The Elastic Surface Layer Model for Animated Character Construction

Russell Turner, Daniel Thalmann

ABSTRACT

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 behaviors 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. A reasonably complex character at low surface resolution can be simulated at interactive speeds so than an animator can both design the character and animate it in a completely interactive, direct-manipulation environment. Once a motion sequence has been specified, the entire simulation can be recalculated at a higher surface resolution for better visual results. An implementation on a Silicon Graphics Iris workstation is described.

Keywords: Character Animation, Physically-Based Models, Deformation, Elasticity, Articulated Figures.

1. INTRODUCTION

Computer generated character animation remains an open research subject. While many other aspects of commercial animation have developed well-established computerized techniques, character animation is still, for the most part, the domain of the traditional animator drawing by hand. In the two-dimensional realm, computerized ink and paint, morphing and image processing have become standard tools. In the three-dimensional area, computer animation of backgrounds and rigid bodies are common techniques. Several commercial software systems now allow the creation and animation of articulated figures, however, at least two major problems still prevent 3D character animation from becoming versatile enough to be generally accepted by the commercial animation community.

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. 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. Spacetime 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. We will not be addressing this problem in this paper.

The second problem is that of deformation of the skin surface shape. 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. 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 fewer 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.

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 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, we believe 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.

Fortunately, the two problems of skeleton animation and skin deformation can usually be separated, and for the remainder of this paper, we will present a new approach to the skin deformation problem for animated characters, relying on standard techniques for animating the articulated skeleton. This approach, called the elastic surface layer model, falls into a general class of what we call *elastic layered models* of animated character construction. In section 2 we review previous work in layered construction models and layered elastic models. In section 3 we describe the elastic surface layer model we have developed. Section 4 gives mathematical details of the physical simulation and force constraints. Section 5 discusses our implementation using the LEMAN system, and Section 5 presents our conclusions.

2. THE LAYERED APPROACH

For designing animated characters, a common approach taken by artists and traditional animators is to work in layers. First a stick figure is drawn, representing the skeleton, followed by rounded forms to represent the flesh, followed by the finished outline, representing the skin [Culhane 88]. This same sort of approach is taken in clay animation, where plasticene is wrapped around a metal armature.

2.1 Layered Models

It is therefore not surprising that the first computer animated characters should also be constructed in layers. Magnenat-Thalmann and Thalmann used a two-layered approach to construct human characters in the film "Rendez-vous à Montréal," [Magnenat-Thalmann 87], in which a digitized outer enveloped was deformed by an underlying skeleton using abstract muscle procedures. The film "Tony De Peltrie" used combinations of digitized facial expressions to deform a polygonal surface. Implicit surfaces, called soft objects or blobbies surrounding a stick figure skeleton have been used to create deformable characters [Bloomenthal 90]. Forsey used hierarchical B-splines with control points attached to a skeleton for modeling animals and human joints [Forsey 91]

One of the major advantages of layered computer models is that it allows the animation process to be divided into two stages: character construction, in which the behavior of the layers and attachment to the skeleton is defined, and character animation, in which only the skeleton motion is specified. The outer layers then derive all their input from the skeleton motion alone, greatly simplifying the animation process. One basic limitation with all of these character models is the absence of any physical basis for the model. Both the skeleton and the surface envelope are purely geometric models. Furthermore, outer layers are usually tightly bound to the underlying skeleton, preventing the skin from sliding along the underlying layers.

2.2 Layered Elastic Models

Layered elastic models add physically based elastic components to some or all of the layers to improve realism. A simple example of this type of approach is Pacific Data Images *Goop* system, in which a mass and spring with damping are attached to each vertex of a polygonal model [Walters 89]. Moving the model causes the vertex points to oscillate, causing a jello-like effect, however, the surface points are not attached to each other, so the skin has no surface-like physical behavior.

A more sophisticated examples of layered elastic construction for animated characters is found in the Critter system [Chadwick 89] in which a network of connected springs and masses is used to create a control point lattice for free-form deformations of the geometric surface. Some of the control points are bound to links of the underlying skeleton so that, when the skeleton is animated, the unattached mass points are influenced to move through the spring lattice. In this way, a physical simulation controls a solid deformation. Although the mass-spring lattice allows for shape control over the muscle deformation and a technique for bending at the joints, the skin is still fundamentally a geometric surface model, not a model of a physical skin.

