Layered Construction for Deformable Animated Characters

John E. Chadwick
David R. Haumann
Richard E. Parent

The Ohio Supercomputer Graphics Project
The Advanced Computing Center for the Arts and Design
The Department of Computer and Information Science
The Ohio State University

Abstract

A methodology is proposed for creating and animating computer generated characters which combines recent research advances in robotics, physically based modeling and geometric modeling. The control points of geometric modeling deformations are constrained by an underlying articulated robotics skeleton. These deformations are tailored by the animator and act as a muscle layer to provide automatic squash and stretch behavior of the surface geometry. A hierarchy of composite deformations provides the animator with a multi-layered approach to defining both local and global transition of the character's shape. The muscle deformations determine the resulting geometric surface of the character. This approach provides independent representation of articulation from surface geometry, supports higher level motion control based on various computational models, as well as a consistent, uniform character representation which can be tuned and tweaked by the animator to meet very precise expressive qualities. A prototype system (Critter) currently under development demonstrates research results towards layered construction of deformable animated characters.

CR Categories and Subject Descriptors: 1.3.7 [Computer Graphics]: Graphics and Realism: Animation; 1.3.5 [Computer Graphics]: Computational Geometry and Object Modeling. Additional Key Words and Phrases: free form deformations, robotic manipulators, kinematics, dynamics, character animation.

1. Introduction

Rendering quality has improved to the point that images with very high levels of photo-realism and full of textural detail may readily be achieved. The development of special purpose graphics engines and massively parallel hardware suggest extremely complex rendering will soon be economically feasible for computer character animation production. The largest obstacle facing the realization of computer character animation is the motion specification itself. An anatomically precise geometric model must also move with an equal degree of realism or we do not accept the illusion. Current commercial animation software literally mimics the key frame methodology of traditional animation. Some successful animated films have been realized with these animation systems; however, much of the burden of specifying the motion of the character form is placed on the animator. The current research efforts of the authors focus on providing a more efficient and effective animation environment designed specifically for constructing and animating characters. We are investigating techniques to exploit the power of computational models so as to provide the animator with fine tuned local control as well as the ability to orchestrate complexity with higher level control.

1.1. Traditional Animation

Animation in a general sense could be defined as "things changing over time". If we are to address the problem of character animation, then this definition is not adequate. The problem of character animation can best be described by the title of the animation bible: "The Illusion of Life", written by Thomas and Johnston [1]. The focus is not on the problem of completing a given motion task, but more importantly on how this motion task is performed by the character. All the elements involved in an animated character must cooperate in a very synchronized harmony. This does not necessarily require realistic behavior, but behavior that is believable, full of an expressive quality which captures the personality of the character. The principles of animation as developed in the Disney heydays are very colorfully presented in [1]. Lasseter provides solid working examples of how these principles have been applied to track based key frame animation [2]. The track based key frame animation approach [3][4] provides a general solution to the animation problem; however, the ability to produce quality animation is principally the burden of the animator. Getting an animation to "jump into life" is the craft of the animator, yet to literally model the methodology of traditional drawn animation grossly underestimates the computational power potentially available by the medium. If we can formalize some of the concepts which underlie the principles of traditional animation such as Squash and Stretch, Exaggeration, Follow Through and Overlapping Action, then we may provide the animator with more intuitively parameterized models while more effectively exploiting the available computational resources.

1.2. Geometric Deformations

The animation of deformable characters requires geometric models of soft tissue which change over time. Lundin and Van Bearle and others have applied surface patch descriptions to model smooth character form [5]. Recently Forsey and Bartels describe a method for hierarchical B-spline refinement which al-

©1989 ACM-0-89791-312-4/89/007/0243 $00.75

lows for multiple levels of control ranging from broad high level surface description to low level fine tuning control in regions of intricate detail [6]. Barr introduced geometric modeling deformations which provide abstract data manipulation operators creating a useful sculpting metaphor [7]. Sederberg and Parry introduced the concept of Free Form Deformations (FFDs) based on hyperpatch solids [8]. FFDs provide the flexibility of general free form spline control coupled with the sculptural flexibility of deformations. For purposes of animation, a key advantage to abstracting deformation control from that of the actual surface description is that the transition of form is no longer dependent on the specifics of the surface itself. FFDs provide the foundation for deformations implemented by the authors.

1.3. Simulation Models

A new area of computer graphics research focuses on the simulation of the physical properties of object models. The motion and shape deformation of the objects can be simulated through applied physics [9]. Barr, Terzopoulos, Platt, Fleischer, and Haumann have applied discrete macro molecular abstractions of the substance properties of the object to model flexible elastic behavior [10][11]. The discrete molecular components of the object can be viewed as point masses interconnected by springs with stiffness and damping attributes based upon the physical properties of the object. Hahn, and Moore and Wilhelms have coupled rigid body dynamics with collision detection and reactive forces to provide realistic animation of objects tumbling and colliding through space [12][13]. Terzopoulos and Fleischer have extended their model to include rigid and flexible components as well as inelastic behavior [14][15]. The finite element lattice used by Terzopoulos et al for flexible models is analogous to the hyperpatch control lattice [10]. Here the spline concept has been extended to include physical properties from whence splines originally sprung :) . Physically based models have proven extremely successful at animating "inanimate" or "not consciously moving" objects. Miller has provided striking results based on a physical model for simulating the self motivated motion dynamics of snakes and worms [16]. The field of biomechanics has been concerned for some time with modeling the physical properties of body tissues as it relates to the areas of artificial implants [39] and the healing of surgical wounds [40], to name just a few. The main difficulties in modeling the dynamics of soft body parts are the structural inhomogineity and the inelastic, time-dependent behaviors of the regions involved. For example, skin, fatty tissue, muscle, and skeleton all exhibit different physical properties complicated by the presence of migratory fluids (blood and lymph) as well as actively contracting muscles. Thus accurate physical simulations of these combined structures requires complex, viscoelastic, anisotropic models such as those presented by Fung [41]. All these simulation methods offer great potential for character animation; however, harnessing this computational power remains largely a problem of animator control. Barr, Barzel, Kass, Platt and Witkin have addressed the control issue by providing user specified constraints, which are then resolved through coupling physical models with constraint satisfaction methods based on various optimization criteria [17][18][19][20].

The ability to control articulated figures is a long standing problem addressed by the robotics community. Robotics research has been applied to the problems of computer animation. Girard and Maciejewski introduced inverse kinematics to the figure animation community by providing computational models for legged locomotion [21][22]. Badler et al. have also employed inverse kinematics towards figure animation, as well as a system for specifying figure positioning based on constraint satisfaction [23]. The dynamic simulation of articulated hierarchies has also been applied to computer animation by Armstrong and Green, and Wilhelms [24][25]. With dynamics we achieve greater physical realism, but the ability to control the desired applied forces to meet specific motion requirements remains an open problem. Animator applied forces such as joint torques seem far less intuitive than direct kinematic specification of joint angles. Isaacs and Cohen applied inverse dynamics, also popular within the robotics community, as a mechanism for integrating kinematic and dynamic control [26]. Often in character animation, however the dynamics are secondary to conveying the emotions of the character.

