Animation and Collisions between Complex Deformable Bodies

Marie-Paule Gascuel
Anne Verroust
LIENS, Ecole Normale Supérieure
CNRS URA 1327, 45 Rue d'Ulm,
75230 Paris CEDEX 05, France

Claude Puech
Stanford University/Hewlett-Packard Laboratories*

Abstract

We propose an integrated set of methods for designing and animating bodies whose deformable flesh coats an articulated skeleton. These methods provide automatic positioning of the "skin" after movements and automatic deformation after collisions with rigid or deformable objects. The response to collisions includes a feedback from the flesh to the skeleton, whose movement is adequately modified. Moreover, coating the skeleton gives an easy solution to closed loop collisions and angle constraints control. Our model is modular and gives some high level control to the user.

Keywords: Animation, Deformation, Dynamics, Simulation, Collision detection, Collision response.

1 Introduction

To convey the story of an animated film, it is often simpler to forget about the surfaces surrounding the objects, only keeping in memory the structure of their articulations. Key-frame and dynamic methods [2] have been proposed to control the movements and deformations of such articulated "skeletons". To obtain a more aesthetic result, one has to coat these skeletons with a moving "skin" which is deformed adequately at each animation step; automatic methods can be used to avoid the frame by frame manual intervention of the animator. An even more challenging problem is to create a skin able to detect collisions and to deform properly during and after the impacts (taking into account the flexibility of the "flesh" it covers). In addition, the movement of the skeleton must be modified by the collisions.

Several methods have already been proposed for the automatic computation of the skin surface from the movement of an articulated skeleton:

Chadwick et al. [5] implemented a model with dynamic and geometric layers to simulate various types of muscles and fatty tissues for figure animation. This model provided high level control to the user, but the collision problems were not considered. Thus, even for a dynamic skeleton able to deal with interactions, nothing was devised to deform the skin according to the collisions detected. Moreover, the skin, deformable muscles and fatty tissues were not designed to treat collisions by themselves: They were created with geometric intermediate structures (FFD for free form deformation boxes [16]), making their use quite difficult to detect or compute the response to collisions. In addition, detecting interactions at the flesh level would have induced unrealistic overall behavior (the collision having no effect on the precalculated movement of the skeleton).

Gourret et al. [8] simulated the contact between a hand and a deformable ball through the elasticity equations for a set of finite elements. At each time step, the shape of the flesh was computed from a position of the skeleton specified by the user, the equilibrium position for the finite elements being found after a sequence of oscillations. The computation of the response forces applied by the flesh onto the bones provided a detection of the unrealistic positions of the skeleton. This method gave very good results in the specific situation presented but does not seem well adapted to general animation purposes: The computations involved are important, a very complex database is required and the dynamic equations which are used would not be valid for general collisions.

In this paper, we propose a new model for "complex deformable bodies", e.g. for objects whose deformable flesh coats an articulated rigid skeleton. One of the main advantages of our model is its ability to deal with the difficult problems of collision detection and response. A good collision detection must involve the flesh level, while the response must modify the movement of the skeleton. Therefore, we use a simultaneous dynamic animation of these two structures, which enables them to communicate at each animation step.

The deformable model employed for the flesh must be designed to deal easily with all kinds of interactions. In Section 2, we review the main models of deformable material which have been proposed in the past few years and motivate our need of a new layered structure. Then, we describe our deformable model, whose main feature is the use of two different layers to model independently the local stiffness and the propagation of deformations.

---

*On sabbatical leave from Ecole Normale Supérieure during academic year 1990-1991.

This model yields a fast computation of the deformations during the animation, and an easy evaluation of the response forces due to collisions.

In order to design complex deformable objects easily, we present in Section 3 the "automatic coating of skeletons" method: The automatic coating creates the deformable flesh and the skin surface of the bodies from descriptions of their skeletons at rest. We describe the angle constraints control provided by the method.

In Section 4, we present the animation module associated with our model: A very efficient simulative algorithm is chosen for dynamic animation of the articulated rigid layer of the model. Then, we give the successive steps of the whole animation process.

Section 5 contains comments on animations designed with the system: Our approach provides good automatic positioning of the skin after movements, angle constraint control, automatic deformation after collisions, and feedback from the skin to the skeleton which gives adequate response to collisions.

In Section 6 we conclude and discuss work in progress.

2 A new layered deformable model

2.1 Deformable models

The behavior of very complex objects such as deformable models is hard to control with key-frame systems. In the past few years, several types of "active models" built on dynamic equations have been proposed.

