the bone motion; that is, if you move your arm, the muscles attached at the corresponding bones will also move. Different conditions can also result from a collision between a muscle and another internal structure, such as organs, bones, fat, etc. One way to implement this case is by defining a zonal constraint. With the whole body in the rest position, the user defines the regions of the muscle that are in contact with another organ, bone, etc. and apply the same constraint to all the particles in the region. This constraint can be, for example, the conservation of a kind of link between a muscle particle and the surface of the other organ.

A great part of the force constraints can be implemented with inverse dynamics. We specify the constraints acting on a particle and calculate the force necessary to compel the particle to obey these constraints. This is also called the *induced force*. Knowing the position where a particle $i$ should be located, we apply a force that minimizes the distance by creating a point-to-point displacement. If $x_i$ is the current particle position in euclidean space and $x_{i_{\text{goal}}}$ is its ideal position, a force $f$ should be applied to the particle $i$. This force $f$ is the means of handling animation control when there is a need for a particle to follow paths or to be guided by key-frames. To accomplish that we need to define a new $x_{i_{\text{goal}}}$ on the path at each time step.

The methodology used to satisfy constraints allows the easy inclusion of new ones, without the need to modify the physical model. At present, we are using the geometric constraints only to move the attachment points of muscles.

### 4.4 Deformation parameters

The main parameters that the user needs to set up in order to perform deformations are: two elasticity coefficients, two curvature coefficients, the muscle mass and a damping factor used during the motion simulation. The coefficients consist of physical parameters used in the calculation of elasticity and curvature forces. As we have designed two classes of springs, vertical and horizontal ones (each defined as a function of the muscle action line), we also have two elasticity coefficients, considering that the horizontal springs are perpendicular to the action line. The same idea is applied to the curvature springs, where we can choose the degree of curvature in width and height, with vertical and horizontal springs. The user can also define the total mass of the muscle, subsequently divided by the number of particles that constitutes its surface.

### 4.5 Motion simulation

The movement is simulated by applying systems of motion equations to each particle of the model. These systems rely on second-order differential equations, derived from a definition of elasticity, being the resistance to expansion of a material, and viscosity, being the resistance to a change in expansion for a material (Holton 1995).

We have used Lagrange's equations of motion as in Terzopoulos et al. (1987):

$$m_i \ddot{x}_i + \gamma_i \dot{x}_i + f_{\mathrm{result}}(x_i, x_i^0, x_i^1, x_i^2, x_i^3) = f_{\mathrm{extern}_i}$$

where $x_i$, $\dot{x}_i$, $\ddot{x}_i$ are, respectively, the positions, velocities, and accelerations of their mass elements as functions of material coordinates and time; $m_i$, $\gamma_i$ are the nodal mass and the damping factor that dissipates the kinetic energy of the body's mass elements; and $f_{\mathrm{result}}(x_i, x_i^0, x_i^1, x_i^2, x_i^3)$ is the result force over the node $i$; and $f_{\mathrm{extern}_i}$ is an external force. The external force is balanced against the force terms on the left-hand side of the equation due to the deformable model. The first term represents the inertial force due to the model's distributed mass. The second term is the damping force due to dissipation, and the third term is the elastic force due to the deformation of the model away from its natural shape.

To create animations simulating the dynamics of elastic models, the ordinary differential equations of motion are integrated through time. We use a fourth-order Runge-Kutta method (Press et al. 1992) for this. We have chosen this method because it is more stable than Euler's method while it retains an acceptable execution time.

### 4.6 Results

As an example of our deformation method, we have simulated the motion of the brachialis muscle, reconstructed from the images of the Visible Human Dataset (VHD). 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 that all the muscles designed are in a state of rest before the simulation.