Due to the complexity of articulated figure motion , a level of control which is higher than the manual specification of every individual moving parameter is needed. Zeltzer addresses the problem of articulated figure control by using a higher level "director" control approach [27]. Animator specified goal directed behavior is resolved through decoupling the goals into task level routines provided by a robust motion library. For the animator, behavioral models are a naturally intuitive means of high level control. In addition behavioral models may algorithmically suggest to the viewer conscious decision making processes on the part of the animated characters. Amkraut and Reynolds have provided models for bird flocking behavior [28][29]. Flocking behavior provides a good example of higher level control orchestrating complexity. From the character animation perspective director level control is often desirable but not sufficient, because it does not allow the animator precise control over the fine details of motion. What is needed is an intermediate "actor" level of control which allows the animator to control, perhaps even act out the gestural details of a character's movement. The layered approach to characters presented in this paper is designed to support motion generated by simulation models while also providing precise control of the transition of the character's form.

2. Layered Construction

In his 87 Siggraph paper [2] Lasseter stresses that the advantage offered by computer animation is one of working the animation in layers. The ability to isolate parameters is essential to fine tuning motion via local control. The ability to additively build an animation in layers provides an effective means for creating complex motion. We would like to extend this notion to motion specified through simulation by building the character in layers and specifying the relationship between the layers through parametric constraints. A layer can be defined as a conceptual simulation model which maps higher level parametric input into lower level outputs. The animator specifies various constraint relationships between the layers and can control the global motion from a high level. The animator defines the layers of the character by specifying parameters and conditions of the constraints. This describes how the character moves and not the specifics of the explicit motion. By providing layers of the character related through parametric constraints, low level motion can be automated through higher level control.

This layered philosophy and the notion that the computer becomes manager of the interacting relationships of the various parametric layers is central to the design of the Critter system developed by the authors. The ability to define attributes as constraints provides a methodology where more emphasis is placed on constructing the character so that less emphasis is needed in actually scripting an animation. Parametric constraints provide the animator with control of gestural detail, while providing consistent, automatic motion control based on layered constraints. This layering philosophy has been adopted both in construction and animation for computer generated characters.

Previous research efforts have also attempted to model animated characters through a layered construction approach. Burtnik and Wein presented a 2-D approach to bending a digitized drawing about an underlying articulated figure using constrained deformations [30]. Parent presented a 3-D approach to bending geometric surfaces about articulated forms for purposes of character animation [31]. Tony De Peltrie was an exciting example where a digitized character geometry was applied to an underlying articulated skeleton. The Thalmanns have created a human factory system designed to animate synthetic actors [32]. Dick Lundin and Susan VanBearle provided animated dancers by algorithmically weaving a surface patch skin on top of an underlying articulated skeleton as well as integrating dynamics to model free fitting clothing [5]. John Donkin's Dinosaur provides another example where a digitized surface geometry skin is fit to an underlying articulated skeleton [33]. These methods were designed to meet specific film production demands. They primarily focus on a three layered approach (motion specification, articulated skeleton, geometric skin) and are tailored to special purpose models. The current approach of the authors provides a more general solution by using a four layered construction approach. The approach described in this paper not only fits a geometric skin to an articulated skeleton, but also captures the fluid squash and stretch behavior of the surface geometry by providing volumetric muscle bulging, dynamic fatty tissue response, and creasing at the joint. The Critter system is designed to provide a flexible interface to the animator for constructing and animating deformable characters.

The authors' philosophy supports a four layered approach from high to low levels as follows:

1. Motion specification
   (referred to as the behavior layer in the critter system)
2. Motion Foundation, articulated armature
   (critter skeleton layer)
3. Shape transition, squash and stretch
   (critter muscle and fatty tissue layer)
4. Surface description, surface appearance and geometry
   (critter skin, clothing and fur layer)

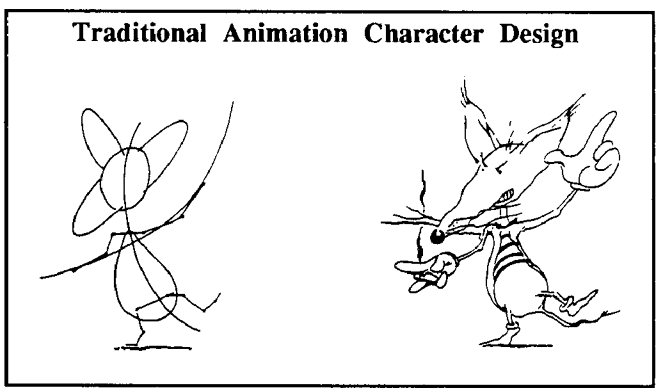

Figure 1.

The layered construction approach adopted by the Critter system is similar to the construction methods described for traditional drawn characters (Figure 1). The stick figure drawing is analogous to the robotics armature or skeleton. The drawn blobby volumetric shapes used to flesh out the character form are analogous to muscle and fatty tissue and help provide consistent fluid squash and stretch transitions in shape. The detailed character sketch is representational of the visible surface geometric "skin". The key advantage to the layered computer methodology is that once the layered character is constructed, only the underlying skeleton need be scripted for an animation; consistent yet expressive shape dynamics are generated automatically. Various character layers can be used as templates for other similar character forms, thereby providing a robust library of extendable character parts.

The skeleton (second) layer is an underlying articulated hierarchy which provides the foundation for controlling the motion of the character. The muscle layer is then added on top of and attached to the skeleton hierarchy. The foundation for the muscles are represented by freeform deformations. The control points of the deformations are constrained by the positioning (joint angles) and forces (joint torques) applied to and by the underlying skeleton. These deformations then act to glue and deform the actual geometric skin to the underlying skeleton. The skin layer represents the actual visible surface geometry to be rendered. The current implementation supports polygonal skin data based on the existing modeling and rendering environment at ACCAD. The application of FFDs as a foundation for muscle and fatty tissue deformations provides for general extension to potential surface patch, algebraic, or volumetric skin. Each skin object is attached relative to a link in the skeletal hierarchy. This attachment defines the local coordinate space of the skin component. The skin object may then be attached to any number of deformations. Rigid "skin" objects may be connected directly to the skeleton with zero connecting muscles. This allows the system graceful degradation to a basic object hierarchy. Figure 2 provides an example where no muscle layer was provided. A skeleton database (Critter skin) of several bones was attached to a simplified articulated skeleton to provide a reasonable number of controllable joints.