A model for articulated figures is proposed by Gascuel et al [Gascuel 91] in which the control points for an interpolating spline surface are bound to a rigid bone layer by springs. The finite element method is used by Gourret et al [Gourret 89], who describe a human hand modeled as a volume element mesh surrounding bones, and Chen et al [Chen 92], who have developed a biomechanically-based model of muscles on bone.

A sophisticated example of a layered elastic model is used by Terzopoulos to implement facial animation [Terzopoulos 91]. In this model, an elastic solid simulation, consisting of a mass-spring lattice of depth three, is attached to a human skull model and deformed by muscles which take the form of force constraints between the skin surface and the underlying bone. The springs are biphasic to emulate the non-linear behavior of real skin, and volume preserving constraints simulate the effects of incompressible fatty tissue.

A limitation of most of these techniques is that they model the layers as components of a single deformable solid material. Although this material can vary its properties from point to point, his 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.

3. 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. To explain how we do this, first let us step back and reexamine the fundamental layered character animation problem: modeling anatomy.

3.1 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. Muscles are highly deformable and furthermore the only structure under active control, so physically-based models of passive materials are probably not appropriate for modeling muscle shape. Geometrical models of deformable surfaces, with a few input parameters such as joint angle and tension, are probably the most useful in this case. 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 and can be defined by its thickness at each point on the skin.

3.2 The 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 under a variety of circumstances.

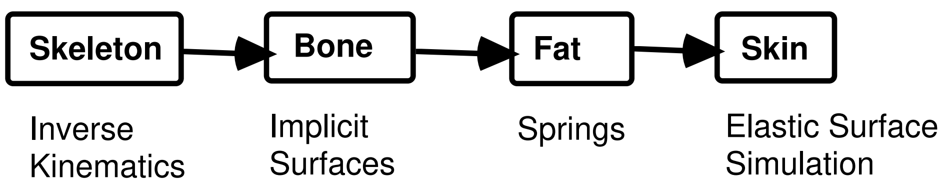

Figure 1: Example of a layered character animation pipeline

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.

3.3 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 2 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.

3.4 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].