A first class of deformable models is based on the principles of mathematical physics. In [18], the models are defined by the non-linear elasticity equations for continuous bodies. Realistic animations are created by numerically solving these equations (using finite elements techniques and discretizations over time). But even for homogeneous and isotropic objects the computations involved are important and the equations tend to be poorly conditioned when rigidity is increased. Moreover, the control of such models is not easy. Various methods have been proposed to cope with some of the limitations of the first model: In [15], constraints techniques offer a better control upon trajectories and deformations. In [19], an hybrid formulation and the use of linear elasticity to model the distortions of the objects enable to simulate more rigid objects with well conditioned discrete equations. Inelastic deformations are introduced in [17].

At the same time, a class of layered deformable models appeared. These models, which do not obey very realistic physical laws, integrate geometric and discrete dynamic components (such as springs and dampers) [11]. They provide easier control, because of the independent parameters of the different layers. Therefore, complex composite objects can be animated in reasonable time, with good visual effects. For instance, snakes' movements are modeled in [13] by covering masses linked by springs with a geometric skin. Fatty tissues are simulated in [5] by linking the control vertex of an FFD with viscously damped springs.

Collision detection and response: Automatic treatment of collisions is an important topic to understand the usefulness of the models for animation purposes.

This problem is especially tricky for models based on elasticity equations for continuous objects: The elasticity theory only describes the behavior of deformable material during small oscillations around equilibrium states. No answer is given to the collisions problem. Thus, only artificial solutions can be used: The authors of [18] avoid interpenetrations with still rigid objects of the scene by surrounding them with force fields. The constraint method of [15] prevents collisions between deformable objects and rigid polyhedral solids by automatic computation of new external forces which are exerted on the finite elements.

For modular models composed of mechanical and geometric independent layers, a precise collision detection is sometimes hard to achieve [11,13]. The response is simulated in [11] by temporarily introducing a spring-like connection between the colliding deformable objects. In [13], the normal component of a mass point velocity is inverted when it collides the floor.

More general collision detection and response methods are provided in [14]: The surfaces of deformable objects are discretized into triangles. Interpenetrations are detected by computing intersections between elementary trajectories of surface points and triangles not containing these points. The response is simulated by introducing between the objects a temporary spring whose stiffness decreases after the impact.

2.2 Our deformable material

Our aim is to create a model for deformable material enabling easy collision detection and convenient computation of response forces. In order to animate in reasonable time several interacting complex objects, we are looking for good visual effects rather than realistic simulation of existing materials. As emphasized in [5], a layered construction is a very efficient tool for creating complex motions in animation. It is also very convenient for the design of the objects: The inter-relations between the layers are managed by the system so that the user has access to easier and more intuitive parameters.

In [11,13,5], the mechanicals components are created with elementary masses linked by damped-springs (*cf.* Figure 1). The dynamic animation is made by numerically integrating through time the differential equations governing the movements of each elementary mass. This kind of model does not give any high level control upon the geometry of the deformations. Indeed, in the real world some flexible objects preserve their volume during deformations (modeling clay for instance), while others (a soft cushion...) deform with constant surface area. This kind of global constraints could not be specified by controlling the stiffness of the springs in Figure 1.

Our deformable model [6], uses a layered structure in order to model (and control) independently two important characteristics of the material:

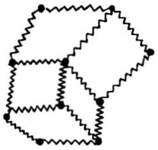


Figure 1: Classical mass-spring systems

- The radial stiffness at sample points of the surface,
- The way the deformations propagate when external forces are applied to these sample points.

We use a third layer for the smooth "skin" which models the external surface of the object.

The radial stiffness layer: A deformable object is modeled by cylinders of deformable material attached in a star-shaped way around a rigid "kernel". By varying the repartition, stiffness and weight of those cylinders, non-homogeneous and non-isotropic objects can be created as well. The axes of the cylinders are fixed w.r.t. the rigid kernel and give the "main deformation directions" at the surface of the object. Each cylinder modeled by a damped-spring, controls the movement of a sample point at the surface of the object (see Figure 2).