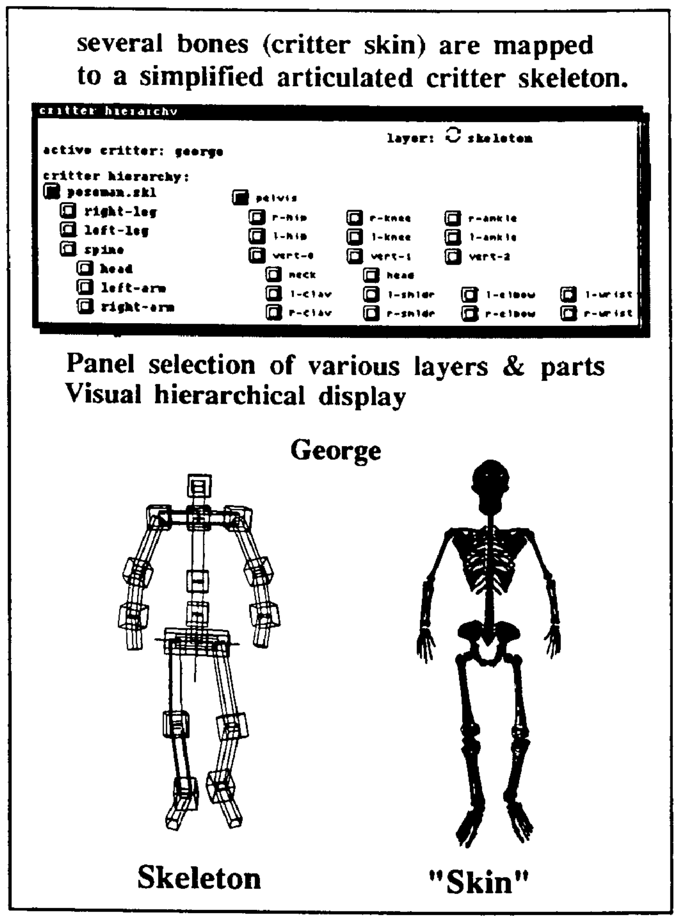

Figure 2.

The behavior (first) layer which represents the actual motion specification need only be applied to the skeleton parameters. A general purpose attribute based behavioral foundation was provided so that the animation system could be easily extended to include various computational motion models. The foundation for behavior provides a pose vector which contains the character position, orientation and joint angles for each frame. Additional structures are provided to house velocities, accelerations, joint torques, and user specified gesture attributes. To date, the interactive and animation behavior models within the Critter system are based on forward and inverse kinematics with additional procedural modeling capabilities. The muscle and skin layers can be automatically generated based on a given skeletal state.

3. Skeleton Layer

A description is provided of the skeleton layer to clarify the interaction of the articulation hierarchy to the deformations which are built on top of and constrained by the skeleton. The skeleton acts as the character foundation, providing the articulation hierarchy from which additional layers are built and constrained.

The skeleton data includes:

1. Tree structured hierarchy of robotic manipulators
2. Robotic joint-link parameters (Denevit & Hartenberg)
3. Joint angle constraints and physical attributes (max, min, zero, stiffness, mass properties)

The basic joint hierarchy has been extended to a hierarchy of manipulators. A manipulator is basically a sequential chain of interconnected joints and links. Each manipulator may contain any number of child manipulator parts. A child manipulator may be connected to any joint within its parent manipulator chain so that none of the generality of a basic joint hierarchy is lost. The manipulator extension lends itself readily to robotic kinematic and dynamic motion models, in particular inverse kinematics. Denevit & Hartenberg parameters (figure 3) are implemented as the basic joint construct, where:

$$\text{Joint}_i = [\ a_i,\ \alpha_i,\ d_i,\ \theta_i\ ]$$

a_i and α_i represent the link parameters where α_i is the twist angle between links and a_i represents the link length. θ_i and d_i represent the joint parameters where θ_i is the joint angle and d_i the offset length along the axis of joint rotation. A more detailed description of the D & H parameters can be found in the robotics literature [34][35][36]. These joint primitives provide only single degree of freedom joint motion. If θ_i is varied over time as a parameter then the joint is rotational. If d_i is varied over time as a parameter then the joint is translational. Primitive robotic joints can be combined at common origin ($a_i = d_i = 0$) with perpendicular joint axis ($\alpha_i = 90^o$) to create universal (2-rotary axis) and ball (3-rotary axis) joints.

Minimum and maximum joint angle constraints provide the animator with the capability to restrict the range of joint motion. The zero angle and stiffness at the joint provide the animator with control over the relative bending at each joint within a manipulator when automated by inverse kinematic control. Mass properties: mass, center of mass, and inertia tensor (distribution of mass) parameters are also provided; however, current behavior models and interactive figure control do not as yet exploit the robotic dynamic models for articulated figures. A more detailed description of the Critter skeleton may be found in [37].

4. Muscle and Fatty Tissue Layer

The constrained deformation layer is central to Critter construction. The deformation acts as the connecting relationship for mapping the geometric skin data to the underlying articulated skeleton foundation, while capturing the flexible fluid quality of squash and stretch behavior. To automate the squash and stretch behavior, the animator specifies muscle and fatty tissue attributes by defining constraint relationships with the underlying skeletal parameter state. The foundation for the muscle and fat deformations is based on Free Form Deformations (FFDs) [8]. Composite tricubic bezier based hyperpatches (or parametric solids) are used as the basis for the FFD as muscle deformation abstractions. The current implementation of muscle and fatty tissue structures represents each muscle primitive as a pair of adjoining FFDs (Figure 4). In an attempt to provide muscle abstractions which deform the skin surface a prototype set of functional deformation operators are provided. Muscles are represented by a pair of FFDs. This provides 7 planes of control points orthogonal to the adjoining joint link axis: four planes for each FFD (cubic bezier) with one plane shared as the adjoining connection between deformations. The two control planes at either end of the muscle deformation (adjoining planes) function principally to preserve continuity across connected muscles. Continuity can also be preserved between a muscle and a non-existing (null muscle) by assuming the null muscle remains undeformed. The remaining three planes (mid planes) function to model the abstract muscle behavior resulting from kinematic or dy-

Denevit and Hartenberg Parameters

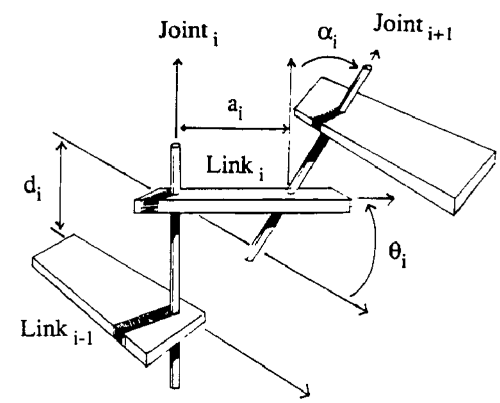

Figure 3.

Abstract Muscle Deformation: pair of adjoining FFDs

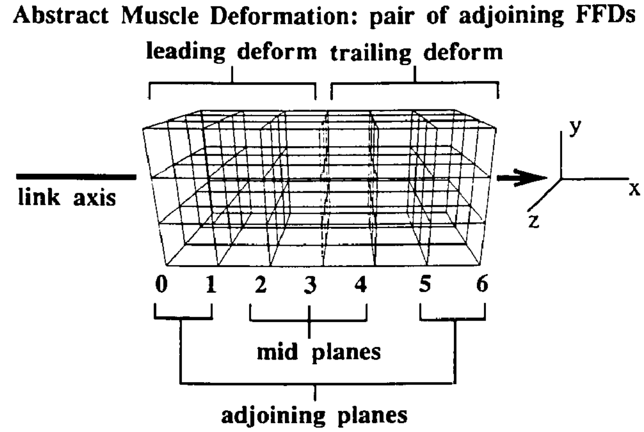

Figure 4.

namic attributes of the skeletal state. Each operator or muscle type provides a set of parameters for defining the relationship of the control points of the FFDs with the kinematic and dynamic skeletal parameters.