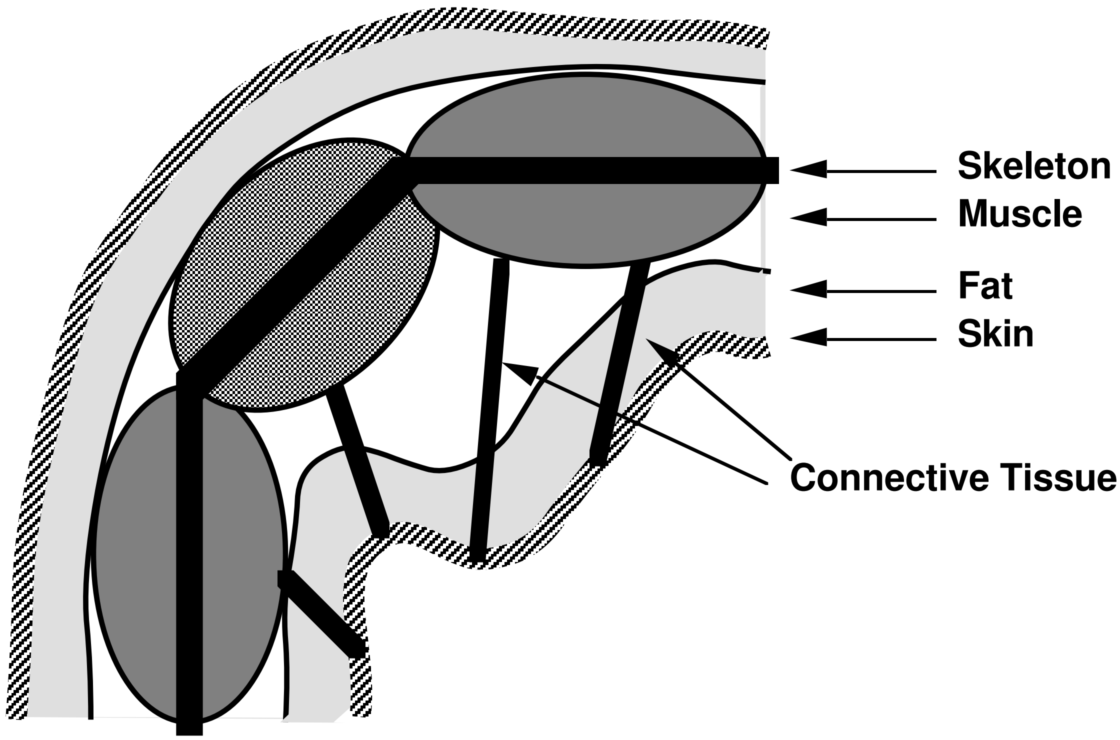

Figure 2: Components of the Elastic Surface Layer Model

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 constraints 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

3.5 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.

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.

3.6 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.

We have therefore chosen to represent the muscle layer 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, either directly using key-frame interpolation, or by tying them to the joint angle values of the joints.

3.7 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 section 1. 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. THE PHYSICAL SIMULATION

4.1 Deformable Surface

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 Lagrangian form, we can write the equation of motion for an elastically deformable surface as:

$$\mu \frac{\partial^2 \vec{r}}{\partial t^2} + \gamma \frac{\partial \vec{r}}{\partial t} + \frac{\delta \mathcal{E}(\vec{r})}{\delta \vec{r}} = \vec{f} \tag{1}$$

where t is time, $\vec{r}(u_1, u_2, t)$ is the surface position, $\vec{f}(u_1, u_2, t)$ is the external applied force density, μ is the surface mass density, γ is the surface damping density, $\mathcal{E}(\vec{r})$ is the total elastic energy, and $\mathcal{E}(\vec{r})/\delta \vec{r}$ is the 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. We use the elastic model proposed by [Terzopoulos 87] which defines the energy functional to be:

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

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:

$$\frac{\mathcal{E}(\vec{r})}{\delta \vec{r}} = \sum_{i,j=1}^{2} -\frac{\partial}{\partial u_i} \left(\eta_{ij} (G_{ij} - G_{ij}^0) \frac{\partial \vec{r}}{\partial u_j} \right) + \frac{\partial^2}{\partial u_i \partial u_j} \left(\xi_{ij} (B_{ij} - B_{ij}^0) \frac{\partial^2 \vec{r}}{\partial u_i \partial u_j} \right) \tag{3}$$

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 (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.

4.2 Environmental Forces

While the left-hand side of equation (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, $\vec{f}(u_1,u_2,t)$ is therefore composed of the sum of all the external and constraint forces acting on the surface:

$$\vec{f} = \vec{f}_G + \vec{f}_P + \vec{f}_S + \vec{f}_R \tag{4}$$

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

$$\vec{f}_G = \mu\vec{g} \tag{5}$$

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

$$\vec{f}_P = \frac{NRT}{V}a\hat{n} \tag{6}$$

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 65].

4.3 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:

$$\vec{f}_S = -(k_S(\|\vec{x}\| - x_0) + k_D\vec{v})\frac{\vec{x}}{\|\vec{x}\|} \tag{7}$$

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.

4.4 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 3a shows the important vector quantities of an element of the skin surface which is inside the constraint surface,