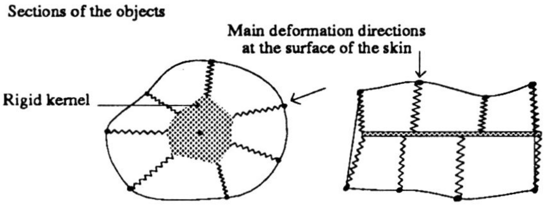


Figure 2: First layer of the model

To compute the displacement of those sample points through time, we do not need to apply any finite differences method: Because of the fixed axes we use for the cylinders, the oscillations of the points are given by closed formulas in the coordinate system of the rigid kernel. Let k be the stiffness of the exponentially-damped spring S which simulates a cylinder of mass m, l_0 be its initial length, and γ its damping coefficient. If, at time t_1, S is expanded or contracted in such a way that its length becomes l_1, and if there is no external action between t_1 and t, then, at time t, the length of S is:

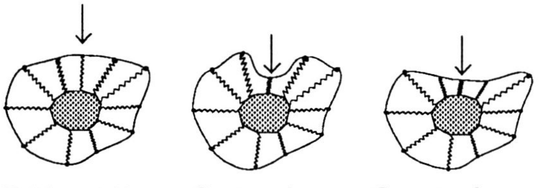
$$l_t = l_0 + (l_0 - l_1)\ e^{-(t-t_1)\gamma}\ \cos(m\ k\ (t - t_1))$$

The sample point controlled by the cylinder is located along the fixed axis, at distance l_t from the rigid kernel[1].

This first layer only models a restricted class of objects since the deformation of one cylinder has no effect on its neighbors. We add a second layer in order to give high level control on the propagation of deformations.

Modeling the propagation of deformations: Suppose that a force $\vec{F}$ is applied on one of the cylinders. The propagation layer provides with precise control on the relationship between the local displacement of the associated sample point P and the overall deformation of the object. The users specifies independently two characteristics of the deformation:

1. The "propagation zone" associated to a cylinder is defined by selecting some "neighboring" cylinders to which the deformation propagates. Such a control can be used for marking the difference between the deformations of a balloon and a sponge[2].
2. The "propagation mode" gives the relationship between the displacements of point P and of its neighbors. We consider two kinds of criteria:
    - Geometric propagation mode: Some flexible materials deform with constant volume, and some with constant surface area (see Figure 3). Most of them are intermediates between these two extreme models. So, we compute approximations of the results given by these models[3], and apply a linear combination of the results to the sample points of the propagation zone.



Undeformed object | Constant volume deformation | Constant surface area deformation

Figure 3: Propagation modes based on geometric criteria

- Intuitive propagation mode: The radial force used to move P (multiplied by a dissipative coefficient k, $|k| < 1$) is shared between the cylinders of the propagation zone. They use these amounts of forces to deform[4]. This propagation mode is not founded on any mechanical law, but provides a fast computation of the propagation, and good visual effects.

The skin layer of the model: The skin must stay very close to the deformable cylinders; we also need easy and efficient calculations. Thus we model the skin as a spline surface whose control points are exactly the sample points associated with the cylinders: A deformation of the two first layers will immediately translate into a deformation of the skin. In order to change the aspect of the object, various type of splines can be used. B-splines produce very smooth and regular deformable objects. Splines with tension [3,10] could provide local precise tuning of deformations.

---

[1] The damped-springs could be replaced by the visco-plastic elements used in [17]. Then, l_0 would change through deformations, leading to materials with plastic behavior.

[2] The propagation zone can be computed from a default value by taking into account the direction of $\vec{F}$

[3] Using the envelope of the sample points to approximate the surface of the object.

[4] Constant volume deformations are imitated using $k < 0$, constant surface area deformations correspond to $k > 0$.

2.3 Collision detection and response

The layered structure gives an elegant solution to the problem of detecting and giving an adequate response to collisions. The deformable body can collide with rigid or flexible objects, dynamic or static. Response forces are calculated without the artificial introduction of new mechanical elements. As usual, we use three different phases:

1. Detecting collisions,
2. Modeling the contact between the two objects,
3. Computing the response.

Collision detection: The method used in [14] consisted in testing the intersections between each elementary trajectory of control points between time t and time $t + dt$ and the trajectory of each triangular facet not including this point. Each of these tests required solving a polynomial equation of degree 5 by binary search to compute the time t of the possible impact.

In contrast, we do not need to know the exact time of the collisions (we do not go back to the moment of the impact when we compute the response). So, a detection of the interpenetrations at a fixed time (and then in fixed positions) is sufficient. This detection requires two steps:

- We first use a bounding box/sphere method to get rid quickly of most non-intersecting cases;
- Then, we test whether the sample points associated to each cylinder of deformable material penetrate inside another cylinder (our method can report self-intersections). To do so, we approximate the surface of the second object by the envelope of its sample points, and compute intersections between triangles of this envelope and spring-segments linking a sample point of the first object to its rigid kernel.