Free form deformations can best be described as a cubical volume in which geometric objects are submersed. If we think of this cube as a chunk of jello which can be bent, shaped, or contorted, then the objects within the volume are distorted accordingly. The basis of the FFD is a trivariate hyperpatch or parametric solid. A set of control points form a three dimensional lattice within the cube. These control points are used by the blending functions to map parametric weights to positions in space. For each vertex comprising the objects embedded within the cube, the three parametric weights can be determined, which when substituted into the blending functions for the undeformed cube, produce the position of that vertex in the undeformed object. By manipulating the control points which form the lattice, the cube solid is deformed. The resulting vertex positions of the deformed object are computed by using the deformed lattice control points in the hyperpatch blending functions and then sampling at the parametric weights associated with the original undeformed vertices.

Current deformations are based on kinematic, dynamic, or sculpted constraints. To provide automatic squash and stretch behavior of the character, muscle deformations are modeled to provide bulging and bending of the character geometry based on the kinematic state of the articulated skeleton. To increase the expressiveness of the character the dynamics of the passively deformable body parts such as flesh and underlying soft tissue are modeled to provide automatic follow through and overlapping action. Exaggeration can be produced by adding sculpted deformations which give the animator explicit control over the shape transition of the character form.

4.1. Kinematic Deformation

By providing muscle deformations which are constrained by the kinematic skeletal state, automatic, consistent squash and stretch behavior can be achieved. While in reality our muscles control the skeletal joint motion, from an animation perspective we inversely would like the skeletal joint motion to automatically create the resulting muscle flexion required to meet the specified motion. The kinesiology literature suggests the following mechanical properties of joint and muscle action [38]:

Elasticity- When tension is applied to a muscle, a passive elongation results accompanied by a reduction of the cross sectional area of the muscle. A weight attached to a relaxed muscle causes an elongation (E) which is directly proportional to the original muscle length (L) and to the pulling force (F) and a constant (k) which varies for each body, and inversely proportional to the cross sectional area (A).

$$E = \frac{F * L * k}{A}$$

Contractility- is the ability of the muscle to shorten by nervous stimuli. The contraction of the muscle is an active process, as opposed to the passive elastic elongation. The muscle tension represents the force, while the change in length during contraction represents the distance covered by the application of this force, whose product is the visible work accomplished. A constant relationship exists between the natural length of the muscle, the variable length of the contracted muscle, and the degree of rotation at the joint. For each unit of shortening (S) there is a constant rotation angle (θ).

$$S = \theta * k$$

The basic property of contraction can be applied to algorithmic models such that the kinematic joint angles act as controlling parameters for abstract muscle behavior. Flexor and extensor muscle deformation models function to provide the visible result of muscle contraction. The algorithm for resolving the resulting control points of a flexor - extensor for each frame follows:

1. The overall shortening of the muscle = joint angle * displacement ratio. This shortening is propagated across each control plane of the muscle by the square of the ratio of length up to control plane / total muscle length.

2. The implied shortening is countered by scaling up and out along the local y & z coordinates. If all muscle boundaries are active, then this scaling will be distributed equally in all directions for each of the 3 midplanes (see Figure 4). If one boundary side is not active, then the scale factored is doubled along the active boundary opposing the not active side. Fixed boundaries can be used to shape the deformation, as in the biceps example (Figure 5.), or to maintain null continuity, in particular, areas in which the deformation intersects a skin component object. The radius ratio determines the relative scale factors of the two exterior mid planes (2 & 4). The center mid plane (3) is resolved by line plane intersection to maintain continuity across the contracted surface.

3. The adjoining planes of the flexor - extensor are maintained to assure continuity with actively connected muscles, or remain undeformed with the exception of shortening along the link axis (x).

In addition to kinematic muscle contraction, tendon deformations model the bending at the joint. This is designed to cover very short regions of the character where a single geometric skin crosses over a joint, covering part of two or more skeletal links to account for underlying bone tissue. The control points of the tendon FFDs are resolved for each frame based on the following algorithm:

1. Determine the bisection angle of the joint. If the angle exceeds the threshold angle, then bisect the threshold angle; else bisect the joint angle. For a hinge joint bisection angle = angle / 2. For joints with more than one degree of freedom we need to know the resulting axis of rotation. Quaternions provide a convenient means of resolving the single axis, and angle about the axis resulting from multiple rotations about several axes. Using this axis & and angle we can now resolve the bisection angle.

2. Rotate mid planes by bisection angle about their local z axes. We rotate planes 2 & 3 by the bisection angle, and plane 4 by -bisection angle. (When the trailing deform is rotated about the joint by the joint angle plane 4 will have proper bisection alignment)

3. Project mid planes 2 & 4 onto boundary cube of deform. Scale control points to avoid intersections with interior region of deform. (This scaling is motivated by underlying bone structure forcing the skin and muscle to stretch, not penetrate deform interior)

4. Slide planes 0 & 1 away from plane 2; slide planes 5 & 6 away from plane 4. Find closest point to adjoining planes. Use this point as reference to maintain distance from adjoining plane to midplane.

5. Rotate trailing deform by joint angle at joint.

6. Use line plane intersection for line segments connecting each control point pair from midplanes 2 & 4, intersected with bisection plane or midplane 3 (plane at joint). Modify control points of joint plane 3 at intersection to maintain C1 continuity.

7. If threshold is exceeded, rotate joint plane by additional angle (bisection of (joint angle - threshold angle))

8. Use line plane intersection to maintain continuity for outside control points only (crease now forms for interior points)

9. Working out from midplanes (2 & 4) towards adjoining planes (1, 2, 5, 6) check for intersection with bisection plane at joint (3). If intersection occurs within bounded plane (0 < t < 1 for parametric line segments) then set control points which cross bisection plane 3 to intersection point. Use parametric weight t to scale control plane out to maintain volume from squash.

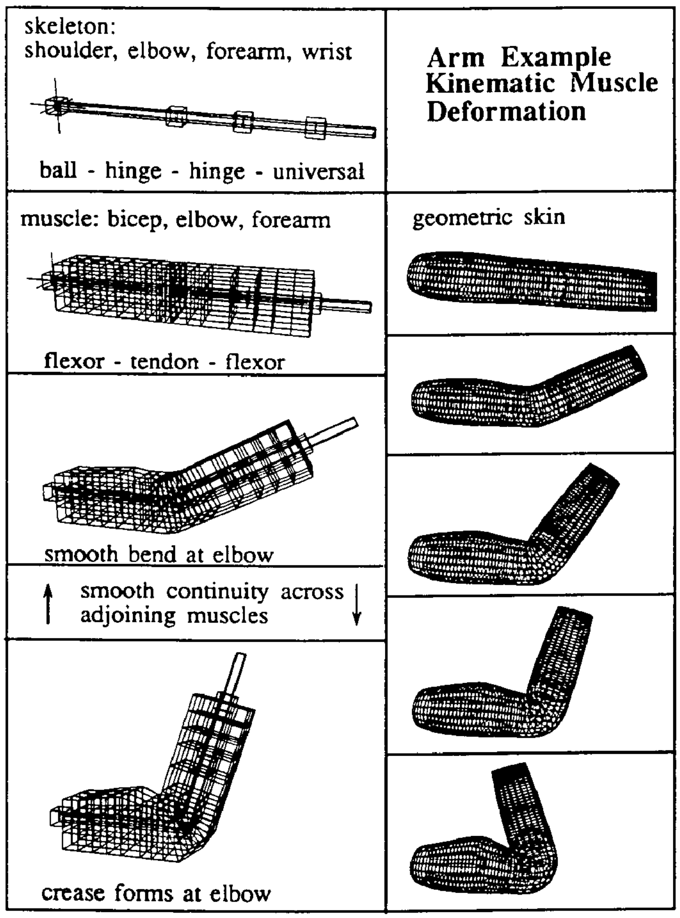

Figure 5.