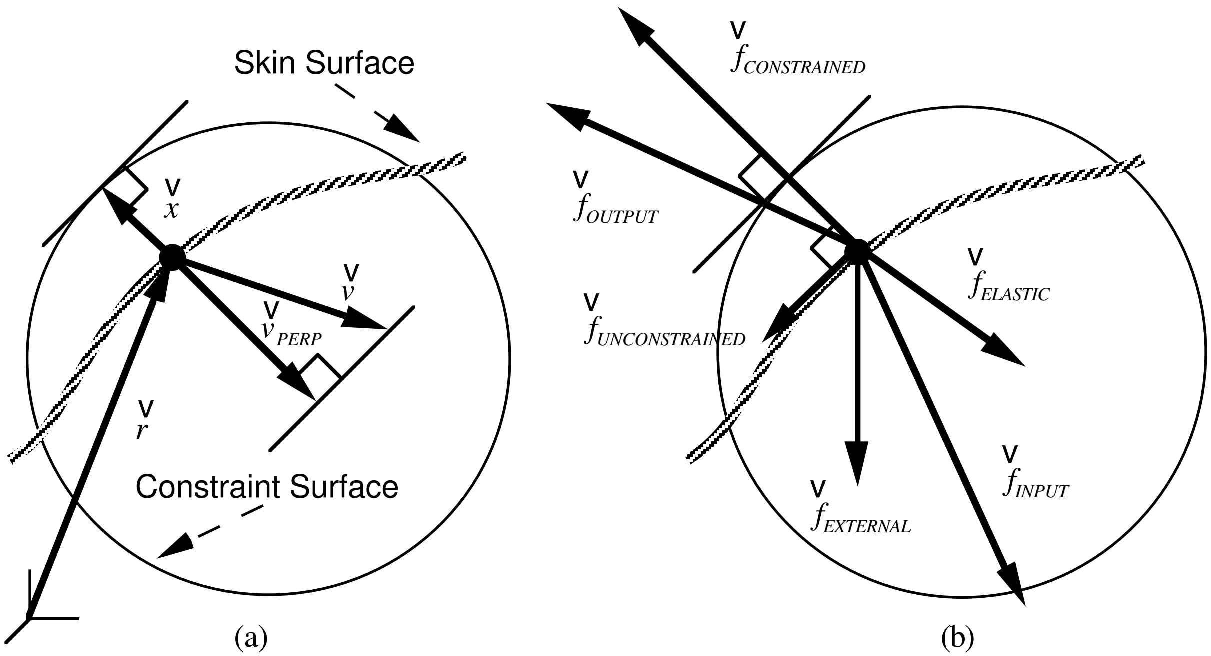

Figure 3: Reaction Constraint Force Components

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 3b shows the vector forces acting on the surface element which are used to calculate the reaction constraint force, where $\vec{f}_{EXTERNAL} = \vec{f}_G + \vec{f}_P + \vec{f}_S$ is the sum of the external driving forces, $\vec{f}_{ELASTIC}$ is the sum of the internal elastic forces, $\vec{f}_{INPUT} = \vec{f}_{EXTERNAL} + \vec{f}_{ELASTIC}$ is the input force to the reaction constraint, $\vec{f}_{UNCONSTRAINED}$ is the unconstrained component of the input force, $\vec{f}_{CONSTRAINED}$ is the calculated constrained component, $\vec{f}_{OUTPUT} = \vec{f}_{CONSTRAINED} + \vec{f}_{UNCONSTRAINED}$ is the force output by the reaction constraint.

The reaction constraint calculation takes an input force, $\vec{f}_{INPUT}$ 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, $\vec{f}_{UNCONSTRAINED}$.

$$\vec{f}_{UNCONSTRAINED} = \vec{f}_{INPUT} - (\vec{f}_{INPUT} \bullet \hat{x})\hat{x} \tag{8}$$

The force necessary to drive the mass element toward the constraint surface with damped motion is then calculated. In the absense 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, $\vec{f}_{ELASTIC}$, which is perpendicular to the constraint surface, yielding the final constraint force, $\vec{f}_{CONSTRAINED}$.

$$\vec{f}_{CONSTRAINED} = \mu(\omega_0^2 \vec{x} + \gamma_0 \vec{v}_{PROJ}) - (\vec{f}_{ELASTIC} \bullet \hat{x})\hat{x} \tag{9}$$

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 be significantly higher than the animation frame rate.

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 (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.

4.5 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:

$$\mathbf{M}\frac{\partial^2 \mathbf{r}}{\partial t^2} + \mathbf{C}\frac{\partial \mathbf{r}}{\partial t} + \mathbf{K}(\mathbf{r}) = \mathbf{f} \tag{10}$$

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 (3), and **f** is the vector of driving forces.

The positions of the mesh points can then be evolved through time from a given set of initial conditions using a semi-implicit integration procedure to solve a series of boundary value problems. See [Terzopoulos 87] for a more detailed description of the discretization process.

5. IMPLEMENTATION AND RESULTS

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. Figure 4 shows several stages of a penguin character being constructed with the system.

