4. The Elastic Surface Layer Model

A model is described for creating three-dimensional animated characters. In this new type of layered construction technique, called the elastic surface layer model, a simulated elastically deformable skin surface is wrapped around a traditional kinematic articulated figure. Unlike previous layered models, the skin is free to slide along the underlying surface layers constrained by reaction forces which push the surface out and spring forces which pull the surface in to the underlying layers. By tuning the parameters of the physically-based model, a variety of surface shapes and behavior can be obtained such as more realistic-looking skin deformation at the joints, skin sliding over muscles, and dynamic effects such as squash-and-stretch and follow-through. Since the elastic model derives all of its input forces from the underlying articulated figure, the animator may specify all of the physical properties of the character once, during the initial character design process, after which a complete animation sequence can be created using a traditional skeleton animation technique.

4.1. Problems of 3D Character Animation

The elastic surface layer model was developed to provide a solution to some of the outstanding problems with current 3D character animation techniques. Several commercial software systems currently allow the creation and animation of articulated figures and some make use of layered construction techniques. However, at least two major problems still prevent 3D character animation from becoming versatile enough to be generally accepted by the commercial animation community.

4.1.1. Joint Animation

The first problem is that of natural-looking motion of the articulated skeleton. This is without doubt a difficult problem because the skeletal motion of a real human or animal is a result of both its passive physical properties and its active nervous system activity. As we have seen in previous chapters, much progress has been made in the development of kinematic techniques such as inverse kinematics and key-frame interpolation of joint angles [Girard 87], which provide good control at the expense of realistic-looking dynamics. Forward dynamic techniques [Armstrong 85], [Wilhelms 85], which can generate natural-looking motion, are nonetheless difficult to control. Space-time constraint techniques [Witken 88], [Isaacs 87] hold much promise to provide both control and natural dynamics simultaneously, but they are currently limited to fairly simple articulated figures. For some time to come, practical articulated skeleton animation systems will probably rely on a hybrid combination of these techniques. The elastic surface layer does not attempt to address this problem.

4.1.2. Skin Deformation

The second problem, which the elastic surface layer model does address, is that of deformation of the skin surface shape. As we have seen in Chapter 2, many geometric deformation techniques from the area of solids modeling are available, such as global deformation [Barr 84] and free-form deformations [Sederberg 86]. A variety of surface modeling methods have been proposed for representing deformable animated characters from standard polygonal surface meshes to implicit surfaces such as soft objects [Bloomenthal 90] and parametric surfaces such as hierarchical B-splines [Forsey 88]. These geometric techniques provide ease of control and rapid computation but they have little relation to the physical reality of a flesh and blood creature, and therefore tend to lack realism. In particular, they tend to represent characters either as geometric surfaces, or as uniform solids, both of which ignore the complex internal structure of human or animal anatomy. This problem can be attacked by using layered construction techniques, where an inner skeleton is used to deform the outer skin surface. In Chapter 3, we saw how layered construction can be enhanced by adding dynamic, physically-based components to the layered structure, resulting in what we refer to as *layered elastic models*.

4.1.3. Elastic Simulation

Physically-based deformable models [Terzopoulos 87], are based on the elastic and viscous properties of continuous media and therefore can produce very realistic looking simulations of deformable materials. Physically-based models, however, are usually difficult to control and therefore to be useful, elastic models must be constrained properly [Platt 88]. Because they represent continuous media as large numbers of discrete nodal elements, elastic models can also be very CPU-intensive, especially when simulating solids using three-dimensional lattices. However, elastic surfaces, simulated as two-dimensional lattices, require smaller numbers of discrete nodes to produce useful results and therefore are not as demanding of CPU time. It is now possible, using high-end workstations, to simulate a reasonably complex surface of a few hundred mass points in real-time.

4.1.4. Interactive Environment

The issue of speed is important because for 3D character animation to become practical, it must be possible to create and animate the characters interactively. However sophisticated the models become, whether animating "inanimate" objects, simple animals or realistic-looking human characters, character animation will always be an essentially creative process and it is necessary that the software tools be accessible to non-technical, creative animators. For these reasons, it is proposed that a model for three-dimensional character animation must be developed that provides a compromise between interactive speed and realism, and between control and physically realistic behavior.

4.1.5. Limitations

A limitation of most of geometric and elastic layered construction techniques is that they are essentially two-layered models, with an rigid skeleton and a deformable surface skin. However, much of what makes a character visually and dynamically interesting is what goes on between these two layers. In particular, muscle and fat are materials with very different behavior which each contribute to form the final exterior skin shape. Two-layer construction models cannot address this problem because they are too simple. More sophisticated models such as in the Critter system are limited to free-form deformations of geometrical shapes so that, even though the FFD control point mesh is based on a mass-spring network, the deformation is still fundamentally a geometrical one. The Terzopoulos facial animation model represents a step in the direction of multi-layered anatomical models, however it is very specialized, being derived from scanned data from real human faces. Furthermore, the three-layer mass-spring system used to model the flesh and skin is a descretization of a single solid material. Although this material can vary its properties from point to point, it does not always capture the complexity of real anatomy in which the different layers can have very different mechanical properties and can be loosely bound to each other. It is therefore not easy to reproduce such effects as skin sliding over underlying muscle surfaces. A better layered model might therefore try to represent these separate anatomical layers more carefully as well as the connections between them.

4.1.6. Construction and Animation

One of the advantages of using layered construction techniques is that the two problems of skeleton animation and skin deformation can usually be separated. This has benefits for both the animator and the researcher. From the animator's point of view, it allows a greater range of choices, since he can mix and match between modular skeleton and skin animation systems. For the researcher, it allows the 3D problem to be divided and the two problems tackled separately. Therefore, for the remainder of this Chapter, we will present a new approach to the skin deformation problem for animated characters, relying on standard techniques for animating the articulated skeleton.

4.2. Modeling Anatomy

Human and animal character animation requires a careful study of anatomy, and even the most stylized animated characters still have an underlying structure which contributes to their outer shape and dynamics. How does one begin to try to construct a computer model of human or animal anatomy? Obviously, the anatomical figure is immensely complex, but it is possible to break its important components down into several well-defined layers that contribute to the overall visual appearance and behavior. Going from the inside out, these layers can be defined as: *skeleton, bone, muscle, fat, and skin*. On top of the skin layer can be added *hair* (or *fur*) and *clothing*, if desired, although this will not be addressed in this paper. From the point of view of creating a computer model, each of these layers has distinct geometrical and dynamic properties which make it suitable for particular modeling techniques.

Bones, for example, are for all practical purposes rigid bodies, and their arrangement in a skeleton can be modeled very well as an articulated hierarchy of rigid bodies. Since this is a very well-developed area, the modeling of the bones themselves and their kinematic relationships is basically a solved problem. The motion of the skeleton, however, is normally under active control and therefore a difficult problem which falls under the category of articulated figure animation.

Muscles, which are attached to the bones, are highly deformable and furthermore the only structure under active control. Since muscles are volume conserving and seem to resemble in some ways simple deformable solids, it may seem that they could be modeled well using physically-based techniques. However, when muscles are being used, their motion and deformation is very much a result of active nervous system control, and therefore models of passive materials are probably not appropriate for modeling muscle shape. Furthermore, physically-based models of muscle normally would require a three-dimensional discretization which can be very computationally intensive. Since muscles are generally smooth surfaces without complex topology, and since their shape is somewhat obscured by surrounding layers, geometrical models of deformable surfaces, controlled by a few input parameters such as joint angle and tension, are probably the most useful models to use.