4.2. Dynamic Deformation

From the point of view of computer generated character animation, we wish to capture the dynamic properties of soft body structures, but without incurring the costs of a complete physical model. Of the physical properties that the simulation must handle, one of the most desirable effects is that of large deformations (in homage to Tex Avery!). In addition, viscous effects must be modeled in order to realistically simulate the damped oscillations of soft parts which result from the character's motion. Finally, the model must allow for spatial variations of the physical properties so as to model the different structures within the character. Fortunately, for these purposes, a complete physical model is unnecessary because, given the space-time scales of interest, many of the effects due to the variation in structures are visually insignificant, or can be greatly simplified. The model of deformable body parts presented here was developed specifically to capture these nonlinear, viscoelastic and anisotropic properties, yet includes spatial simplifications which help reduce the computational cost. Our technique maps the control points of the FFD to point masses in a similarly shaped force lattice. Dynamic simulation is performed on the mass particles and the resulting motion is mapped back onto the FFD control points thus determining the resulting object deformations.

The physical model we employ is a three-dimensional extension to one developed using the behavioral test-bed described in [11]. The model consists of a three-dimensional grid of point mass elements (3 degrees of freedom) connected by viscously damped Hookean springs. The spatial simplifications of the dynamics models results from the one to one mapping maintained between the point masses and the control points of the FFDs. To capture shear strain behavior, spring elements are connected diagonally between mass elements on adjacent planes. Thus, if one were to isolate one "cube" from the grid, it would appear as having one point mass at each of the eight corners, with each mass being directly connected by spring elements to the remaining seven (see Figure 6). Intuitively, the springs aligned with the major axes serve to maintain the linear dimensions of the body, while the cross springs help maintain the angular relationships between the grid planes.

The dynamics are simulated by marching along at discrete sub-frame time steps. At each step, the spring forces are calculated and applied to the point masses, which respond by accelerating in the direction of the net force applied. Using a 2nd order runge-kutta scheme with adaptive stepsize, the equations of motion are integrated to determine the subsequent position and velocity of each point mass. In order to transfer the motion of the animated character to the physical model, we allow certain chosen mass points to be rigidly fixed to the character skeleton, while the remaining mass points are free to dynamically respond. In effect, these fixed points represent the rigid structure of the internal skeleton of the character. As the character moves, changes in the positions of the attached points result in spring displacements which force the remaining mass points into motion.

Dynamics is applied as a post processing step after the articulated skeletal motion has been completed. The steps are as follows:

1. The entire motion of the character as specified through high level control is precomputed.

2. Once computed, the position information for each control point for each FFD at each frame in the animation is known. Certain points of each FFD are designated as non dynamic (fixed to move along these pre-computed "paths").

3. The corresponding physical models are constructed. The motion paths of the non-dynamic control points are used as scripts to drive the corresponding point masses within the dynamics model. The remaining points are free to move as a result of forces generated with the force lattice.

4. Once the dynamic simulation is completed the positions of the free point masses are mapped back onto the positions of the corresponding "dynamic" control points of the FFD. This determines the dynamically deformed shapes of the objects affected by the FFD.

5. Continuity is maintained across internal regions of the shared midplane, and with adjoining or null deformations.

The selection of the physical constants is performed upon the basis of the visual appeal of the resulting motion. The user controls the spring constants, damping coefficients, and the mass properties of each element individually or of the model as a whole. We do not attempt to make these parameters correspond to actual measured values, relying instead upon the animators tastes, which can fall anywhere within the real-surreal spectrum.

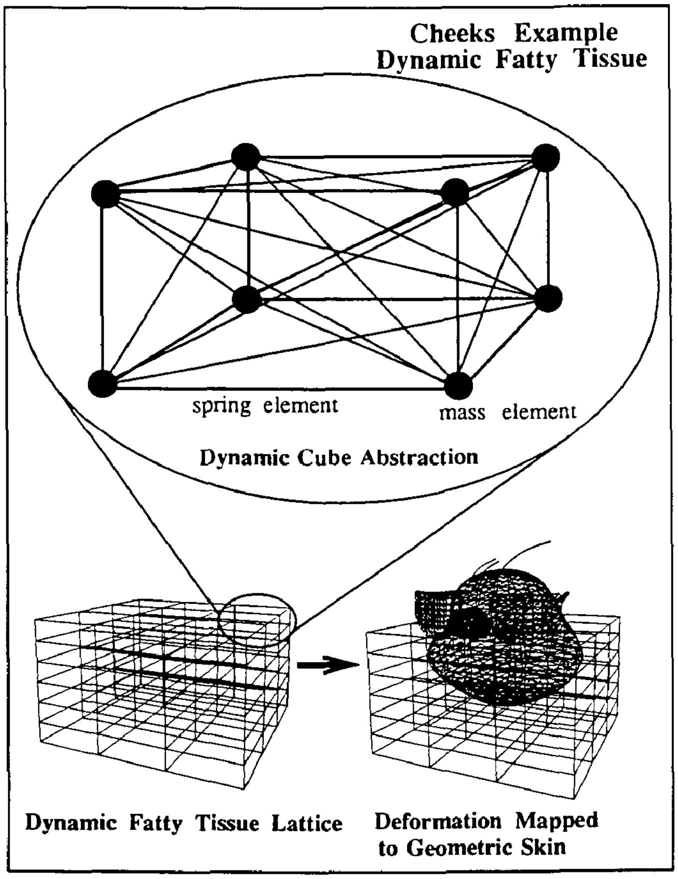

Figure 6.

4.3. Sculpted Deformation

A fundamental design objective of the authors was to provide a general system for constructing and animating computer generated characters. Sculpted deformations provide the most general, but also the most labor intensive deformation primitive supported by the Critter system. Sculpted deformations are principally key framed deformations. The animator sculpts the deformations by moving individual control points of the 3-D lattice defining the FFDs, or through translation, scale or rotation of control point planes or selected control point groupings of the FFDs. These key deformations are then bound to a key attribute. If the application of deformation is to imply muscle reaction, then the key attribute may be constrained by the controlling skeletal joint angle. Several key deformations may be stored as extremes relative to various joint angle positions. The resulting deformation for any given skeletal state is then determined by applying cubic spline interpolation for each corresponding control point of the FFDs. The parametric weight of the interpolation is based on the relation of the current joint angle to the surrounding key angle attributes associated with each key deformation. Sculpted muscles can be used if other muscle models do not adequately provide the shape transition desired, or to add definition to other muscle models.