To construct a character, a skeleton is first built interactively using the hierarchy-building tools. Muscle surfaces are then added to the skeleton joints. The skeleton kinematics may be tested at any point in the construction process using interactive inverse kinematics and key posture interpolation. When the muscle surfaces have been added, a rectangular surface mesh (initially in a spherical shape) is created and connected at both ends directly to the skeleton as fixed boundary conditions. At this point, the numerical solver can be started and, at periodic intervals, the surface is rendered on the screen, displaying a continuous simulation. Then the reaction constraints are turned on, pushing the skin surface outside the muscle layer. With the skeleton in a neutral position, individual surface points are then bound to the underlying bone and muscle layer, either automatically (by ray tracing a nearby surface point) or interactively using the mouse. The spring and damping constants are then adjusted to give the desired tightness or looseness of skin attachment. Finally, the thickness of the fat layer is adjusted, either as a global parameter or by sculpting individual points on the skin surface. The character construction process is entirely iterative, that is, the user may step back to any point in the process without losing work.

To animate a character, the user positions the skeleton into a sequence of key postures, either without the elastic surface, or with the simulation running at a low surface resolution for interactive speed. The resulting interpolated motion may then be played back at full speed to check the animation. Later, the resolution of the surface can be increased and the same motion played back in slow, simulation time to calculate a sequence for final rendering.

Using unoptimized code on an SGI Crimson with VGX graphics, we have been able to construct and animate full characters such as the penguin in Figure 5 at a surface resolution of 16 x 16 mass points in one tenth real-time. With a redraw rate set to five frames per second, which is just adequate for interactive work, about half the CPU time is spent redrawing the screen and half running the simulation. When scaled up to 32 x 32 mass points, the simulation slows down by a factor of eight to 1/80th real-time, which is still fast enough for calculating sequences for final rendering.

6. CONCLUSIONS AND FUTURE WORK

We believe that the elastic surface layer model is a promising approach to constructing animated three-dimensional characters. By modeling the skin as an separate elastic surface, which is free to slide along its underlying muscle layers while being held to it by attracting connective tissue, we have been able to simulate a rich variety of realistic-looking animation effects, with a conceptually simple model. Since the physical simulation is of an elastic surface only, while the underlying layers are geometrical models, we have been able to execute the model at interactive rates, allowing a three-dimensional, direct manipulation environment for layered character construction and animation.

One of the main limitations of the current implementation is the finite difference mesh, which has topological restrictions making it difficult to create surfaces with thin appendages (like arms and legs). We are planning to move to a finite element discretization, which should remove these restrictions. Addition of self-collision detection to the skin would allow greater deformations at the joints and more pronounced wrinkling. Adding dynamic properties to other layers such as the fat and muscle layers would also enhance realism

ACKNOWLEDGMENTS

The authors are deeply indebted to Enrico Gobbetti and Francis Balaguer for innumerable ideas, suggestions and 5D Toolkit software tools. We also wish to thank Prem Kalra, Ying Yang, Tsuneya Kurihara, Geoff Wyvill, and Tat-Seng Chua for valuable discussions. The penguin skin texture was created by Ling Wang using Wavefront software. This work was funded by the Swiss National Research Foundation.

REFERENCES

1. Armstrong WW, Green M (1985) The Dynamics of Articulated Rigid Bodies for Purposes of Animation, Proc. Graphics Interface '85, Montreal, pp.407-416
2. Barr AH (1981) Superquadrics and Angle Preserving Transformations, IEEE Computer Graphics and Applications, Vol. 1, No 1, pp.11-23