The fat and skin layers, by contrast, are completely passive structures and therefore more amenable to physically-based simulation. The skin layer is characterized by being relatively thin and is therefore a good candidate for simulation using an elastic surface. The fat layer separates the muscle layer from the skin. It is normally attached to the skin and can be defined by its thickness at each point on the skin. Fat has the general physical property that it is volume conserving.

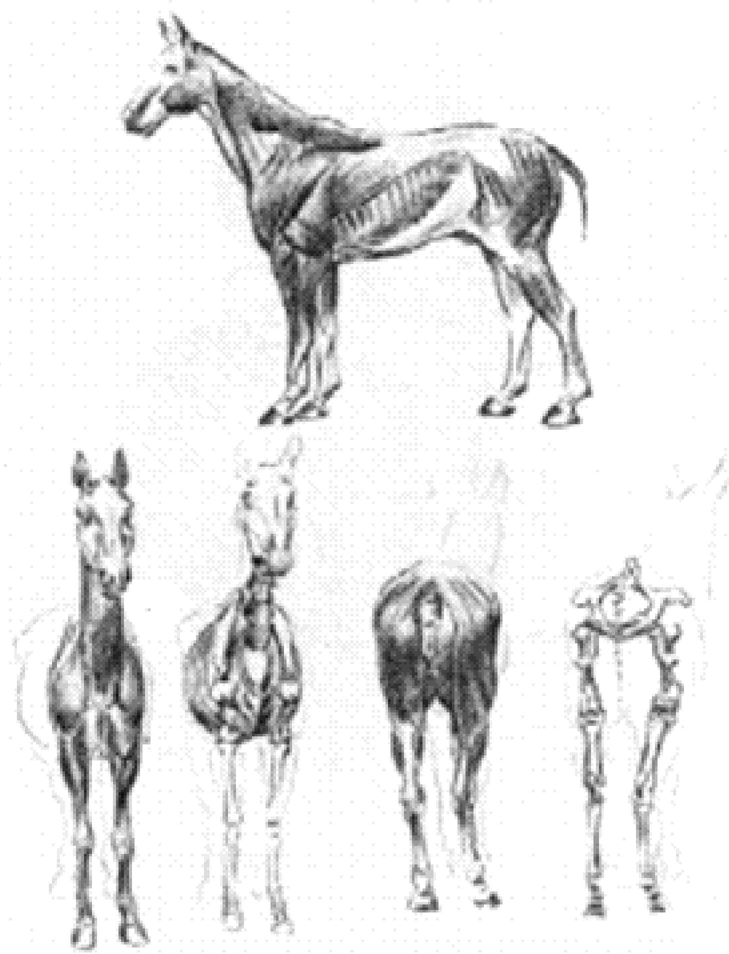

Figure 4.1: Anatomical Study by Shamus Culhane [Culhane 88]

4.3. A Character Animation Pipeline

An attractive aspect of this kind of layered breakdown is that, to a good approximation, each layer is dependent only on what is inside it, not the other way around. Therefore, it is possible to construct a character animation pipeline in which each stage of the pipeline adds another layer and can be modeled by a separate algorithm. For example, there exist a number of techniques for generating animated skeleton sequences. Output from an algorithm using any of these techniques can be used to derive the skeleton motion which can then drive the outer layers. This is not a perfect model, of course. In reality it is the muscles that drive the skeleton and not the other way around, skin and fat contribute mass which effects the dynamic motion of the joint, and collisions with other objects are transmitted back through the skin to the skeleton. Nonetheless, these situations can often be handled as special cases of feedback so that the pipeline model can make a good approximation in a variety of circumstances.

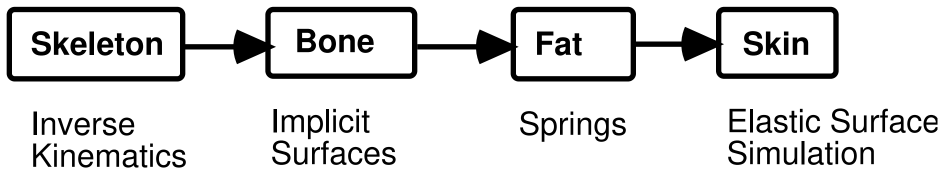

Figure 4.2: Example Of A Layered Character Animation Pipeline

Another interesting aspect of the layered breakdown is that (with the possible exception of the skeleton motion) each successive layer is more and more visible and therefore requires more physical realism. Therefore it makes sense to build more sophisticated physically-based models for the outer layers while the inner layers can be simpler kinematic or geometric models. In fact, the only layer we can actually see is the skin layer, so it makes sense that this should be the starting point for a physically-based simulation.

4.4. A Hybrid Model

The different characteristics of the different anatomical layers suggest a hybrid model in which different modeling techniques are used at each stage of the pipeline to form the final layered model. Each layer uses the type of modeling technique most suited to its characteristics. Figure 4.3 shows a diagram of the different components of the elastic surface layer model. We start with the outer skin layer which we model as an physically-based simulation of an elastic surface. We then work inward, considering each successive layer as a constraint acting on the layer outside it.

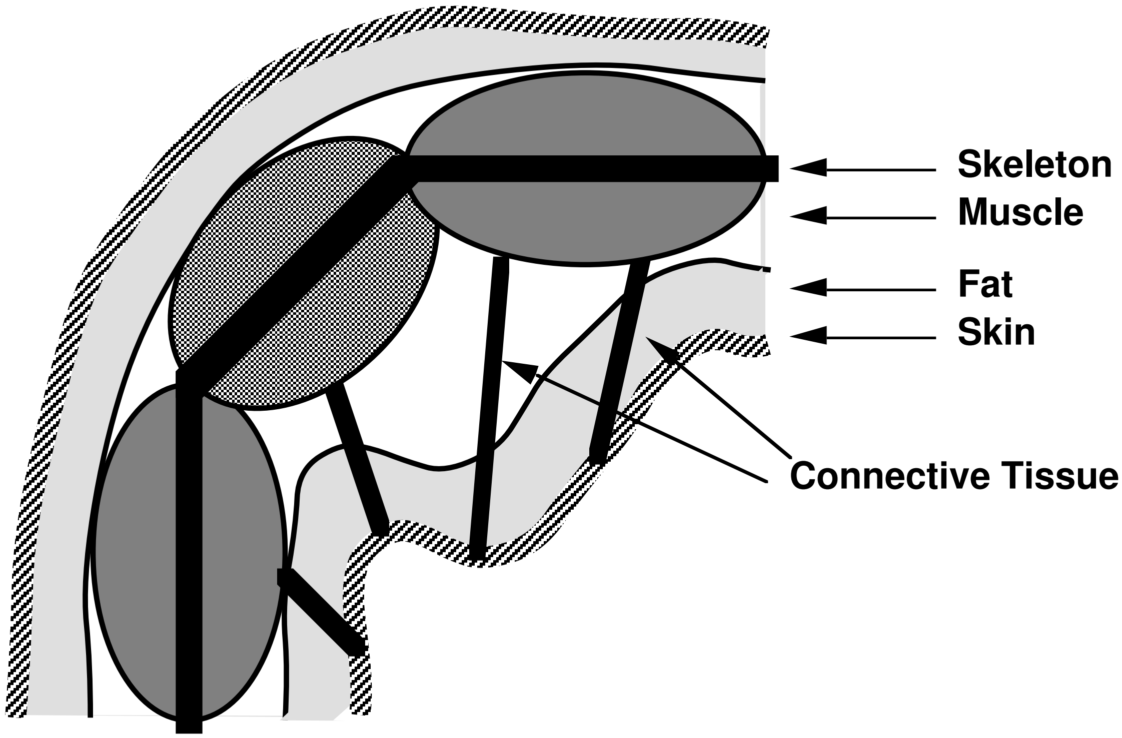

Figure 4.3: Components of the Elastic Surface Layer Model

4.5. The Elastic Surface Layer Model

We have developed the elastic surface layer model in an attempt to improve realism of the skin surface and to find a practical compromise between purely kinematic and purely dynamic layered modeling approaches. By modeling the skin as an independent elastic surface and using reaction constraints to push it outside the underlying layers, we can achieve more realistic skin behavior in a practical system for constructing and animating 3D characters.