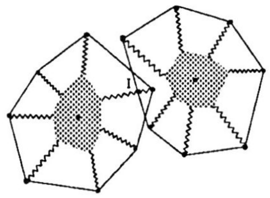


Figure 4: Collision detection between deformable objects

Modeling contact: When a collision is detected, we model the contact (and prevent the interpenetrations) by deforming the objects according to their local stiffnesses and propagation modes. This processing is done easily because the sample points always move along fixed axes.

If the collision occurs between a deformable object and a rigid one, the sample point P which has penetrated inside the other object is positioned at the intersection point I. Then, the propagation mode is used to compute the resulting shape of the object.

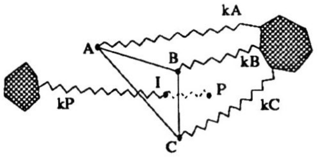


Figure 5: Computation of the new position of P

If both objects are deformable, we must find a deformation which takes into account the relative stiffness of the two objects near the contact zone. Let k_P be the stiffness of the cylinder associated with P, k_A, k_B, k_C the stiffnesses of the cylinders defining facet ABC and (α, β, γ) the barycentric coordinates of I w.r.t. (A, B, C) (see Figure 5). We define the relative stiffness k of the two objects in the contact area by the formula:

$$k = \frac{k_P}{k_P + (\alpha k_A + \beta k_B + \gamma k_C)}$$

We position P at $M_c = kP + (1 - k)I$ and move A, B and C, according to the values of k_A, k_B and k_C, in such a way that M_c becomes a point included in the facet ABC. Then the deformations are propagated according to the propagation layers of the two objects. At the end of this phase, $P = M_c$ is a contact point between the two objects.

Response to a collision: The use of impulses as in the case of rigid objects is not appropriate for collisions between deformable objects:

- contacts often last for several time steps,
- the energy produced during collisions is first stored by the objects as internal deformation energy, and then restored in the form of contact forces (when the objects push each other away while trying to recover their initial shapes).

Computing contact forces seems appropriate, and is easy with our model: At the end of the modeling contact phase, contact points have appeared between the two colliding objects. The internal deformation energy of the cylinders associated to those points is known (the cylinders are modeled with damped-springs). Thus, the response forces exerted at contact points by one of the objects are easy to compute. Action-reaction principle states that the other object exerts a symmetric force on the first one. These response forces are added to the external forces exerted on the objects.

A few remarks: This computation can be used even if one of the objects is rigid. If the objects collide in such a way that their rigid kernel collide, an impulse has to be used as a result. The velocities and inertia of the objects at collision time although not used explicitly, have an effect on the collision through the importance of the deformation.

As in [8], we need to prevent deep interpenetrations, which produce too big forces. Thus, a test is made on the sum of the kinetics energy of the touching objects: If it got too big during collision, we go back in time and choose a shorter time interval. This process uses a dissipating coefficient for the energy, computed from the duration of the collision and the amount of deformation.

3 The automatic coating of skeletons

As depicted in Figure 6, a complex deformable body is made of three parts:

- A rigid articulated skeleton (tree of rigid parallelepipeds called links, each link other than the root being connected to its parent by a hinge).
- A set of deformable elements associated to the links, created with the deformable material of Section 2.2,
- A "skin" which covers all the deformable elements.

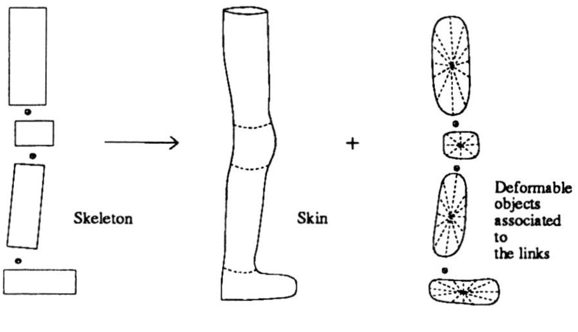


Figure 6: Automatic coating of a skeleton

Although the set of deformable elements associated with the links and the "skin" can be produced interactively by the user, they can often be created more easily by "coating" the skeleton. We use an extension of the "automatic fleshing of skeletons" method [7].

3.1 Automatic fleshing of skeletons

This technique uses two high level tools, welding and pinching, to automatically create an object's "skin" from a geometric description of its skeleton. During a breadth-first traversal of the tree structure representing the skeleton: A spline tube created by sweeping is associated to each link and "welded" to the surface corresponding to its parent; if necessary, it is "pinched" according to its number of sons in the structure (in order to create multiple junctions of tubular surfaces). The resulting skin is as regular as the initial tubes (class C^2 if B-splines are used), except near the "pinching points" where only a C^0 continuity is achieved.