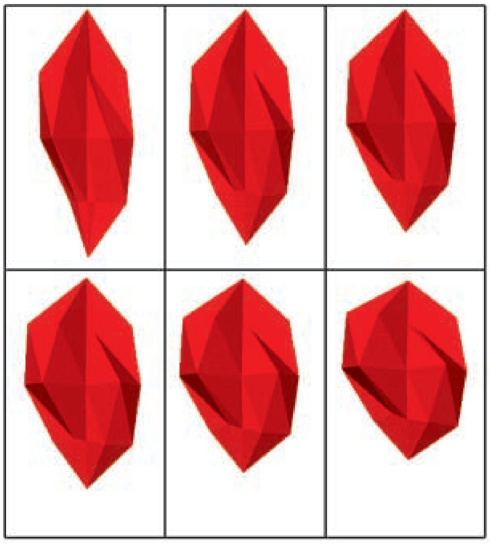
In Fig. 12, we can see a compression process of the brachialis muscle. We have used a muscle composed of 50 mass points and a compression rate of 30% of the total length.

In Fig. 13, we have another example of compression. We have used exactly the same parameters as for the first example, but this time we have simulated the deformation of the brachialis with bone motion. Note that the compression rate is much less significant than that in Fig. 12, where we exaggerated the motion, in order to show the model's behavior better.

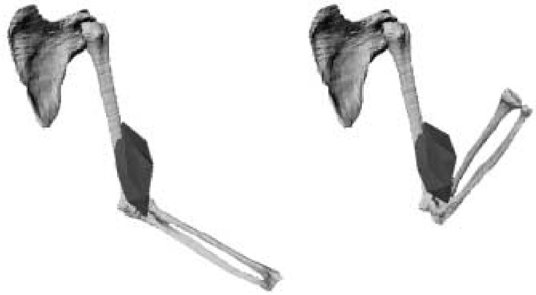
## 5 Skin layer

To generate the skin covering all the bones and muscles, we use a sampled surface and ray casting on semiregular cylindrical grids. These sample points are used directly as cubic B-spline control points to smooth out the skin surface. Individual B-spline patches are triangulated, and these triangular meshes are stitched together to connect various parts of the human body for final rendering and output.

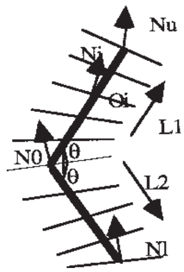
This approach takes advantage of the fixed topology of the human skeleton. Human limbs exhibit a cylindrical topology, and the underlying skeleton provides a natural centric axis upon which a number of cross-sections can be defined. Each limb link is associated with a number of contours (Fig. 14a). The cross-sectional skin contours can automatically be extracted with the ray-casting method. We cast rays in a star-shaped manner for one contour, with ray origins sitting on the skeleton link. For each ray, we compute the outermost intersection point with the muscles and bones surrounding the link. The intersection is a sample point on the cross-section contour (see Fig. 14b).



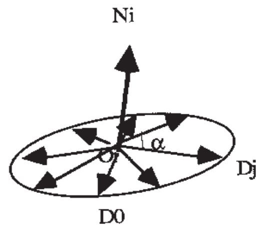
12



13




14a




14b

**Fig. 12.** Muscle compression sequence
**Fig. 13.** Bone motion with muscle contraction
**Fig. 14. a** Cross-sectional plane orientation; **b** ray distribution (Shen 1996)

To facilitate the manipulation, the body envelope was divided into seven parts (the so-called skin pieces), each defined around a junction between two or more links that contain a group of joints. These are: front torso, back torso, hip, left leg, right leg, left arm, right arm.
However, some postures could produce errors on the skin generation if we use the method just described. An example is when a leg or an arm of a character is bent. Considering the arm example, our algorithm would generate the skin covering (as a unique piece) at the same time as it generates the upper and lower part of the arm. Using the same principle, we could also have a single piece of skin covering the two legs of the character. To solve this problem, we have decided to assign each primitive (a bone or a muscle) to a group. Each group will blend with primitives in its own group or in the root group. Fifteen groups were defined to cover the whole body.
Each primitive is assigned to a group according to its attachment to the skeleton. Some primitives, located in special positions (for example, a muscle covering a joint), may belong to several groups simultaneously, as they may contribute to all of them. To sample a contour, we only need to consider the primitives in its associated group.
The skin generation method presented here is based on techniques developed by Shen and Thalmann (1995) and Shen (1996).