4.5.1. Skin Layer

The skin layer is at the starting point for the elastic surface layer model and is the only layer that is purely physically based, using a simplified physical model of a continuous elastic surface. The surface is discretized using the finite difference technique and represented as a rectangular mesh of three-dimensional points, together with their physical characteristics (e.g. mass, elasticity) and their current state information (e.g. position, velocity). When the numerical solver is turned on, the state is evolved over time at a fixed simulation time step. [Terzopoulos 87].

The various physical parameters of the surface such as elasticity, rest metric, mass and damping, can all be specified at each point on the surface, allowing fine control of the intrinsic aspects of the surface behavior. Increasing the mass, for instance, increases the dynamic follow-through and squash-and-stretch effects of the skin, while increasing the damping density retards them. Globally adjusting the elasticity tensor affects the relative looseness or tightness of the skin, while selectively setting the elasticity tensor values at certain regions of the surface can simulate such effects such as wrinkles and tendons under the skin.

As the surface mesh evolves over time, external forces can be applied to each point. These forces constitute the constraints which bind it to the underlying surface layers. Two types of force constraint are used: reaction constraints, which force the surface to be on the outside of a given solid volume, and hookian spring constraints, which attract a given point on the surface to a particular point on the surface of the volume.

One of the advantages of using a rectangular mesh data structure to represent the skin is that it is fairly straightforward to increase or decrease the resolution of the mesh in response to the need for either fast interaction or high visual quality. All the current values of the mesh (e.g. position, velocity, elasticity, spring constants) are bilinearly interpolated to determine the values of the higher resolution points. In practice, we usually design our characters and animate them at a fairly low resolution (for interactive speed) and then increase the resolution to calculate a final sequence at simulation speed. The resulting sequence is stored as a large array of successive elastic surface meshes and can be played back at interactive rates and viewed from different angles to check the final animation. Then the entire sequence can be rendered off-line using a standard rendering package. As a final step, textures and colors can also be mapped onto the surface of the skin to simulate hair, fur, natural skin colors or clothing

4.5.2. Fat and Connective Tissue Layers

The fat and connective tissue layers both separate and attach the skin to the underlying muscle and bone layers. The repulsive component is the fat layer which is specified simply as a thickness between the skin and muscle layers and is implemented using reaction constraints to push the skin the required distance out from the underlying layers. This thickness can be specified at each point on the skin surface so that a considerable amount of sculpting of the final character's appearance can be performed simply by adjusting this parameter.

Skin does not does not simply hang off the body, but rather is attached with varying degrees of tightness to the muscle via connective tissue. The manner in which it is attached and the degree of tightness versus looseness of the attachment contributes as much to the appearance and behavior of the skin as any physical properties of the skin itself. For example, a very loose attachment will allow skin to hang down in folds under the influence of gravity, and the nature of the connective tissue has considerable influence on the manner in which skin wrinkles and folds around joints.

In the elastic surface layer model, the connective tissue can be thought of as rubber bands strung between points on the skin surface and on the muscle layer surface. They are implemented as hookian spring force constraints acting between the two points. The spring constants of the rubber bands can be varied individually as well as their damping coefficients, allowing the degree of looseness or tightness of the skin attachment to be controlled. Also, various amounts of squash-and-stretch and follow-through effects of the skin can be controlled this way. An important parameter of the connective tissue layer is the exact binding points between the skin surface and the muscle layer surface. We have developed some interactive techniques for specifying this mapping. In the simplest case, we place the skeleton in a neutral position with the simulation running, and then calculate the nearest perpendicular muscle layer point for each skin surface point.

4.5.3. Muscle Layer

The muscle layer of the elastic surface layer model actually represents all of the solid components of the anatomy underneath the fat layer. Normally, this is primarily composed of muscle tissue, but where it comes close to the skin surface, it can also represent bone or cartilage. Since these components are either effectively rigid bodies (bone) or surfaces whose shape is primarily determined by active forces (muscle), physically-based elastic models are not necessarily the most appropriate technique to represent them. Also, since the shape of the muscle layer is somewhat obscured by the overlying layers, the need for a physically accurate model is not as important as for the skin.

The muscle layer has therefore been represented by deformable geometrical solid surfaces which the skin may not penetrate. Reaction constraints are used to force the skin outside of the muscle layer surface, but leave it free to slide along the surface until it finds an energy minimum. Since, in the worst case, every point on the surface must be tested for penetration, it is important that the muscle geometric models allow for quick inside/outside tests. For this we use spheres and implicit surfaces such as super quadrics which have a simple inside/outside function, together with global deformation functions [Barr 84]. The muscle shapes are attached as links to the skeleton joints so that they move as rigid components of an articulated figure. The flexing and bending of muscles is simulated by animating the parameters of the global deformations which can be tied to the joint angle values.

4.5.4. Skeleton Layer

We use the term skeleton in the computer animation sense of the word: a stick figure representing the positions and orientations of the joints which make up the articulated figure. The skeleton can be animated using a variety of techniques, as discussed in Chapter 2. Fortunately, the problem of skeleton animation is largely orthogonal to the problem of deformation, given our pipeline model, and therefore we can treat the problems separately. For our purposes, we have chosen a simple key-frame interpolation method, together with an interactive inverse kinematic positioning technique [Girard 87]. This can easily be replaced with a more sophisticated method when desired.

To set the posture of the articulated figure, a portion of the skeleton forming a kinematic chain is selected by the user. Using interactive input devices, the end-effector of the chain is moved incrementally. This movement can be either a translation, a rotation or both. This differential vector is then multiplied by the inverse Jacobian of the kinematic chain to determine the corresponding differential joint angle values, which are added to the current joint angles [Klein 83]. This process is repeated for each event at interactive speeds so that the user has the impression of directly manipulating the skeleton by moving a particular joint.

Once a desired posture has been found, the user can store its joint angles as a key posture. A series of key postures can then be used to interpolate a smooth motion, using interpolating splines on the joint angles [Kochanek 84]. This motion sequence can be played back in real time, to check the animation, or in (usually non-real) simulated time to calculate an animation sequence at high surface resolution.

4.6. The Physical Simulation

4.6.1. Deformable Surface

An obvious choice of physically-based model for representing skin would be a two-dimensional mass-spring system. For several reasons, however, a continuous model was used. Since a primary goal of the model was to achieve more natural-looking skin surface, it was felt important to have a model that took into account the true, non-linear nature of large surface deformations. Although this can be done with mass-spring systems by choosing an appropriate spring energy function, a continuous model based on a variation principle is a more natural way to described such phenomena mathematically. It was also desired to achieve as much functional separation as possible between the physical skin model, its solution techniques, and the other components of the elastic layered model such as the constraint mechanism. With a continuous model, it is simpler conceptually to swap in and out different types of solution techniques, constraint methods, or to adaptively resample the discretization at different sizes.