Sculpted deformations may also be used to provide deformations representational of emotional contortions. Gestural qualities may be sculpted to provide visual exaggeration. Classic cartoon examples of gestural deformation are pride ("V") and sorrow (slumped). The ability to drive a sculpted deformation by an attribute exterior to the skeletal parameters frees the animator to control the parametric blending of various key deformations by any behavior parameter. This removes the automatic nature of the deformation, requiring that animator to now control the level of deformation from the scripted behavior. This was added as an exception to the rule of only specifying skeletal attributes via behavior for added flexibility and generality. For example, breathing can effectively be modeled by oscillating the driving parameter of a sculpted deformation algorithmically.

4.4. Continuity

Maintaining smooth continuity is integral to the function of muscle deformations. The ability to build complex musculature in layers depends readily on the ability of these composite muscle deformations to maintain smooth C1 continuity where desired. For bezier curves this requires a colinear relationship of the three corresponding control points from each midplane. Initial implementations discussed in [37] provided this colinear condition by placing the central midplane control point at the line of intersection of the neighboring midplane control points based on a line - plane intersection calculation. This worked. However it was extremely limited in controlling the shape of the bend, and provided little capability for the animator to control the implied underlying bone tissue. Least square line fitting was implemented to provide the model for securing continuity across adjoining deformations. Rather than an all or nothing line fit, a best fit strategy is employed based on the chi-square merit function. Animator specified weights for each control point can be used to sculpt the resulting continuous boundary regions.

Least square line fitting:

$$p(u) = p(u; a,b) = a + bu$$

Chi-square merit function:

$$\chi^2 (a,b) = \sum_{i=1}^{N} ((p_i - a - bu_i) / \sigma_i)^2$$

σ_i represents the uncertainty associated with each p_i assuming the u_i's are known. We want to minimize the merit function to determine a and b, by setting the partial derivatives with respect to a and b to zero:

$$\delta\chi^2/\delta a = -2 \sum_{i=1}^{N} ((p_i - a - bu_i) / \sigma_i^2) = 0$$

$$\delta\chi^2/\delta b = -2 \sum_{i=1}^{N} ((u_i (p_i - a - bu_i)) / \sigma_i^2) = 0$$

We can now solve for the best fit line given two equations and two unknowns (a & b). Let $w_i = 1 / \sigma_i^2$ were w_i represents the respective relative weight of contribution for the associated p_i. We now consider these weights as an intuitive means of controlling the shape of the resolved continuous curve as opposed to a measure of uncertainty. The animator provides weights for each control point of the deformations to sculpt the resolved continuous form.

Overlapping continuity is resolved by providing "null" continuity across borders which affect overlapping regions. Null continuity simply implies that the first two control points remain fixed so that the border region maintains unchanged with respect to its first derivative.

5. System Overview

Figure 7 provides a diagram of the information flow within the current Critter system implementation and how it relates to the animator interface. The construction of Critters is tightly coupled with the interactive animation process. The animator can tune and tweak the character while scripting an animation.

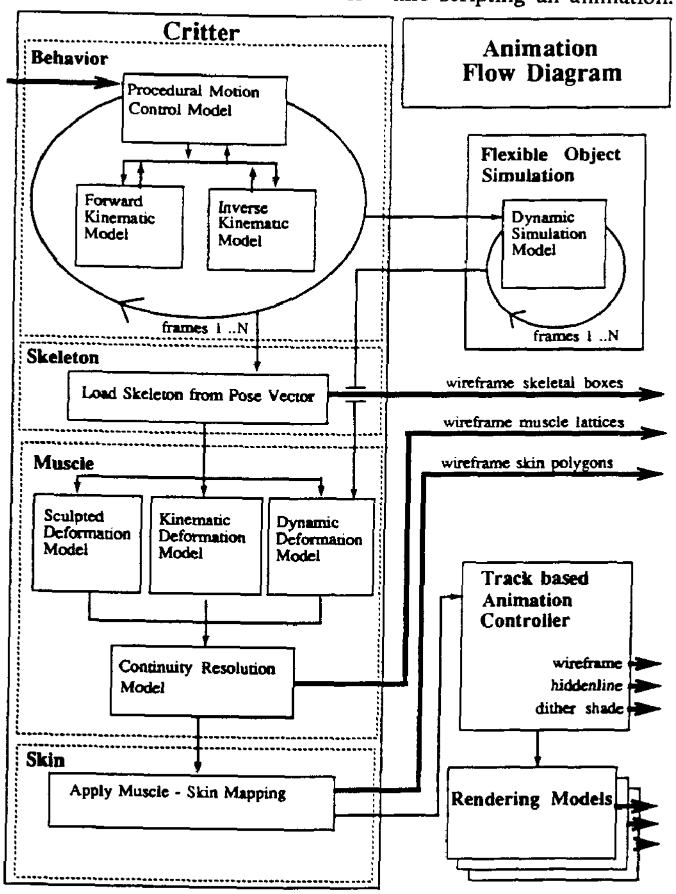

Figure 7.

The system allows the animator to construct motion studies which aid in the intuitive understanding of the character parameter space. Given an established Critter, the behavior attributes are processed initially for each frame. This creates a skeletal pose vector for each frame. If dynamic muscles are present then this precomputed behavior is fed to the dynamic simulation process which then computes the dynamics for the complete sequence of frames. Kinematic and sculpted deformations as well as the deformation to skin mappings are computed on the fly, so that direct access to any frame can be provided in pseudo real time (with the exception of muscle- skin mapping which is on the order of a second running on a sun4/110 workstation). The animator may interactively select viewing of skeleton, muscle or skin layers within the Critter system, or pipe this information into Chalk, a track based animation system developed by John Donkin. The integration with Chalk provides animators with the ability to integrate Critter motion with track based motion and supports various display capability (B&W dithered shading, and hiddenline, as well as a link to various rendering algorithms developed by Scott Dyer).

5. Results

Bragger Bones, a character designed by Don Stredney provided the first attempt at a full character representation (Figures 8, 9, 10 & 11). Initial studies concentrated on the character's limbs and head. Kinematic deformations were placed on the arms and legs and dynamic deformations were placed on the cheek region of the head. Flexor deformations were used to model the biceps, forearms, thighs and calves, while tendon deformations were used to model the elbows and knees. Figure 9 presents every 5th frame from an animation test. The biceps contraction was exaggerated by a factor of 1.5. The bending at the elbows was thresholded to form a crease at an angle of 40 degrees., while the bending at the knees was thresholded at 30 degrees. Default continuity weighting was used for all deformations. Adjoining continuity was maintained between biceps, elbow and forearm deformations, as well as between thigh, knee and calf deformations. Null continuity was maintained at the top of the dynamic cheek deformation so that a smooth surface resulted between the dynamic cheeks and the rigid top of the head.

6. Summary

The Critter system developed at ACCAD provides multi-layered construction and animation of deformable characters. The deformation layer can be constructed by layering several local and global deformations which are constrained by the underlying articulated skeleton. By placing more emphasis on constructing a more elaborate character model, less emphasis is placed on the parameter specification required to script an animation. Squash and Stretch, Follow through and Overlapping Action as well as Exaggeration become more automatic, and consistent. The methodology presented provides a more automatic environment for the animator to meet desired animation specifications. Through computational models which formalize the principles of animation a more intuitive parameterization for the animator is provided. This approach does not attempt to replace animators by algorithmic models, but intends instead to provide them with a more powerful medium for artistic creativity.