## 6 Integration framework

To integrate our joint-based skeleton, the bones, the muscles, and the skin, we have extended the *Body Builder* system (Shen 1996), originally designed for deformable human bodies with a stick skeleton, muscles and fat tissues represented by ellipsoids, and the skin. Our extended system, *Body Builder Plus*, allows the construction of an anatomically based model of humans, created entirely with bones and reconstructed muscles. A second objective is to integrate the modeled deformable muscles, described in Sect. 4, with ellipsoids to represent some muscles and other fat tissues. The idea, in this case, is to use physically based deformable muscles to simulate only the muscles that influence the human external appearance quite a lot.

### 6.1 Overview

The *Body Builder Plus* is an interactive human body modeling and deformation system developed on SGI workstations and based on a multilayered model. The system allows the design of 3D animated human figures in an interactive environment. Models may be animated by motion sequences input to the system or by changing a joint angle individually and interactively.

The model is divided into three layers: the skeleton (composed of a topological tree structure) and the bones, as explained in Sect. 3; the volumetric primitives, composed of ellipsoids and the muscles mentioned in Sect. 4; and the skin, represented by a B-spline surface covering all the body (see Sect. 5). The head, feet, and hands are modeled separately and attached to the body afterwards.

In *Body Builder Plus*, the volume primitives are the pieces used to represent muscles, fat, organs and so on. We have used two kind of primitives: the physically based muscles and the ellipsoids. Here we explain shortly the use of the ellipsoids included in the system by Shen (1996). They are divided into two categories: *blendable volume*, which blends with other blendable primitives in the same group, and *unblendable volume*, which does not blend with other primitives. For the sake of simplicity, we use only ellipsoids for *unblendable* primitives and isosurfaces with ellipsoidal density distribution for *blendable* primitives.

Each primitive can also be classified as *deformable* or not. Each deformable primitive is associated with a reference joint, whose value dynamically determines the center, orientation, and shape of that primitive. When the skeleton moves, all primitives attached to their relevant joints undergo the joint hierarchy transformations as rigid body motions. Deformable primitives also change their state. In fact, each DOF of a joint is defined by three angles: the minimum, the maximum, and the current angle (initially corresponding to the default posture). We define parameters for the primitives when the joint angle is in one of these three states, and afterwards, during the animation, these parameters are interpolated.

The body extremities (the head, the feet, the hands, and the penis) are not treated in detail in *Body Builder Plus*. In fact, they are modeled separately and saved in individual files (<filename>.sm) containing triangular-mesh surfaces. The system reads the sm files (Kalra 1997) and allows the user to position them interactively. The link with the body is made at the skin level.

### 6.2 Building a body

Due to the complexity of the system, the design of a new body in *Body Builder Plus* requires at least four steps. The first step is the definition of the skeleton characteristics. From a basic human topology, it is possible to edit the length of the limbs in order to create a number of different humans, with distinct sizes and proportions. Five normalized parameters are used to scale the standard skeleton template to accommodate variations in age, sex, and race (Boulic et al. 1994b). *Global scaling* consists of a uniform scaling along the $x$, $y$, $z$, axes to reach a given height. *Frontal scaling* is a specific scaling along one independent direction. *High lateral scaling* and *low lateral scaling* are proposed to differentiate the higher body from the lower body (useful for generating characterization). Finally, the *spine origin ratio* expresses the ratio of the spine origin height to the total height, while the same total height is maintained. Figure 15 furnishes an example of a skeleton instance definition, in a model file.

The second step comprises the addition of the bones. Bones are represented by triangle meshes and have been stored in individual files beforehand. They are inserted in *Body Builder Plus* by editing the model file and by adding the related data. As shown in the example of Fig. 16, each instance of a bone has a name, a joint name where the bone is attached, a topology file name, and an associated group name (the concept and use of groups is explained in Sect. 5).