The deformable surface is modeled as a three-dimensional function of a two-dimensional material coordinate system, referred to as the u-coordinate system. Using the energy-based Lagrangian form, we can write the equation of motion for an elastically deformable surface as:

$$\mu \frac{\partial^2 \vec{r}}{\partial t^2} + \gamma \frac{\partial \vec{r}}{\partial t} + \frac{\delta \mathcal{E}(\vec{r})}{\delta \vec{r}} = \vec{f} \tag{4.1}$$

where

$t =$ time

$\vec{r}(u_1, u_2, t) =$ surface position

$\vec{f}(u_1, u_2, t) =$ external applied force density

$\mu =$ surface mass density

$\gamma =$ surface damping density

$\mathcal{E}(\vec{r}) =$ total elastic energy

$\dfrac{\delta \mathcal{E}(\vec{r})}{\delta \vec{r}} =$ elastic force density

$\mathcal{E}(\vec{r})$,the total elastic energy of the surface, is a scalar-valued functional, that is, a single-valued function of the entire surface $\vec{r}(u_1, u_2, t)$. $\delta \mathcal{E}(\vec{r}) / \delta \vec{r}$ is its variational derivative and represents the elastic force at each point on the surface [Gelfand 63]. The fact that this local quantity depends on a functional of the entire surface is the reason this kind of elastic model is non-linear. Note that this is an equation in force per u-coordinate area, or force density.

The choice of functional, $\mathcal{E}(\vec{r})$, is determined by the elastic properties of the surface and choosing a good model of these properties can help to make the solution of the equations easier. According to the rules of differential geometry, the local geometric properties which define a surface are its 2x2 metric and curvature tensors. These are defined as:

$$\begin{aligned}
G_{ij} &= \frac{\partial \vec{r}}{\partial u_1} \cdot \frac{\partial \vec{r}}{\partial u_2} \\
B_{ij} &= \hat{n} \cdot \frac{\partial^2 \vec{r}}{\partial u_1 \partial u_2} \\
\hat{n} &= \frac{\vec{n}}{\|\vec{n}\|} \\
\vec{n} &= \frac{\partial \vec{r}}{\partial u_1} \times \frac{\partial \vec{r}}{\partial u_2}
\end{aligned} \tag{4.2}$$

where G_{ij} is the metric tensor, B_{ij} is the curvature tensor and $\hat{n}$ is the unit normal to the surface. It is therefore reasonable to model the deformation energy as a function of these local properties We use the elastic model proposed by [Terzopoulos 87] which defines the energy functional to be:

$$\mathcal{E}(\vec{r}) = \int_{\Omega} \sum_{i,j=1}^{2} \left(\eta_{ij} (G_{ij} - G_{ij}^0)^2 + \xi_{ij} (B_{ij} - B_{ij}^0)^2 \right) du_1 du_2 \tag{4.3}$$

where G_{ij} and G_{ij}^0 are the 2x2 symmetric current metric tensor and rest metric tensor, respectively, while B_{ij} and B_{ij}^0 are the current curvature and rest curvature tensors. The difference between the current and rest metric tensors is the metric strain tensor, which is the tensor analog of the displacement of a one-dimensional spring. The difference between the current and rest curvature tensors is the curvature strain. η_{ij} and ξ_{ij} are 2x2 symmetric "weighting functions", which determine the elasticity of the surface at each point, although they are not true elasticity tensors, which are rank four tensors. This places some limits on the range of elastic materials that can be represented.

The variational derivative of this expression can then be approximated by the equation:

$$\frac{\mathcal{E}(\vec{r})}{\delta \vec{r}} = \sum_{i,j=1}^{2} -\frac{\partial}{\partial u_i} \left(\eta_{ij} (G_{ij} - G_{ij}^0) \frac{\partial \vec{r}}{\partial u_j} \right) + \frac{\partial^2}{\partial u_i \partial u_j} \left(\xi_{ij} (B_{ij} - B_{ij}^0) \frac{\partial^2 \vec{r}}{\partial u_i \partial u_j} \right) \tag{4.4}$$

which is the elastic force in the force equation and is manageable enough to be solved numerically.

We now have a complete left-hand side of our force equation (4.1), which represents the behavior of an elastic surface over time. We can affect this behavior by adjusting various intrinsic properties associated with each point on the surface: the mass density, μ, which determines its inertia; the damping density, γ, which determines its viscous damping; the rest metric, G_{ij}^0 which determines its desired size; the stretching elasticity, η_{ij}, which determines its resistance to stretching; the rest curvature, B_{ij}^0, which determines its desired curvature; and the bending elasticity, ξ_{ij}, which determines its resistance to bending. Since skin does not have very much resistance to bending, and also to speed up the solution, we usually set ξ_{ij} to zero.

4.6.2. Environmental Forces

While the left-hand side of (4.1) represents the behavior of the passive skin layer of the elastic surface model, the right-hand side represents the active, driving input forces to the skin caused by the environment and the inner layers of the character model. The driving force term, $\check{f}(u_1,u_2,t)$ is therefore composed of the sum of all the external and constraint forces acting on the surface:

$$\check{f} = \check{f}_G + \check{f}_P + \check{f}_S + \check{f}_R$$

The first two of these, $\check{f}_G = \text{gravity}$ and $\check{f}_P = \text{air pressure}$, are environmental forces which act on all surface points. The force of gravity is simply:

$$\check{f}_G = \mu \check{g}$$

where g is the acceleration of gravity (in the appropriate direction). The internal air-pressure force is:

$$\check{f}_P = \frac{NRT}{V} a\hat{n}$$

where N is the amount of air in moles, R is the gas constant, V is the volume of the elastic surface, T is the temperature, and a is the ratio of surface area to coordinate area, and $\hat{n}$ is the unit vector pointing in the direction of the surface normal [Feynman 63].