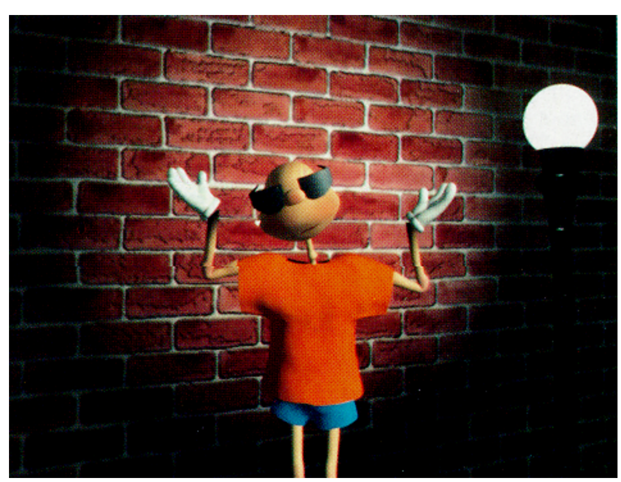

Figure 8. Bragger Bones

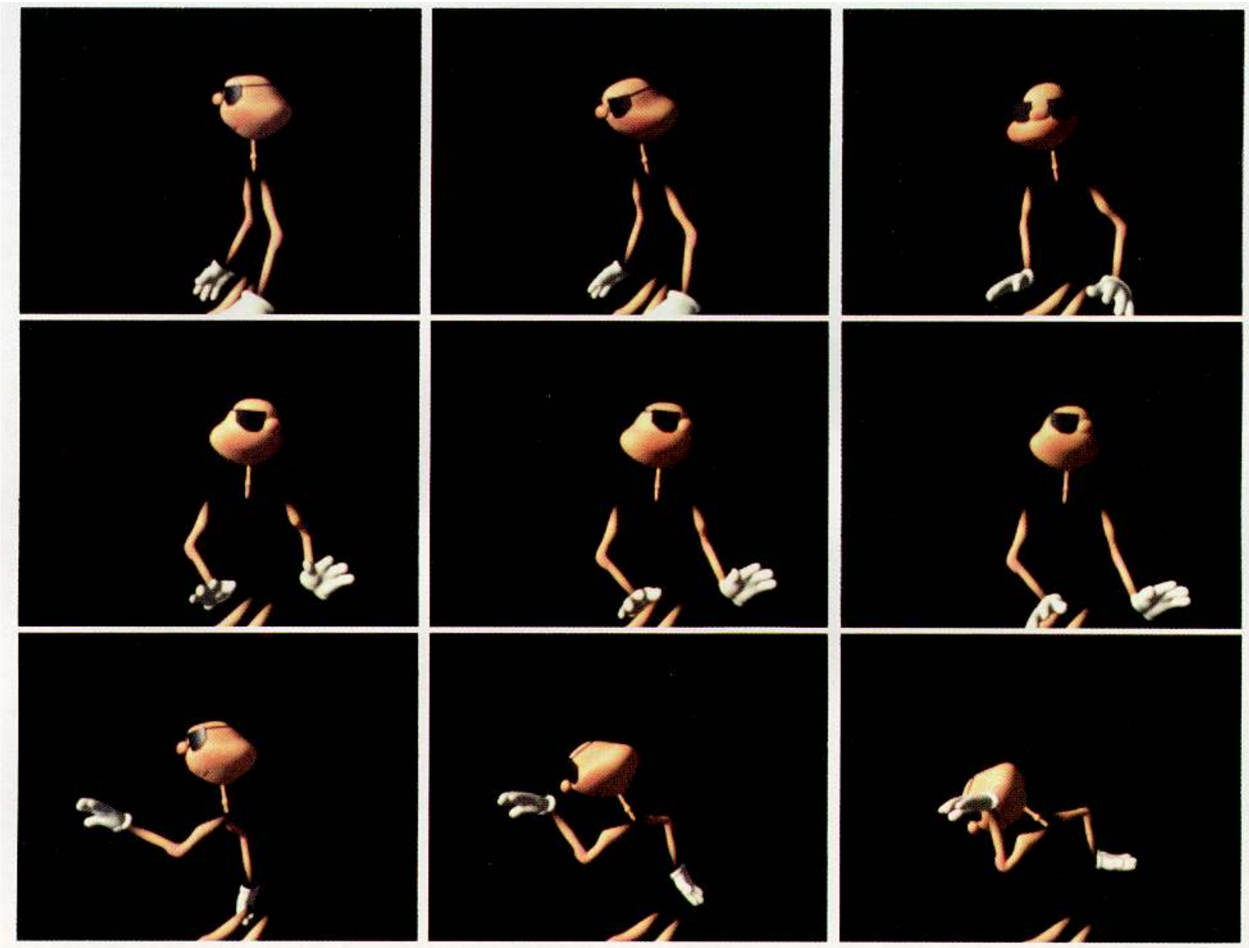

Figure 9. Bragger Animation Test: Kinematic and Dynamic Deformations.
(Every 5th frame, left to right, top to bottom)

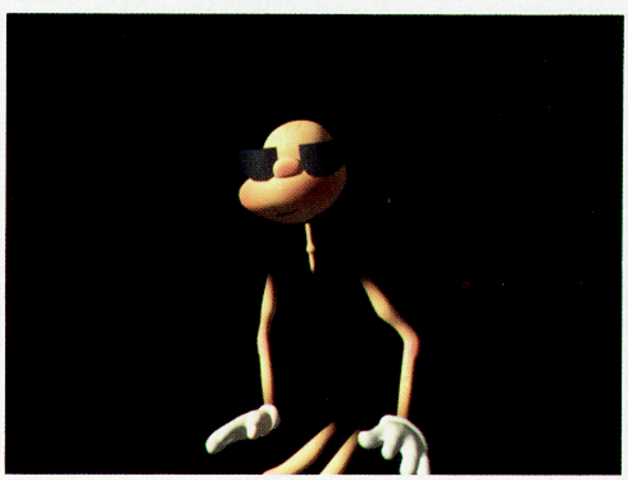

Figure 10. Dynamic Cheek Deformation
(top right frame from above)

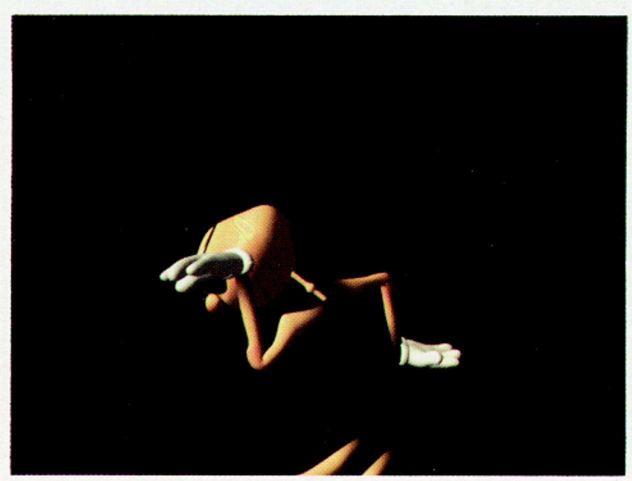

Figure 11. Kinematic Arm Deformations
(bottom right frame from above)

7. Acknowledgments

Don Stredney designed the initial prototype critters, and provided valuable background assistance in anatomy and kinesiology, as well as providing the hand drawn illustrations for this paper. Michael Girard provided valuable insight into related robotics issues. John Donkin provided the chalk - critter interface, with system support from John Fujii, who was invaluable in getting the illustrations for this document together. The rendering software was developed by Scott Dyer. The illustrations and video examples were produced using apE utilities provided by the Ohio Supercomputer Graphics Group. And a special thanks to C.G. for the pasteups and everything. Thanks to Chuck Csuri and Tom Linehan for the environment and opportunity at ACCAD, and to the Ohio Supercomputer Center. This work was supported in part by a grant from Cray Research.