The third step is the muscle creation. In our system, the user can interactively define the muscles action lines, saving the data in the model file. Muscle shapes with deformation parameters are defined in a structure called muscle model simulation (*mms*) data and interactively attached to the action line segments. The deformation parameters can be added by editing the model file or by reading it from a *mms* file.

```
inst skeleton 1 {
    high                  1749.790039 ,
    spine_ratio           1.000 ,
    high_lateral_scale    0.950 ,
    low_lateral_scale     1.000 ,
    frontal_ratio         1.000 ,
    gender                male ,
    template              "newton.tpl" ,
 } ;
```

**15**

```
inst bone 5 {
    name      "lhumerus" ,
    parent    joint l_shoulder_twisting ,
    file      "lhumerus.iv" ,
    group     l_arm_upper l_shoulder ,
 };

    inst bone 40 {
    name      "lfemur" ,
    parent    joint l_hip_twisting ,
    file      "lfemur.iv" ,
    group     l_leg_upper left_ankle ,
 };
```

**16**

```
inst model 297 {
    name     "right_triceps" ,
    type     muscle ,
    group    r_arm_upper r_arm_lower r_shoulder ,
    origin joint r_clav_abduct position -52.0 117.0 -34.0 keyblob 10,
    insertion joint r_shoulder_abduct position -5.0 13.0 -13.0 keyblob 5,
    tendon_origin 0.0 ,
    tendon_insertion 36.0 ,
    mms 1 ,
    rotation 36.0 ,

    origin joint r_clav_abduct position -52.0 117.0 -34.0 keyblob 10,
    insertion joint r_shoulder_abduct position -5.0 13.0 -13.0 keyblob 5,
    tendon_origin 42.0 ,
    tendon_insertion 158.0 ,
    mms 2 ,
    rotation 180.0 ,

    origin joint r_clav_abduct position -52.0 117.0 -34.0 keyblob 10,
    insertion joint r_shoulder_abduct position -5.0 13.0 -13.0 keyblob 5,
    tendon_origin 15.0 ,
    tendon_insertion 120.0 ,
    mms 3 ,
    rotation 57.0 ,
 } ;

 inst mms 1 {
    file     "triceps_Lportion.iv" ,
    degree_of_subdivision_on_x10 ,
    degree_of_subdivision_on_y5 ,
    elasticity_horizontal    0.0 ,
    elasticity_vertical    5.0 ,
    curvature_horizontal    0.0 ,
    curvature_vertical    5.0 ,
    total_mass    20.0 ,
    damping    0.5 ,
} ;
```

**17**

**Fig. 15.** Extract of a model file with the definition of the skeleton instance
**Fig. 16.** Model file extract with the definition of two bones
**Fig. 17.** Extract of a model file with the definition of a complex muscle

In Fig. 17, we can see an example of an instance model of type "muscle". Each muscle has a name, is associated with one or more groups, and is defined by one or more action lines. Each extremity of the action line is attached to a joint and has a position defined in the reference joint's local coordinate system. Each action line may or may not be related to a muscle shape by indicating this on *mms* index (−1 if no muscle is attached). Each *mms* instance contains the muscle-shape file name and the parameters required for the deformation. The tendon lengths are given for each action line, as well as a rotation angle used to adapt the muscle shape to the desired orientation over the skeleton.

The fourth step of the process is the skin generation, and it is done automatically by *Body Builder Plus*. We can also consider another intermediate step, if the user wishes to model a part of the body by using ellipsoids. This procedure is on the same level as the muscle definition step and could be done interactively.

## 6.3 The motion motor

Currently, there are two ways to perform human motion. The first way consists of picking a joint and interactively changing the value of this joint, while the second one is to read and execute the motion from an animation file. In both cases, the system detects the joint motion and enables the deformation of the volumetric primitives, the motion of bones, and the update of the action lines.