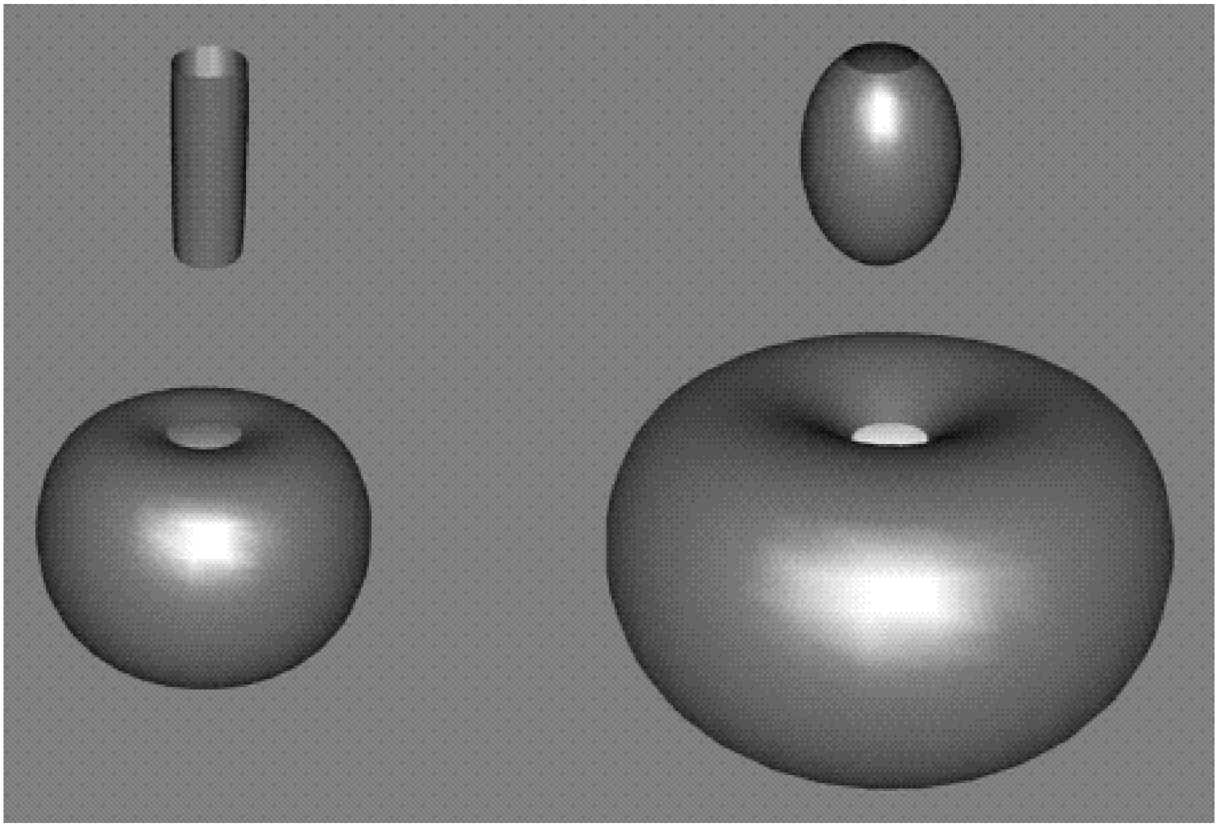

Figure 4.4: Air Pressure Force

4.6.3. Spring Constraint Forces

Each point on the elastic surface may be bound to a point on the underlying layers by a "rubber-band" constraint which exerts a spring force on the surface point. The rubber-band analogy is a good one because the force is attractive only and starts to act only beyond a given separation distance between the two points (the initial length of the rubber band). Beyond this separation distance, the force is hookian, i.e. proportional to the displacement. A spring damping term is also added to control oscillations caused by the spring forces. The force equation for the spring constraints can then be written as:

$$\vec{f}_S = -(k_S(\|\vec{x}\| - x_0) + k_D \vec{v})\frac{\vec{x}}{\|\vec{x}\|}$$

where $\vec{x}$ is the difference vector from the fixed point to the surface point, x_0 is the initial length of the spring, k_S is the spring constant and k_D is the damping coefficient.

The spring constraints simulate the connective tissue which binds skin to muscle. By tuning the parameters of the springs (k_S, k_D and x_0), the animator can fine-tune the behavior of the skin. High values of k_S, for example, result in skin that clings tightly to the skeleton, while low values result in loose, floppy skin that hangs down below the skeleton under the influence of gravity. Adjusting the values of k_D controls how fast the skin follow-through dies down.

4.6.4. Reaction Constraint Forces

The reaction constraint forces act to force the skin surface outside the bone and muscle layers, preventing them from penetrating these surfaces. Unlike the environmental and spring constraints, which merely add forces to the system, reaction constraints remove "undesirable" forces (i.e. forces that cause penetration) and replacing them with forces that drive the elastic surface towards the constrain surface with critically damped motion [Platt 88]. "Desirable" forces (i.e. those that do not cause penetration) are retained. Figure 4.5 shows the important vector quantities of an element of the skin surface which is inside the constraint surface,

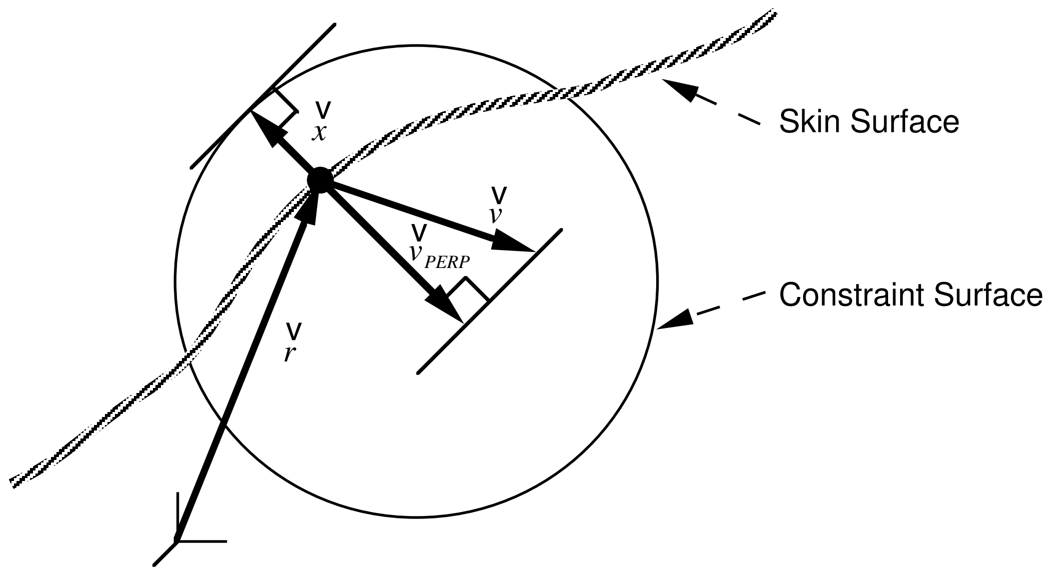

Figure 4.5: Element of Skin Inside Constraint Surface

where $\vec{r}$ is the position of the skin surface element, $\vec{v}$ is the velocity of the surface element, $\vec{v}_{PERP}$ is the component of the velocity perpendicular to the constraint surface, $\vec{x}$ is the perpendicular displacement vector to the constraint surface, $x = \|\vec{x}\|$ is the magnitude of $\vec{x}$, and $\hat{x} = \vec{x}/\|\vec{x}\|$ is the normal vector pointing in the direction of $\vec{x}$.

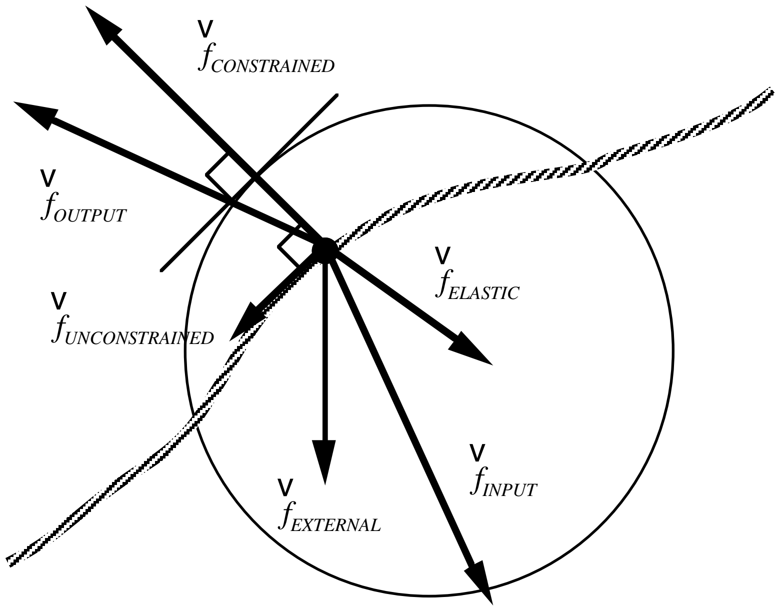

Figure 4.6: Reaction Constraint Force Components

Figure 4.6 shows the vector forces acting on the surface element which are used to calculate the reaction constraint force, where $\check{f}_{EXTERNAL} = \check{f}_G + \check{f}_P + \check{f}_S$ is the sum of the external driving forces, $\check{f}_{ELASTIC}$ is the sum of the internal elastic forces, $\check{f}_{INPUT} = \check{f}_{EXTERNAL} + \check{f}_{ELASTIC}$ is the input force to the reaction constraint, $\check{f}_{UNCONSTRAINED}$ is the unconstrained component of the input force, $\check{f}_{CONSTRAINED}$ is the calculated constrained component, $\check{f}_{OUTPUT} = \check{f}_{CONSTRAINED} + \check{f}_{UNCONSTRAINED}$ is the force output by the reaction constraint.