3.2 Basic coating method

We extend the fleshing method: A deformable element is created for each link each time a skin tube is constructed. The sample points of its deformable cylinders are the control points defining the skin tube. Thus a deformation of the element will immediately translate into a deformation of the skin.

In order to create deformable elements with desirable properties for the animation of complex deformable bodies, we make the following choices:

- We close the surface of the deformable elements by adding triangles with one vertex located at an end-point of the link (we need closed volumes whose surface has to be triangulated for collision detection). See Figure 7.

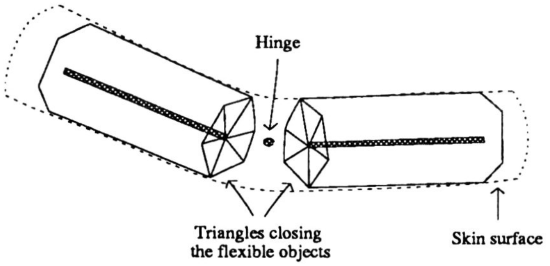


Figure 7: Addition of triangular facets

- We attach the internal end-points of the springs modeling the deformable cylinders to the center of gravity of the link. This choice guaranties fast computations and is also well suited to angle constraints control: As shown on Figure 8 if the angle at the articulation point becomes too small, a collision will be detected between the deformable elements associated to the two links connected to this hinge and the forces produced will prevent the angle from becoming smaller.

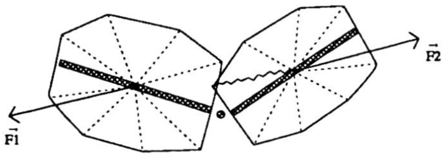


Figure 8: Angle constraints controlled by a collision

When the deformable elements are created the lengths of the springs (*i.e.* their lengths at rest) associated with the deformable cylinders are computed and their axes (which are fixed w.r.t. the link) are determined.

Default values are given to the other parameters such as stiffnesses or damping coefficients. Four neighboring cylinders are associated to each cylinder to define a propagation zone, and a propagation mode is chosen. The user can modify interactively each of these parameters. The automatic coating is mainly intended to give default values in order to produce very quickly an "initial" object which can be tuned interactively.

3.3 Skeletons with branching points

The basic coating method proves to be sufficient for coating chain-like articulated objects. The fleshing method of [7] produces multiple branching through pinching. This primitive is not adapted to the skin creation needed here. As shown in Figure 9 when coating the skeleton, we choose to decompose the underlying tree structure into chains which are coated separately. When necessary, the skin surface is extended and can locally penetrate into another part of the skin. But the deformable elements associated with links must not penetrate one another because no collision should be detected "at rest".

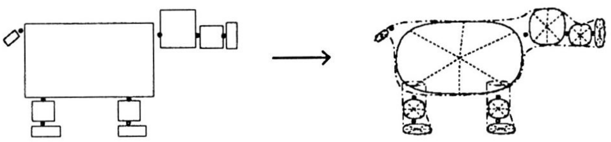


Figure 9: Coating of a skeleton with branching points

Precise angle constraint control

The constraints imposed on angles at articulation points can be tuned very precisely by the user. This is not a mere control of the thickness of neighboring elements:

- As shown on Figure 10, asymmetric shape of the deformable elements will produce asymmetric constraints on the angles. We can obtain such shapes automatically while positioning the skeleton before coating: we put each link in a median position w.r.t. the aimed extreme positions. We can also tune the shapes of the deformable elements after coating by moving some of the sample points.
- The shape of the deformable elements being given (and thus the thickness of the object), the articulations "stiffness" can be controlled by changing interactively the spacing of neighboring deformable elements: If the deformable objects are farther located, the articulation can be bent more easily.