Each object in *Body Builder Plus* is represented globally and locally. For simplification purposes during the motion process, we consider the same global reference system for all the human figure and several local reference systems, one for each DOF of the body. Every object that is part of a human body is attached to a joint reference and is represented in its local reference system. Then, when this joint moves, the object moves at the same time, automatically. This is true for bones, for the extremities (head, feet, hands, and penis), and for non-deformable primitives. In the case of deformable primitives and physically based muscles, yet another procedure is necessary. Deformable primitives have their parameters interpolated in the local reference system, while the physically based muscle motion is guaranteed by updating the action lines.

The action line is the lowest level in our muscle simulation system. We first simulate the action line motion and then the muscle deformation. This way, the muscle compression or extension depends directly on the action line. However, the final deformation also depends on other physical parameters. In fact, action lines are attached to bones that move relative to joint movement. When a joint angle changes, bones move and, consequently, muscles attached to these bones also move in the same way.

The action line motion is produced by a module that, after each movement of the skeleton, detects which action lines are concerned and updates their attachments and control point positions. The module also calculates the new tendon locations over the action line. To accomplish this, we maintain a list of pointers to all the segments of action lines and a flag for each one indicating if it is in motion or not. If the action line changes during a joint motion, then the flag is set.

To enable the physically based muscle deformation, the system maintains a callback function that is called 20 times/s. This function verifies any motion of an action line or muscle. If one of the conditions is satisfied, the muscle deformation procedure starts. A muscle will be "static" again only when the motion of all the particles are considered insignificant (i.e., they are very small).

### 6.4 Results

In Fig. 18, we can see an example of a body composed of metaballs and a physically based deformable muscle (the red one) to simulate the *biceps brachii*. The body is shown in two postures: with a straight arm and a bent arm. Comparing the two, we can see that this specific muscle is contracted in the second posture and, consequently, swollen. The bodies on the right side of the figure are covered by the skin, allowing a better understanding of the deformation result.

Figure 19 shows the current state of our work in designing real muscles for the representation of a complete human body. Analyzing the representation of the muscles in the figure, we can compare, for instance, the stretch of the left *pectoralis major* and the shrink of the right *deltoideus* (at the shoulder) in the transition between the first and the second postures. Our current model is composed of 31 joints with 62 DOF, 73 bones, 33 muscles (represented by 105 fusiform muscle parts), and 186 action lines.

Figure 20 shows an example of elbow flexion performed by a body composed of bones and some muscles, in particular the *biceps brachii* and the *triceps brachii*.

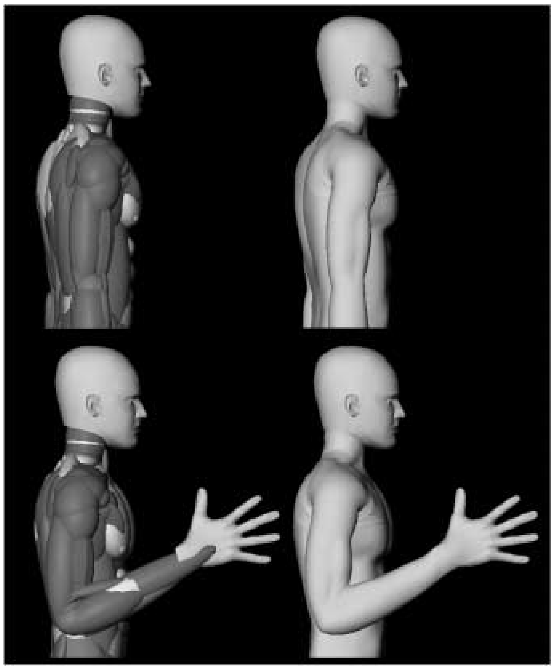
## 7 Conclusions and future work

To implement a human representation model, we have presented a multilayered approach containing the skeleton layer, intermediate layers to simulate the body volume (muscles, organs, fat, and so on), and the skin layer. In order to accomplish this task, each component of the model has been studied and designed separately.