8. References

[1.] Thomas & Johnston, "Disney Animation: The Illusion of Life", Abbeville Press, New York, '84.

[2.] Lasseter, John, "Principles of Traditional Animation Applied to 3D Computer Animation", Computer Graphics, Vol. 21, No. 4, '87.

[3.] Gomez, Julian, "Twixt: A 3-D Animation System", Proceedings of Eurographics 84, '84.

[4.] Sturman, David, "Interactive Keyframe Animation of 3-D Articulated Models", Proc. Graph. Interface, '84.

[5.] Van Bearle, "A Case Study of Flexible Figure Animation", 3-D Character Animation by Computer Course Notes, Siggraph '87.

[6.] Forsey & Bartels, "Hierarchical B-Spline Refinement", Computer Graphics, Vol. 22, No. 4, '88.

[7.] Barr, Alan, "Global and Local Deformations of Solid Primitives", Computer Graphics, Vol. 18, No. 3, July, '84.

[8.] Sederberg & Parry, "Free Form Deformations of Solid Geometric Models", Computer Graphics, Vol. 20, No. 4, '86.

[9.] Barr, Alan, "Dynamic Constraints", State of the Art in Image Synthesis Course Notes, Siggraph, '86.

[10.] Terzopoulos, Platt, Barr, and Fleischer, "Elastically Deformable Models", Computer Graphics, Vol. 21, No. 4, July, '87.

[11.] Haumann & Parent, "The Behavioral Testbed: Obtaining Complex Behavior from Simple Rules", The Visual Computer, Special Issue on Mechanics, Control and Animation, Vol. 4, No. 6, Dec., '88.

[12.] Hahn, James, "Realistic Animation of Rigid Bodies", Computer Graphics, Vol. 22, No. 4, '88.

[13.] Moore and Wilhelms, "Collision Detection and Response for Computer Animation", Computer Graphics, Vol. 22, No. 4, '88.

[14.] Terzopoulos and Fleischer, "Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture", Computer Graphics, Vol. 22, No. 4, August'88.

[15.] Terzopoulos and Witkin, "Physically Based Models with Rigid and Deformable Components", IEEE CG&A, November, '88.

[16.] Miller, Gavin, "The Motion Dynamics of Snakes and Worms", Computer Graphics, Vol. 22, No. 4, '88.

[17.] Witkin, Fleischer, & Barr, "Energy Constraints on parameterized Models", Computer Graphics, Vol. 21, No. 4, '87.

[18.] Barzel & Barr, "A Modeling System Based on Dynamic Constraints", Computer Graphics, Vol. 22, No. 4, '88.

[19.] Witkin and Kass "Spacetime Constraints", Computer Graphics, Vol. 22, No. 4, '88.

[20.] Platt and Barr, "Constraint Methods for Flexible Objects", Computer Graphics, Vol. 22, No. 4, '88.

[21.] Girard & Maciejewski, "Computational Modeling for Computer Animation of Legged Figures", Computer Graphics, Vol. 19, No. 3, '85.

[22.] Girard, Michael, "Interactive Design of 3-D Computer Animated Legged Animal Motion", Chapel Hill Interactive Computer Graphics Interface Workshop, '86.

[23.] Badler, Manoochehri, Walters, "Articulated Figure Positioning by Multiple Constraints", IEEE CG&A, June, '87

[24.] Armstrong & Green, "The Dynamics of Articulated Rigid Bodies for purposes of Animation", Proc. Graphics Interface 85, '85.

[25.] Wilhelms, Jane, "Using Dynamic Analysis for Realistic Animation of Articulated Bodies", IEEE CG&A, June, '87.

[26.] Isaacs and Cohen, "Controlling Dynamic Simulation with Kinematic Constraints, Behavioral Functions and Inverse Dynamics", Computer Graphics, Vol. 21, No. 4, July, '87.

[27.] Zeltzer, David, "Representation and Control of Three dimensional Computer Animated Figures", Ph.D.. Dissertation, The Ohio State University, '84.

[28.] Amkraut, Susan, "Flock: A Behavioral Model for Computer Animation", M.A. Thesis, The Ohio State University, '89.

[29.] Reynolds, Craig, "Flocks, herds and Schools; A Distributed Behavioral Model", Computer Graphics, '87.

[30.] Burtnyk & Wein, "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation", CACM, Vol. 19., No. 10, Oct. '76.

[31.] Parent, Richard, "A System for Generating 3-Dimensional Data for Computer Graphics", Ph.D. Dissertation, The Ohio State University, '77.

[32.] Magnenat-Thalmann & Thalmann, "The Direction of Synthetic Actors in the Film Rendezvous a Montreal", IEEE CG&A, Dec., '87.

[33.] Donkin, John, personal communication '88.

[34.] Denavit & Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", J Appl Mech, Vol. 23, '55.

[35.] Whitney, "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", Journal of Dynamic systems, Measurement, and Control, Dec., '72.

[36.] Klein, Huang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators", IEEE Transactions on Systems, Man, and Cybernetics, Vol SMC-132, No. 3, March/April '83.

[37.] Chadwick and Parent, "Critter Construction: Developing Characters for Computer Animation', Proceedings Pixim, '88.

[38.] Steindler, Arthur, "Kinesiology of the Human Body", Charles C. Thomas Publisher, Springfield Illinois, '55.

[39.] Burstein, A., Frankel, V., "The Viscoelastic Properties of Some Biological Materials", Materials in Biomedical Engineering Vol. 146, Article 1, p. 158-165, Annals of the New York Academy of Sciences, January 1968.

[40.] Schneider D., et al.,"In Vitro Biaxial Stress-Strain Response of Human Skin", Arch Otolaryngol Vol 110, p. 329-333, May 1984

[41.] Fung, Y. C., Biomechanics: Mechanical Properties of Living Tissues, Springer-Verlag, New York, 1981.

[42.] Lundin, R. Ruminations of a Model Maker IEEE Computer Graphics and Applications 7(5):3-5 May 1987

[43.] Waters, K. A Muscle Model for Animating Three-Dimensional Facial Expression Computer Graphics 21(4):17-24 (Siggraph Proceedings) July 1987

[44.] Weil, J. The Synthesis of Cloth Objects Computer Graphics 20(4):49-54 (Siggraph Proceedings) August 1986