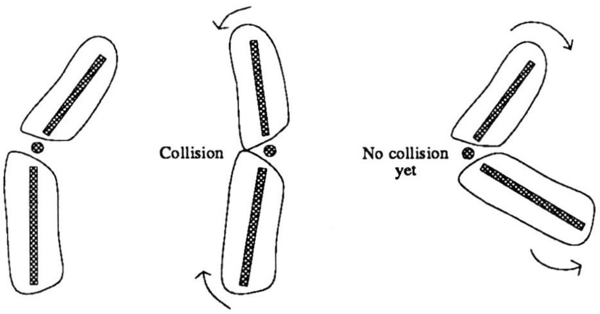


Figure 10: Asymmetric angle constraints

4 Animating complex deformable objects

4.1 Algorithm used for the skeleton

In order to animate a complex deformable body, the skeleton animation module must take into account the response forces and torques given by the "flesh level". In addition, we want to obtain "interactive" animations. So, among the methods using dynamics to animate articulated bodies [1,21,4,9,12], we privilege the efficiency of the computations. The deformable flesh of the objects will avoid closed loops problems, control the angle constraints, and detect collisions. Thus one of the fastest dynamic method, *i.e.* Armstrong and Green's one [1], in spite of its limitations (it allows only rotations between a link and its parent) is for us a good choice to animate the skeleton (its computation time grows linearly with the number of links). So, our skeletons are such that the root has six degrees of freedom and the other links have three degrees of freedom (rotation around the hinge joining them to their parents).

Animation module

Our animation algorithm for complex deformable bodies can be decomposed as follows[5]:

(for the first iteration, the initial positions are known and the algorithm begins at step 2; for the next iterations, the values obtained at step 5 of the previous iteration are used in step 1)

1. Update the transformation matrices, positions of the links, accelerations and velocities.
2. Update the positions of all the deformable cylinders of the model (their length can vary because of oscillations) without considering any possible collision.
3. Test for collisions between deformable elements and rigid objects[6]. For each such collision, compute and store the resulting forces and torques. Compute the new shape of the deformed object.
4. Test for collisions between deformable elements[6]. For each such collision, compute and store the resulting forces and torques. Compute the new shapes of the deformed objects (once a spring has been deformed it will be not be modified until the end of the current iteration in order to prevent cycles).
5. Compute the acceleration of the root through a traversal of the skeleton using the forces and torques due to collisions and external forces and torques.

At the end of step 4, all the forces and torques due to collisions detected at the skin level have been stored. They are used at step 5 to compute their effect on the movement of the skeleton.

As noticed at the end of Section 2, the time interval is "adaptive". Kinetics energy is controlled and the test can require going back in time.

5 A sample of animations

A new kind of bowling: In Figure 11, five "articulated skittles", each one composed of four deformable elements, are stroked by a chain, whose extremity is fixed, and which is thrown with an initial rotation velocity. The first four skittles, touched lightly by the chain, roll symmetrically sideway (see $t = 11$, $t = 17$, etc.). The red skittle is crashed with a head-on collision, and stops the rotation movement of the chain. The lower part of the red skittle is composed of two thick deformable elements, coating an articulated skeleton. The significant angle constraints between these two elements explain the crumpling which can be observed at $t = 23$ and $t = 27$.

---

[5] We use Armstrong and Green's resolution method: Steps 1. to 4. correspond to the top-down and step 5. corresponds to the bottom-up traversal of the tree of links.

[6] Using a hierarchical bounding box/sphere method.

Colliding octopus: The "octopus" of Figure 12 has six tentacles, each of them composed of four deformable elements, and its head includes two deformable elements. The octopus is first thrown up in the air and external forces are applied to the end of the tentacles to make them gather. Collisions between tentacles occur ($t = 80$) and the tentacles react. The gravity force tends to make the whole body fall.

6 Conclusion

We have proposed an integrated set of methods for designing and animating complex deformable bodies. The proposed methods achieve:

- Good automatic positioning of the "skin" after movements. Results similar to the ones of [5] can be obtained, and the parameters used here are not purely geometrical: Collisions can be detected, and the deformations are propagated in an easily controlled way.
- Feedback from the skin to the skeleton: The skeleton receives information from the outside world through the skin layer which computes response forces and torques to be applied to the links of the skeleton.
- General collision detection: Dynamic and static objects, deformable and rigid objects. Adequate response without introducing new mechanical elements.
- Easy solution to the closed loops problem through the use of the skin level: As the skin prevents loops on the skeleton, a more efficient animation method can be used for the skeleton.
- Angle constraints control.
- Modularity and high level control from the user.

The system proves to be efficient. Interactive animations can easily be computed on standard graphics workstations such as the Personal Iris on which the system is implemented.

Work in progress includes attempts (using methods inspired by the ones in [15,4,22]) to incorporate some control on the trajectories and intermediate positions of the bodies, in order to give an easier set of parameters to the user. Another extension would be to generalize the method to complex objects bounded by geometric constraints, such as the objects with pinching points of [7].

Acknowledgements: We wish to thank J.D. Gascuel for his multiple helps during the implementation and J.P. Thirion who provided us with a very quick ray-tracer.

References

[1] W. Armstrong and M. Green. The dynamics of articulated rigid bodies for purposes of animation. *The Visual Computer*, 1:231–240, 1985.