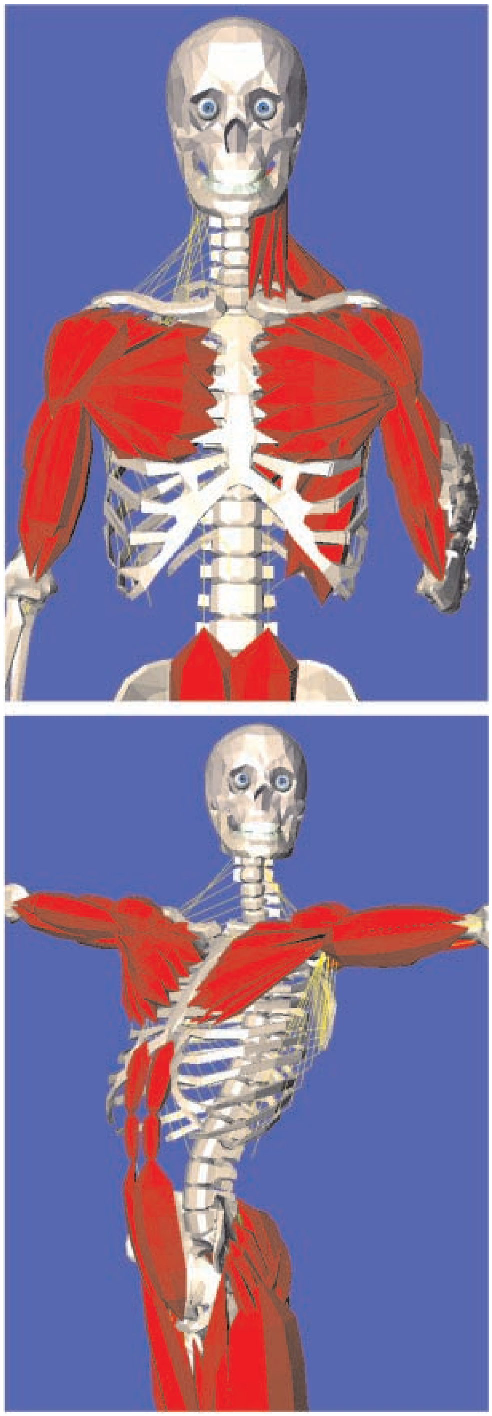
From a 3D skeleton and our studies on anatomy and some practical experiments, we have redefined all the joint positions and orientations. Concerning the visual aspect, we have added bone representation to the model. Our main contribution to this topic is the design of a new human template.

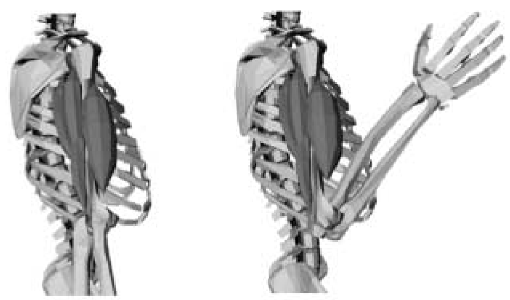
Considering the second layer of our model (volume primitives), we have concentrated our efforts on muscle representation, developing a technique to represent action lines. Tendons are defined over the action lines.

Similar techniques have been proposed in biomechanics research works, but they are not current in computer graphics applications. As a means towards



18





20

19

**Fig. 18.** Body design with metaballs and physically based deformable muscles
**Fig. 19.** Two muscled body postures
**Fig. 20.** Elbow flexion with muscles deformation

providing for reuse of sets of action lines, we have saved all the action lines in their normalized form. The normalization is a function of the lengths of the limbs.

We have chosen a physically based approach to simulate muscles in interactive applications because we believe we can obtain more fluid and realistic movements if we use the same principles as those used in nature. For the sake of performance, we apply a mass-spring model on the muscle external surface. The concept of angular springs was developed to control the volume of muscles during the deformation process, without the need of a post-processing step.