The reaction constraint calculation takes an input force, $\check{f}_{INPUT}$ which consists of all of the forces acting on the surface element, both external forces (including forces from other constraints) and internal elastic forces. It then projects out from this vector the component perpendicular to the constraint surface to yield the unconstrained component, $\check{f}_{UNCONSTRAINED}$.

$$\check{f}_{UNCONSTRAINED} = \check{f}_{INPUT} - (\check{f}_{INPUT} \bullet \hat{x})\hat{x}$$

The force necessary to drive the mass element toward the constraint surface with damped motion is then calculated. In the absence of any internal elastic forces, this force would be sufficient to make the mass element fit its constraint exactly. Since there are internal elastic forces on the left-hand side of the differential equation, however, we must counteract these on the right-hand side by subtracting out the component of the internal elastic force on the mass point, $\check{f}_{ELASTIC}$, which is perpendicular to the constraint surface, yielding the final constraint force, $\check{f}_{CONSTRAINED}$.

$$\check{f}_{CONSTRAINED} = \mu(\omega_0^2 \overset{\vee}{x} + \gamma_0 \overset{\vee}{v}_{PROJ}) - (\check{f}_{ELASTIC} \bullet \hat{x})\hat{x}$$

where ω_0 and γ_0 are the spring and damping angular frequencies (inverses of the time constants), respectively, of the reaction constraint. When $\gamma_0 = 2\omega_0$ the surface element will move towards the constraint surface with critically damped motion. The time constants are generally set as fast as possible without causing numerical instability and should bet set to cause constraint resolution in less than the frame time of the animation.

Therefore, reaction constraint forces are in reality functions of all other forces in the system, including the internal forces of the elastic model itself. This means that to calculate the reaction constraint term, $\check{f}_R$, in equation (4.1), we must know the values of all the other forces on both the left-hand and right-hand sides of the equation. While this can be done easily for the right-hand side of the equation, simply by calculating the reaction constraint forces after the environmental forces, this is more difficult for the left-hand side since it involves getting into the details of the elasticity model and therefore decreases the functional separation between the elastic surface model and the constraint forces.

We have therefore developed a numerical method for estimating the elastic forces on each surface point using the applied forces at the previous time step in the solution. If, during the course of the physical simulation, we apply a reaction constraint force to a surface point, it will start to move toward the constraint surface. At the next time step, we can estimate the elastic force exerted on the point by taking the previous time step's reaction constraint force and subtracting out the component which went into accelerating the point. This difference between the actual applied force and the observed acceleration force is assumed to be the negative of the elastic force on the point. This estimated elastic force is then used to calculate a new value for the reaction constraint force. Using this form of feedback, the reaction constraint forces quickly grow until they become equal to the elastic forces, balancing them exactly when the mass point comes to rest at the constraint surface.

Since the reaction constraint forces only act perpendicularly to the bone and muscle layer surfaces, the skin surface is free to slide over these inner layers, eventually finding its own energy minimum within the constraints imposed on it. This results in a more realistic-looking effect because, like real skin, the skin surface is not tightly bound to a specific joint on the skeleton as in other elastic layered models.

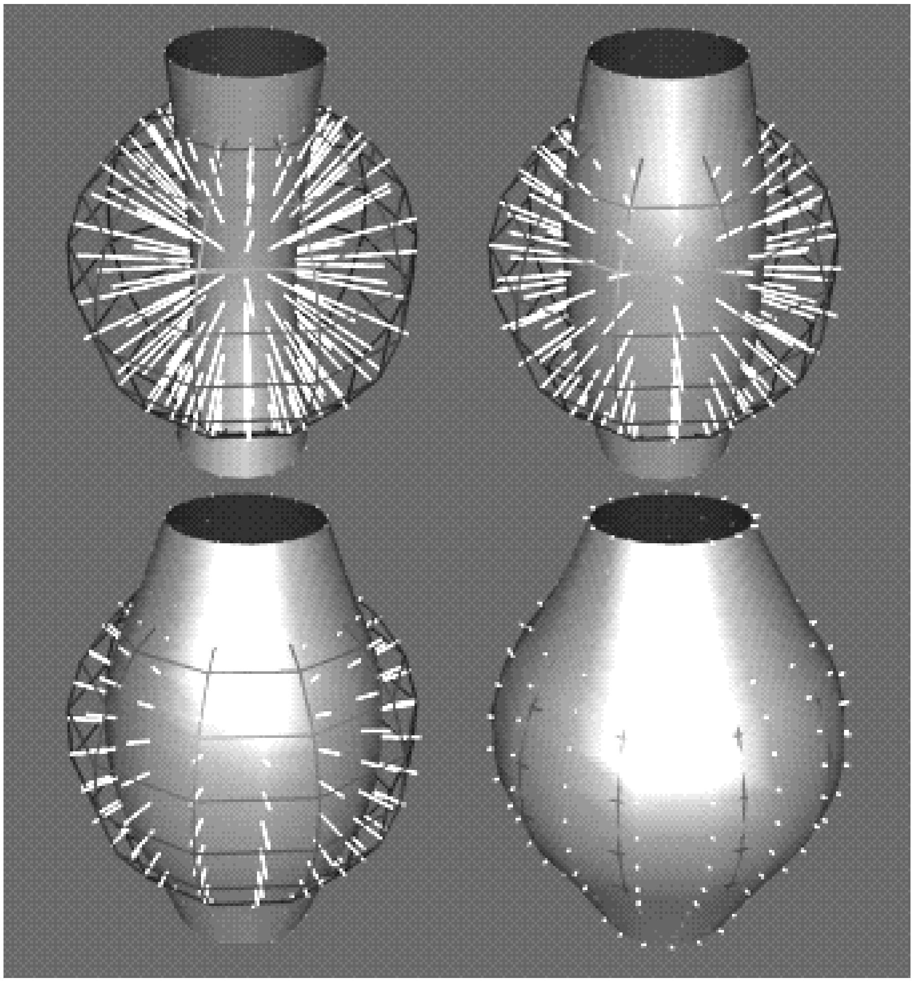

Figure 4.7: Demonstration of Reaction Constraints

4.6.5. Discretization And Solution

Although finite element methods are generally more powerful, finite difference methods require simpler data structures and numerical techniques, so following [Terzopoulos 87] we have chosen the finite difference technique. The surface is discretized as an MxN rectangular mesh of mass points at regular intervals in material coordinates. If we regard this entire mesh of points as a single MxNx3 dimensional vector, **r**, we can write the canonical equation of motion for the surface as:

$$\mathbf{M}\frac{\partial^2 \check{\mathbf{r}}}{\partial t^2} + \mathbf{C}\frac{\partial \check{\mathbf{r}}}{\partial t} + \mathbf{K}(\check{\mathbf{r}})\check{\mathbf{r}} = \check{\mathbf{f}} \qquad (4.5)$$

Where **M** is the diagonal inertia matrix, **C** is the diagonal damping matrix, **K** is the diagonal banded stiffness matrix which results from discretizing the elastic force equation (4.3), and **f** is the vector of driving forces.