3. Barr AH (1984) Global and Local Deformations of Solid Primitives, Proc.SIGGRAPH '84, Computer Graphics, Vol. 18, No3, July 84
4. Bloomenthal J, Wyvill B. (1990) Interactive Techniques for Implicit Modeling, Proc. SIGGRAPH Symposium on Interactive 3D Graphics, Computer Graphics, Vol. 24, No2, pp.109-116
5. Chadwick J, Haumann DR, Parent RE (1989) Layered Construction for Deformable Animated Characters, Proc. SIGGRAPH '89, Computer Graphics, Vol. 23, No3, pp.234-243
6. Chen DT, Zeltzer D (1992) Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method, Proc. SIGGRAPH'92, Computer Graphics, Vol. 26, No 2, pp.89-98
7. Culhane S (1988) Animation From Script To Screen, St. Martin's Press, New York 1988
8. Feynman RP, Leighton RB, Sands M (1965) The Feynman Lectures on Physics, Addison-Wesley, Reading Massachusetts 1965
9. Forsey DR (1991) A Surface Model for Skeleton-Based Character Animation, Proc. Second Eurographics Workshop on Animation and Simulation, INRIA/IRISA, Rennes, pp.55-74
10. Forsey DR. Bartels RH (1988) Hierarchical B-Spline Refinement, Proc. SIGGRAPH'88, Computer Graphics, Vol. 22, No4, pp.205-212
11. Gascuel MP, Verroust A, Puech C (1991) A Modelling System for Complex Deformable Bodies Suited to Animation and Collision Processing, The Journal of Visualization and Computer Animation, Vol. 2, No3, pp.82-91
12. Gelfand IM, Fomin SV, (1963) Calculus of Variations, Prentice-Hall, Englewood Cliffs, New Jersey 1963
13. Girard M (1987) Interactive Design of 3D Computer-Animated Legged Animal Motion, IEEE Computer Graphics and Applications, Vol. 7, No 6, pp.39-51
14. Gourret JP, Magnenat-Thalmann N, Thalmann D (1989) Simulation of Object and Human Skin Deformations in a Grasping Task, Proc. SIGGRAPH '89, Computer Graphics, Vol. 23, No 3, pp.21-30
15. Isaacs PM, Cohen MF (1987) Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics, Proc. SIGGRAPH'87, Computer Graphics, Vol. 21, No4, pp.215-224
16. Klein H, Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators, IEEE Transaction on Systems, Man and Cybernetics Vol SMC-132, No. 3, March/April '83
17. Kochanek DH, Bartels RH (1984) Interpolating Splines with Local Tension, Continuity, and Bias Control, Proc. SIGGRAPH '84, Computer Graphics, Vol. 18, pp.33-41
18. Lasseter J (1987) Principles of Traditional Animation Applied to 3D Computer Animation, Proc. SIGGRAPH '87, Computer Graphics, Vol. 21, No4, pp.35-44
19. Magnenat-Thalmann N, Thalmann D (1987) The Direction of Synthetic Actors in the Film Rendez-vous à Montréal, IEEE Computer Graphics and Applications, Vol. 7, No12, pp.9-19
20. Platt JC, Barr AH (1988) Constraint Method for Flexible Models, Proc.SIGGRAPH '88, Computer Graphics, Vol. 22, No4, pp.279-288
21. Sederberg TW, Parry SR (1986) Free-Form Deformations of Solid Geometric Models, Proc. SIGGRAPH'86 Computer Graphics, Vol. 20, No4, pp.151-160
22. Terzopoulos D, Platt JC, Barr AH, Fleischer K (1987) Elastically Deformable Models, Proc.SIGGRAPH'87, Computer Graphics, Vol. 21 No 4, pp.205-214
23. Terzopoulos D, Waters K (1991) Techniques for Realistic Facial Modeling and Animation, in: Magnenat Thalmann N, Thalmann D, Computer Animation '91, Springer-verlag, Tokyo, pp.59-74
24. Turner R, Gobbetti E, Balaguer F, Mangili A, Thalmann D, Magnenat-Thalmann N (1990) An Object-Oriented Methodology Using Dynamic Variables for Animation and Scientific Visualization, in: Chua TS, Kunii TL, CG International '90, Springer, Tokyo
25. Walters G. (1989) The Story of Waldo C. Graphic, 3D Character Animation By Computer, SIGGRAPH '89 Tutorial Notes.
26. Wilhelms J, Barsky BA (1985) Using Dynamics Analysis for the Animation of Articulated Bodies Such as Human and Robots, Proc. Graphics Interface '85, Montreal, pp.97-104
27. Witkin A, Fleischer K, Barr AH (1987) Energy Constraints on Parameterized Models, Proc. SIGGRAPH'87, Computer Graphics, Vol. 21, No4, pp.225-232
28. Witkin A, Kass M (1988) Spacetime Constraints, Proc. SIGGRAPH '88, Computer Graphics, Vol. 22, No4, pp.159-168

Russell Turner is a research assistant and doctoral candidate at the Computer Graphics Laboratory of the Swiss Federal Institute of Technology in Lausanne, Switzerland. He received his B.S. in Physics and his M.S. in Computer and Information Science from the University of Massachusetts at Amherst. He has also worked as a software engineer for V.I. Corporation of Amherst, Massachusetts. His research interests include animation, 3D interaction, physically-based modeling, and object-oriented graphics.