*Body Builder Plus* allows the construction of anatomically based humanoids created entirely with bones and physically based deformable muscles. However, another possibility is to mix the modeled deformable muscles with metaballs representing parts of the musculature and the fat tissues. The idea, in this case, is to use physically based muscles to simulate only the most important muscles in terms of the external appearance they produce.

In future work, we intend to:

- Include levels of detail in the skeleton definition allowing the choice of how many joints to represent according to the application
- Simulate other muscle forms
- Specify automatically the deformation parameters, as the elasticity and curvature coefficients
- Study the possibility of using multiresolution techniques during muscle deformation
- Improve the integration tool interface (*Body Builder Plus*).

*Acknowledgements.* We are grateful to Jianhua Shen for the development of the first version of *Body Builder* and the skin generation method, and to Ronan Boulic and Walter Maurel for useful discussions and help. We also thank Thierry Michellod for his help and motivation on the design aspects of muscles and Rafael Bordini for the English review. This work was supported by CAPES-Brazil and by the Swiss National Foundation for Scientific Research. Part of the research has also been developed in the framework of the BIOMED MIAS European Project, partly sponsored by the Swiss Federal Office for Science and Education.

## References

1. Badler N, Smoliar S (1979) Digital representations of human movement. ACM Computer Survey, vol 11
2. Badler N (1992) Graphical behavior and animated agents. In: Advanced Techniques in Human Modeling, Animation, and Rendering (Course Notes No. 17). ACM SIGGRAPH
3. Balcisoy S, Thalmann D (1997) Interaction between real and virtual humans in augmented reality. Proceedings of Computer Animation'97. IEEE Computer Society Press, Los Alamitos, pp 31–38
4. Becheiraz P, Thalmann D (1996) The use of nonverbal communication elements and dynamic interpersonal relationship for virtual actors. Proceedings of Computer Animation'96, IEEE Computer Society Press, pp 58–67
5. Boulic R, Renault O, (1991) 3D Hierarchies for animation. In: Magnenat-Thalmann N, Thalmann D (eds) New Trends in Animation and Visualization, John Wiley, Chichester
6. Boulic R, Huang Z, Magnenat-Thalmann N, Thalmann D (1994) Goal-oriented design and correction of articulated figure motion with the track system. Comput Graph 18:443–452
7. Boulic R, Çapin T, Huan Z, Molet T, Shen J, Magnenat-Thalmann N, Thalmann D (1994b) Human data structure & parallel integrated motion – specialized hierarchy for a general vertebrate body – User reference manual. Report of the ESPRIT Project 6709 – HUMANOID, EPFL, Lausanne, Switzerland
8. Certain A, Popovic J, DeRose T, Duchamp T, Salesin D, Stuetzle W (1996) Interactive multiresolution surface viewing. New Orleans Proceedings ACM SIGGRAPH '96, Addison-Wesley, pp 91–98
9. Chadwick JE, Haumann DR, Parent RE (1989) Layered construction for deformable animated characters. Comput Graph 23:243–252
10. Cody MDJ (1990) Visualizing muscles – a new ecorché approach to surface anatomy. University Press of Kansas
11. Forsey DR (1991) A surface model for skeleton-based character animation. In: The 2nd Eurographics Workshop on Animation and Simulation, Vienna, Austria, Springer, Wien, pp 55–73
12. Gascuel M-P, Verroust A, Puech C (1991) A modeling system for complex deformable bodies suited to animation and collision processing. J Visualization Comput Anim 2:82–91
13. Holton M, Alexander S (1995) Soft cellular modeling: a technique for the simulation of nonrigid materials. In: Earnshaw RA, Vince JA (eds) Computer graphics: developments in virtual environments. Academic Press, London, pp 449–460
14. Kalra P, Moccozet L, Werner HM, Boulic R, Huang Z, Shen J, Magnenat-Thalmann N, Thalmann D (1997) Interactive construction of 3D human shapes – Surface manager – User reference manual. Report for the ESPRIT Project 6709 – HUMANOID, University of Geneva, Geneva, Switzerland
15. Komatsu K (1988) Human skin model capable of natural shape variation. Visual Comput 3:265–271
16. Lindstrom P, Koller D, Ribarsky W, Hodges LF, Faust N, Turner GA (1996) Real-time, continuous level of detail rendering of height fields. Proceedings ACM SIGGRAPH '96, Addison-Wesley, pp 109–118
17. Magnenat-Thalmann N, Thalmann D (1987) The direction of synthetic actors in the film Rendez-vous à Montréal. IEEE Comput Graph Appl 7:9–19
18. Magnenat-Thalmann N, Thalmann D (1991) Complex models for animating synthetic actors. IEEE Comput Graph Appl 11:32–44
19. Monheit G, Badler N (1991) A kinematic model of the human spine and torso. IEEE Comput Graph Appl 11:29–38

20. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C – the art of scientific computing, 2nd Edn. Cambridge University Press
21. Scheepers F, Parent RE, Carlson WE, May SF (1997) Anatomy-based modeling of the human musculature. In: Proceedings ACM SIGGRAPH '97, Addison-Wesley, pp 163–172
22. Shen J, Thalmann D (1995) Interactive shape design using metaballs and splines. In: Proceedings of Eurographics Workshop on Implicit Surfaces'95, Grenoble, France, pp 187–196
23. Shen J (1996) Human body modeling and deformations. PhD Thesis. Computer Graphics Lab, École Polytechnique Fédérale de Lausanne
24. Singh K, Ohya J, Kishino F, Parent R (1995) Human figure synthesis and animation for virtual space teleconferencing. Virtual Reality Annual International Symposium'95, Research Triangle Park, NC, IEEE CS Press, Los Alamitos, pp 118–126
25. Terzopoulos D, Platt J, Barr A, Fleischer K (1987) Elastically deformable models. Comput Graph 21:205–214
26. Turner R (1995) LEMAN: a system for constructing and animating layered elastic characters. In: Earnshaw RA, Vince JA (eds) Computer graphics: developments in virtual environments. Academic Press, London, pp 185–203
27. Wilhelms J, Van Gelder A (1997) Anatomically based modeling. Proceedings ACM SIGGRAPH '97, Addison-Wesley, pp 173–180
28. Yoshimoto S (1992) Ballerinas generated by a personal computer. J Visualization Comput Anim 3:85–90

LUCIANA PORCHER NEDEL is currently a Postdoctoral Research Fellow at the Federal University of Rio Grande do Sul (UFRGS) in Porto Alegre, Brazil. She received her PhD in Computer Science from the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland, in 1998. Since 1991 she has been involved in computer animation research and has won four awards, including the first place in the Twelfth Thesis and Dissertation Contest, sponsored by the Brazilian Computing Society, and the Young Researcher Award bestowed at the Computer Graphics International Conference 1998 from the Computer Graphics Society. Her research interests include virtual humans, deformation and virtual reality. She is currently working on minimally invasive surgery simulation, human simulation applied to the medicine, and virtual reality.

DANIEL THALMANN is a pioneer in research on Virtual Humans. He is Coeditor-in-Chief of the *Journal of Visualization and Computer Animation*, a member of the Editorial Board of *The Visual Computer*, the *CADDM Journal* (China Engineering Society) and *Computer Graphics* (Russia). He is a cochair of the EUROGRAPHICS Working Group on Computer Simulation and Animation and a member of the Executive Board of the Computer Graphics Society. He is a Program Cochair of IEEE VR 2000. He has also organized 4 courses at SIGGRAPH on human animation. He has published more than 250 papers on graphics, animation, and virtual reality. He is a coeditor of 25 books, and a coauthor of the recent book Avatars in *Networked Virtual Environments*.