[2] N.L. Badler, B.A. Barsky, and D. Zeltzer. *Making Them Move: Mechanics, Control and Animation of Articulated Figures*. Morgan Kauffmann, 1990.

[3] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Palo Alto, California, 1987.

[4] R. Barzel and A. Barr. A modeling system based on dynamic constraints. *Computer Graphics*, 22(4):179–188, August 1988.

[5] J.E. Chadwick, D.R. Haumann, and R.E. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3):243–252, July 1989.

[6] M.P. Gascuel. Déformations de surfaces complexes : techniques de haut niveau pour la modélisation et l'animation. Thèse d'université, Université de Paris-Sud, Centre d'Orsay, October 1990.

[7] M.P. Gascuel. Welding and pinching spline surfaces: New methods for interactive creation of complex objects and automatic fleshing of skeletons. In *Graphics Interface'89*, June 1989.

[8] J.P. Gourret, N. Magnenat Thalmann, and D. Thalmann. Simulation of object and human skin deformations in a grasping task. *Computer Graphics*, 23(3):21–29, July 1989.

[9] P.M. Isaacs and M.F. Cohen. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. *Computer Graphics*, 21(4):215–224, July 1987.

[10] D.H.U. Kochanek and R.H. Bartels. Interpolating splines whith local tension, continuity, and bias control. *Computer Graphics*, 18(3):33–41, 1984.

[11] A. Luciani and C. Cadoz. Utilisation de modèles mécaniques et géométriques pour la synthèse et le contrôle d'images animées. In *Deuxième Colloque Image, CESTA*, Nice, April 1986.

[12] M. McKenna and D. Zeltzer. Dynamic simulation of autonomous legged motions. *Computer Graphics*, 24(4):29–38, August 1990.

[13] G. Miller. The motion dynamics of snakes and worms. *Computer Graphics*, 22(4):169–177, August 1988.

[14] M. Moore and J. Wilhelms. Collision detection and response for computer animation. *Computer Graphics*, 22(4):289–298, August 1988.

[15] J.C. Platt and A.H. Barr. Constraint methods for flexible models. *Computer Graphics*, 22(4):279–288, 1988.

[16] T.W. Sedeberg and S.R. Parry. Free-form deformations of solid geometric models. *Computer Graphics*, 20(4):151–160, 1986.

[17] D. Terzopoulos and K. Fleischer. Modeling inelastic deformations: Viscoelasticity, plasticity, fracture. *Computer Graphics*, 22(4):269–278, August 1988.

[18] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205–214, July 1987.

[19] D. Terzopoulos and A. Witkin. Physically based model with rigid and deformable components. *IEEE Computer Graphics and Applications*, 8-6:41–51, November 1988.

[20] A. Verroust. Etude de problèmes liés à la définition, la visualisation et l'animation d'objets complexes en Informatique graphique. Thèse d'Etat, Université PARIS-SUD, Orsay, December 1990.

[21] J. Wilhelms and B. Barsky. Using dynamic analysis to animate articulated bodies as humans and robots. In *Graphics Interface 85*, pages 97–104, Montréal, 1985.

[22] A. Witkin, M. Gleicher, and W. Welch. Interactive dynamics. *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, 24(2):11–21, March 1990.