E-mail: `russell.turner@di.epfl.ch`

Daniel Thalmann is currently full Professor, Director of the Computer Graphics Laboratory and Head of the Computer Science Department at the Swiss Federal Institute of Technology in Lausanne, Switzerland. He is also adjunct Professor at the University of Montreal, Canada. He received his diploma in nuclear physics and Ph.D in Computer Science from the University of Geneva. He is coeditor-in-chief of the *Journal of Visualization and Computer Animation*, member of the editorial board of the *Visual Computer* and *the CADDM Journal* and cochairs the EUROGRAPHICS Working Group on Computer Simulation and Animation. Daniel Thalmann's research interests include 3D computer animation, image synthesis, virtual reality and scientific visualization. He has published more than 100 papers in these areas and is coauthor of several books including: *Computer Animation: Theory and Practice* and *Image Synthesis: Theory and Practice*. He is also codirector of several computer-generated films.

E-mail: `daniel.thalmann@di.epfl.ch`

The authors may be contacted at:

Computer Graphics Lab
Swiss Federal Institute of Technology
CH 1015 Lausanne, Switzerland
tel: ++41-21-693-5214 fax: ++ 41-21-693-5328

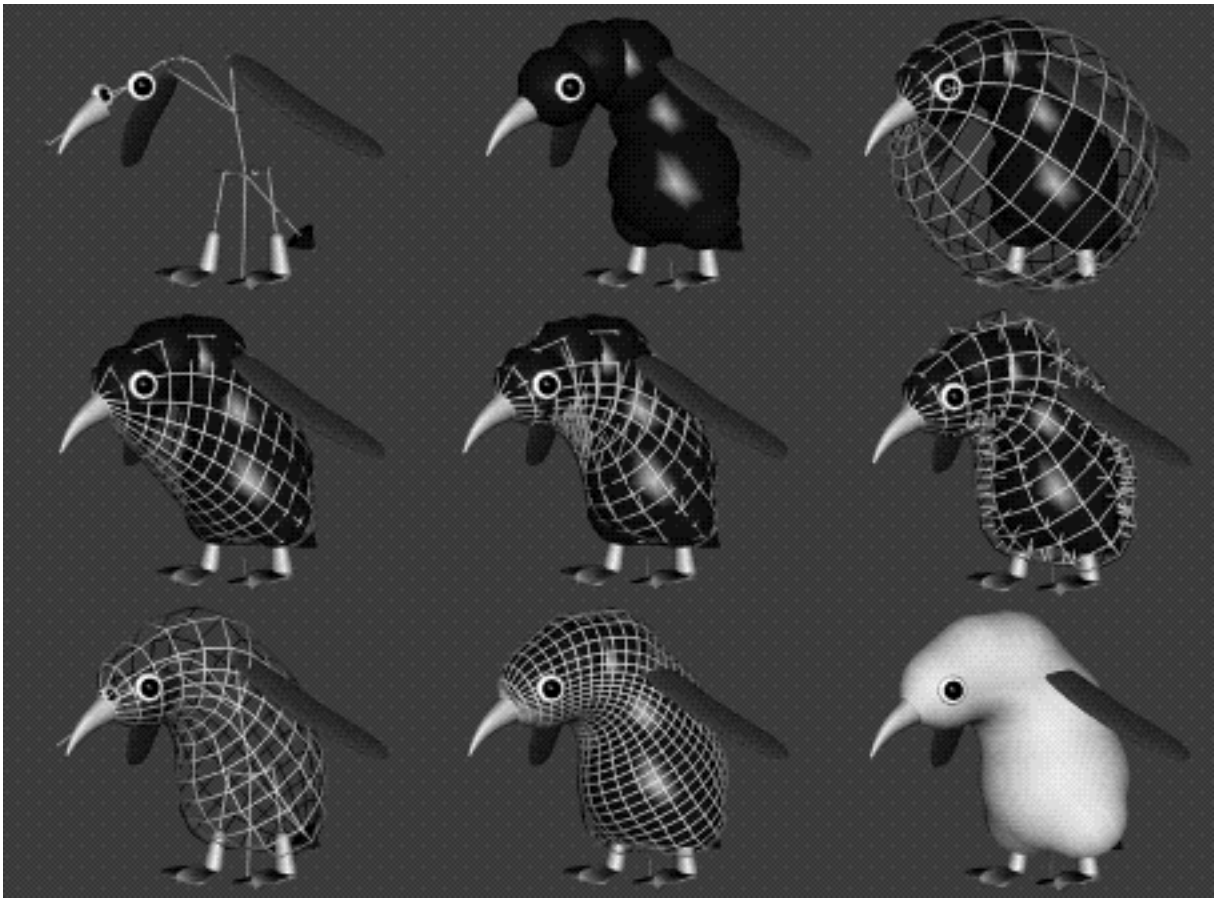

Figure 4: Stages in the Construction of a Layered Elastic Character

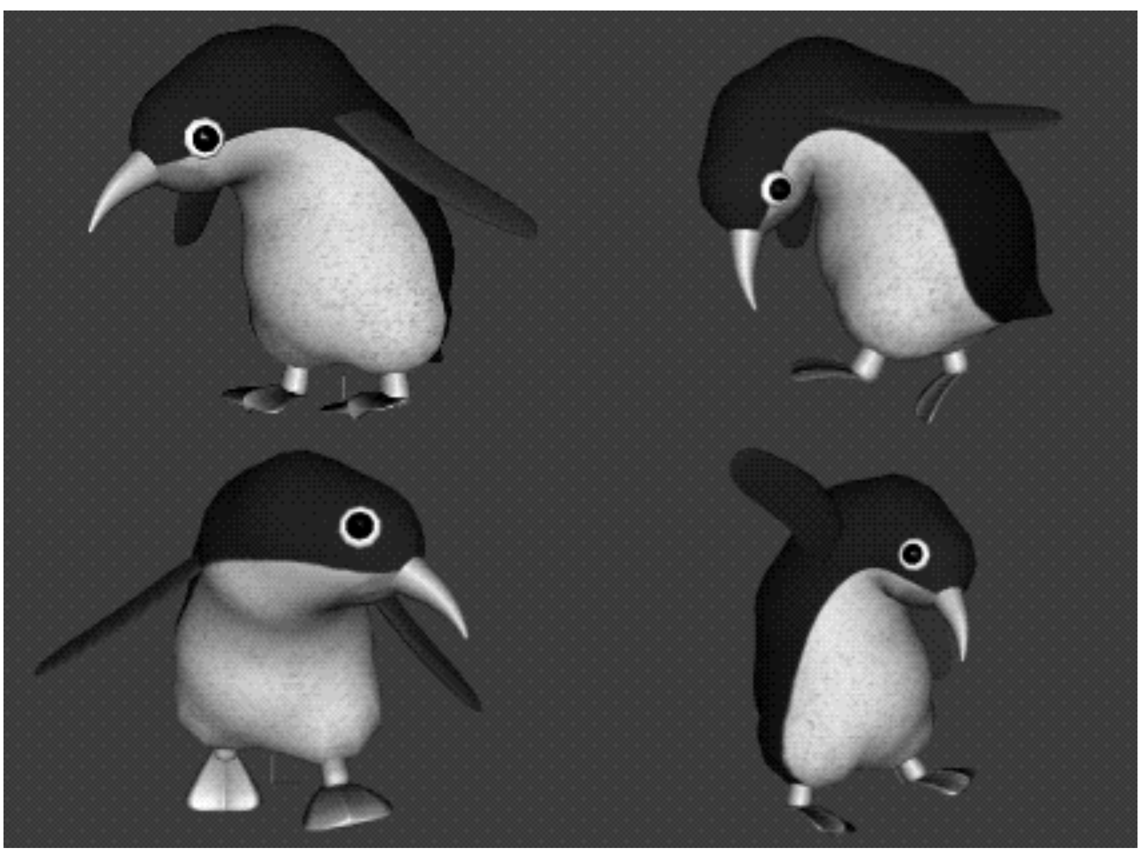

Figure 5: Layered Elastic Character Deformation Under Various Skeleton Postures