To derive the **K** matrix, following the method of [Terzopoulos 87] we define metric and curvature "stresses" to be:

$$\alpha_{ij} = \eta_{ij}(G_{ij} - G^0{}_{ij})$$
$$\beta_{ij} = \xi_{ij}(B_{ij} - B^0{}_{ij})$$

respectively, and the partial difference operators for a function $\check{\mathbf{r}}(u_1, u_2)$

$$D_1^+\check{\mathbf{r}}^{m,n} = (\check{\mathbf{r}}^{m+1,n} - \check{\mathbf{r}}^{m,n}) / h_1$$
$$D_2^+\check{\mathbf{r}}^{m,n} = (\check{\mathbf{r}}^{m,n+1} - \check{\mathbf{r}}^{m,n}) / h_2$$
$$D_1^-\check{\mathbf{r}}^{m,n} = (\check{\mathbf{r}}^{m,n} - \check{\mathbf{r}}^{m-1,n}) / h_1$$
$$D_2^-\check{\mathbf{r}}^{m,n} = (\check{\mathbf{r}}^{m,n} - \check{\mathbf{r}}^{m,n-1}) / h_2$$

where h_1 and h_2 are the grid spacings in the two local coordinate dimensions. The elastic force in equation (4.3) can then be expressed in difference form (for metric stress only) as:

$$\frac{\varepsilon(\check{r})}{\delta \check{r}} = \check{\mathbf{e}}^{m,n} = \sum_{i,j=1}^{2} -D_i^-\left(\alpha_{ij}{}^{m,n} D_j^+ \check{\mathbf{r}}^{m,n}\right)$$

This can be represented using a partial difference stencil operator:

$$\check{\mathbf{e}}^{m,n} = \begin{Bmatrix} A_{12}{}^{m-1,n} & \left(-A_{12}{}^{m,n} - A_{22}{}^{m,n}\right) & 0 \\ \left(-A_{11}{}^{m-1,n} - A_{12}{}^{m-1,n}\right) & \left(A^{m,n} - A_{11}{}^{m-1,n} + A_{22}{}^{m,n-1}\right) & \left(-A_{11}{}^{m,n} - A_{21}{}^{m,n}\right) \\ 0 & \left(-A_{21}{}^{m,n} - A_{22}{}^{m,n}\right) & A_{21}{}^{m,n-1} \end{Bmatrix} \check{\mathbf{r}}^{m,n}$$

where

$$A_{ij}{}^{m,n} = \frac{\alpha_{ij}{}^{m,n}}{h_i h_j}$$
$$A^{m,n} = \sum_{i,j=1}^{2} \frac{\alpha_{ij}{}^{m,n}}{h_i h_j}$$

The positions of the mesh points can then be evolved through time from a given set of initial conditions by a semi-implicit integration procedure to solve a series of boundary value problems, using the discrete time approximations:

$$\frac{\partial^2 \check{\mathbf{r}}}{\partial t^2} = (\check{\mathbf{r}}_{t+\Delta t} - 2\check{\mathbf{r}}_t + \check{\mathbf{r}}_{t-\Delta t}) / \Delta t^2$$
$$\frac{\partial \check{\mathbf{r}}}{\partial t} = (\check{\mathbf{r}}_{t+\Delta t} - \check{\mathbf{r}}_{t-\Delta t}) / 2\Delta t$$
$$\check{\mathbf{v}}_t = (\check{\mathbf{r}}_t - \check{\mathbf{r}}_{t-\Delta t}) / \Delta t$$

These can be substituted into the matrix equation of motion to form a final semi-implicit time-stepping matrix equation:

$$\mathbf{A}_t \check{\mathbf{r}}_{t+\Delta t} = \check{\mathbf{g}}_t$$

where

$$\mathbf{A}_t = \mathbf{K}(\check{\mathbf{r}}_t) + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C} \right)$$

and

$$\check{\mathbf{g}}_t = \check{\mathbf{f}}_t + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C} \right)\check{\mathbf{r}}_t + \left(\frac{1}{\Delta t}\mathbf{M} + \frac{1}{2}\mathbf{C} \right)\check{\mathbf{v}}_t$$

See [Terzopoulos 87] for a more detailed description of the discretization process.

4.7. Results and Analysis

The elastic surface layer model for character animation has been implemented as part of the LEMAN (Layered Elastic Model ANimation) system, an animation system developed for studying layered elastic models at the Swiss Federal Institute of Technology. LEMAN is written in C using an object-oriented style and is built on top of the Fifth Dimension Toolkit [Turner 90] which runs on Silicon Graphics Iris workstations. The LEMAN system allows elastic surface layer model characters to be constructed and animated in a totally interactive, direct manipulation environment, using various multi-degree-of-freedom input devices such as the spaceball, dataglove and MIDI keyboard. The details of character construction and animation using LEMAN are given in Chapter 5.

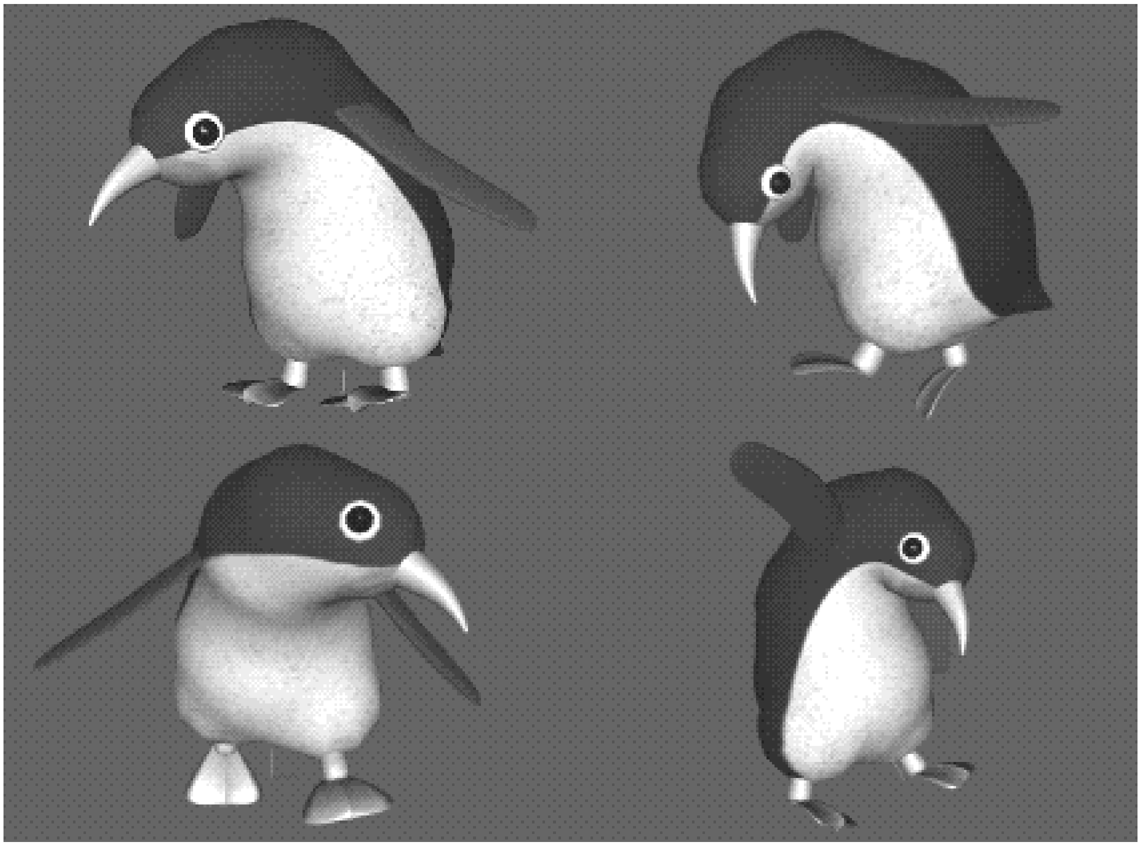

Figure 4.8: Character Deformation Under Various Skeleton Postures

4.7.1. Deformation at Joints

The first and most obvious result of using the elastic surface layer model is that it presents a solution, within certain limits, to the problem of deformation at joints. This in itself is a useful contribution because the generation of reasonable looking joint deformations is far from a solved problem. Most existing techniques attempt to interpolate either a digitized polygonal surface or a set of control points or, in the case of soft objects, equipotential surfaces. By shrinking an elastic membrane around an underlying surface shape, however, we achieve a surface interpolated between solid shapes, which much more closely approximates the real anatomical situation. Furthermore, the nature of the ESL model allows this interpolation to be controlled by intuitive physical parameters such as elasticity, and mass of the skin surface, the strength, damping and spatial arrangement of the connective tissue, and the thickness of the fat. It is therefore possible for a creative animator to specify the shape and nature of joint deformation in terms such as: "make it fatter in this area", "let the skin fall loose around here", and "make the skin cling tightly at this point".