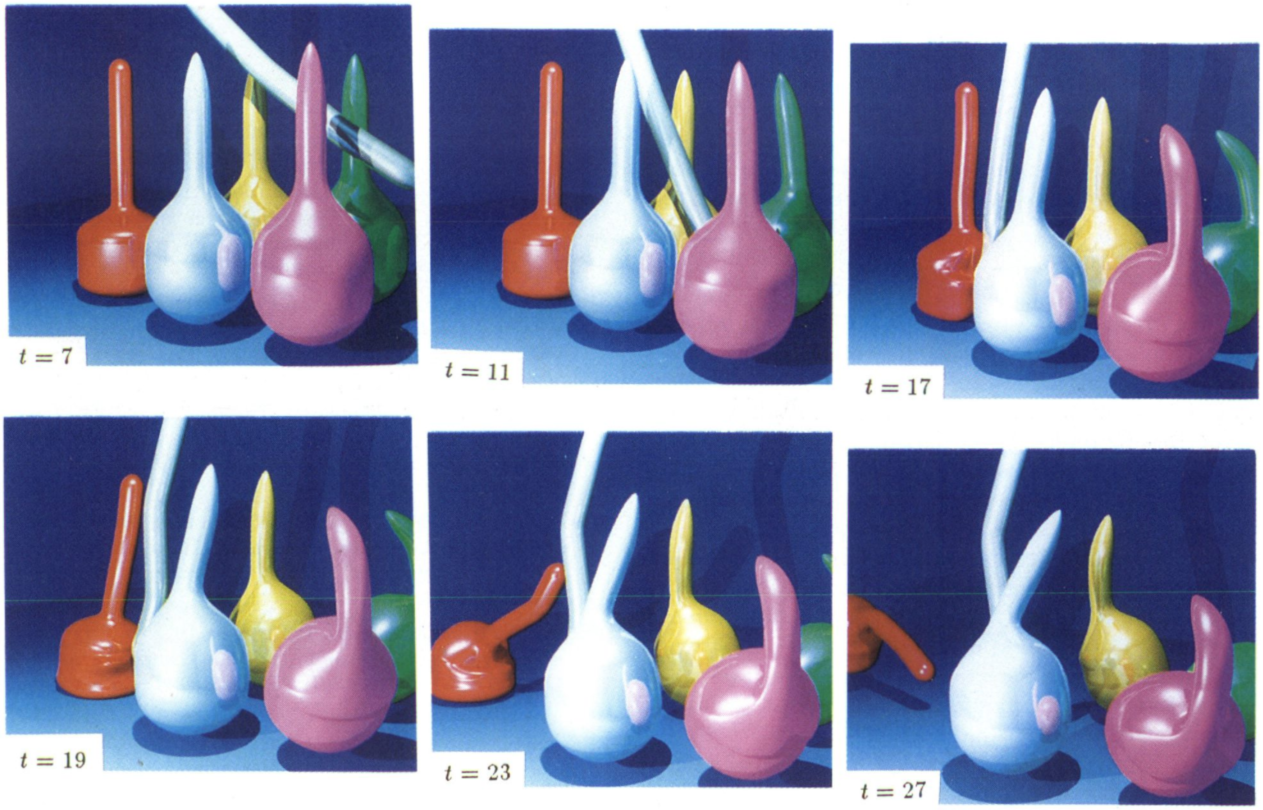
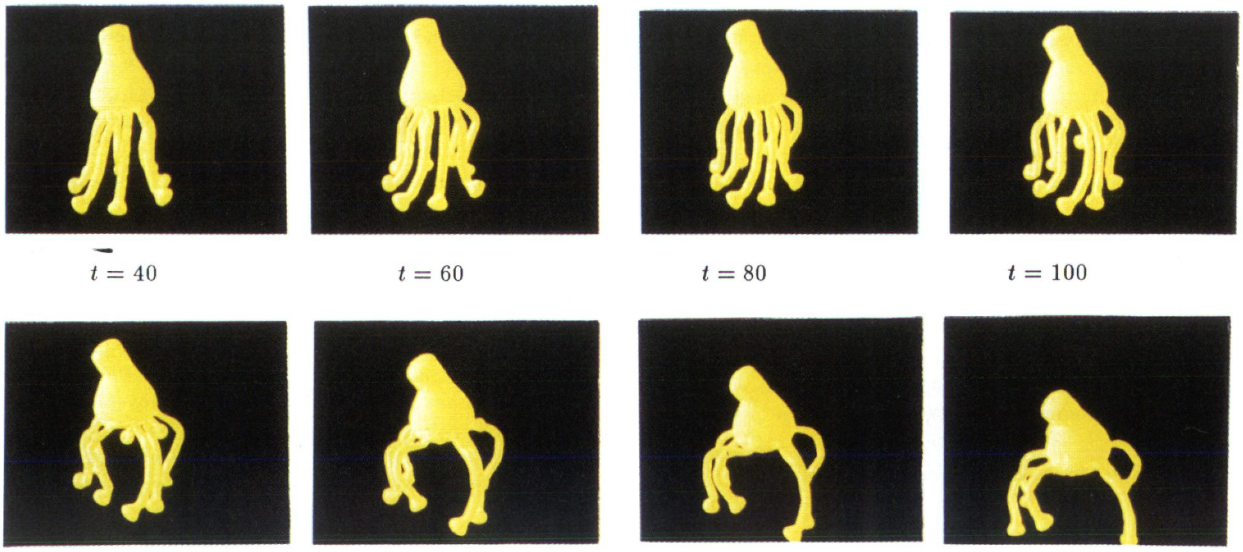


Figure 11: A new kind of bowling


$t = 40$ $t = 60$ $t = 80$ $t = 100$

$t = 120$ $t = 140$ $t = 160$ $t = 180$

Figure 12: Colliding octopus