An important issue in the subject of deformation at joints is how well it allows wrinkling and folding of the skin. In real life this effect is a complex result of the small inherent stiffness in the skin, the nature of the underlying fat and connective tissue, and collision of the skin surface with itself. Self-collision is an important contribution for large-angled joint deformations, but for smaller ones, it does not play a role. Since there is no self-collision detection of the skin surface in the current ESL implementation, the model does not work when the skin surface starts to self-intersect. Normally the skin passes right through itself but, in fact, it can sometimes become unstable because the simulation tries to resolve two conflicting reaction constraints, which is not possible. Modification of the reaction constraint model would be one way to deal with this, perhaps by geometrically averaging an intermediate constraint surface

Since for performance reasons we normally do not use the skin stiffness term in our simulation, skin wrinkling must be implemented by locally controlling the fat and connective tissue layers. There are several ways of doing this. Simply varying the fat thickness across the skin surface in a periodic way creates a limited kind of wrinkling. Augmenting this by strengthening the connective tissue bindings at wrinkle points allows the wrinkling amount to vary automatically as the joint deforms.

4.7.2. Dynamic Effects

The second clear advantage of using the ESL model is the ability to simulate certain kinds of dynamic effects. Dynamic effects in traditional animation usually fall under the terms of *squash-and-stretch*, *anticipation* and *follow-through*. Anticipation implies a certain active control by the character in preparation for acceleration, but both squash-and-stretch and follow-through can be described as passive dynamic deformations resulting from deceleration. In a real vertebrate animal, this deformation takes two forms: bending of the skeleton at the joints under the elastic restoring forces of the muscles, and elastic deformation of the outer layers around the skeleton. The effectiveness of skeletal squash-and-stretch (done kinematically) was demonstrated nicely in the film Luxo Jr. and Witken demonstrated that comparable effects can be achieved automatically using dynamics with space-time constraints [Witken 88].

Dynamic follow-through effects of the skin surface are a natural consequence of the physically-based simulation of the elastic surface layer model. The only modeling difficulties are making sure the dynamics stays within the limits of stability on the one hand, and keeping the time constants of the reaction constraints significantly smaller than the time constants of the skin. If this latter requirement is not maintained, the skin will have a tendency to lag behind the underlying layers during rapid motions. This can be controlled quite well by tuning the skin mass and elasticity parameters, and the connective tissue spring constants, although there can be some discrepancy in the behavior between low and high resolution if the time constants are not adjusted accordingly.

Squash-and-stretch, on the other hand, is generally associated with volume conservation: as the character compresses in the direction of deceleration, it expands in the perpendicular direction. This kind of effect can be simulated in the elastic surface layer model by maintaining a high enough air pressure inside the skin surface to cause the surface to bulge out against the connective tissue constraints when it is compressed, although not in a truly volume conserving way since we are talking about air pressure, not water pressure. True volume-conserving constraints could be added in fairly easily. This may not always have the effect desired by the animator, however, since often squash-and-stretch is a more stylistic than physical process. An additional postprocessing step might therefore be useful for adding in a component of non-physically-based squash-and-stretch using some kind of global deformation technique.

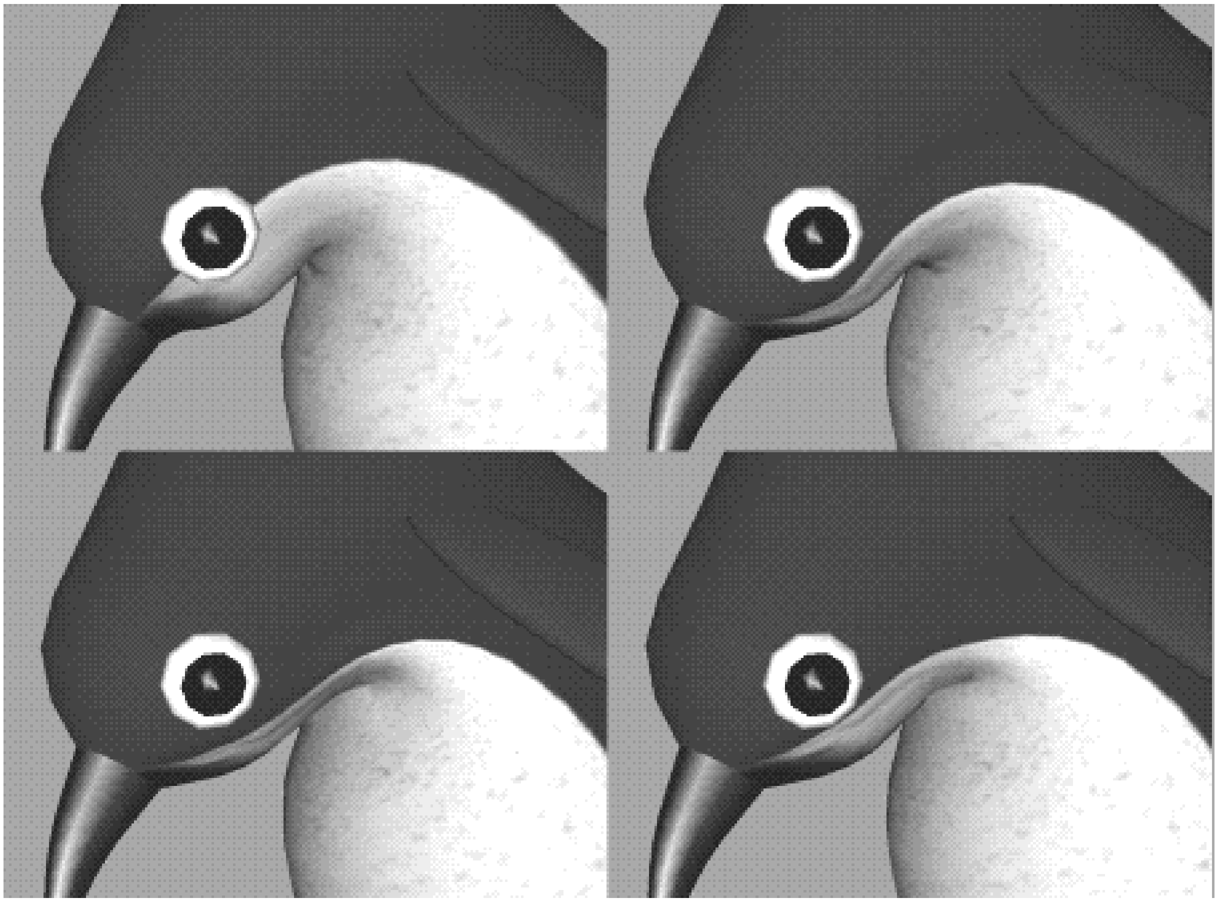

Figure 4.9: Sequence Showing Dynamic Skin Follow-Through