Highly deformable material for animation and collision processing

Mathieu Desbrun, Marie-Paule Cani

Highly Deformable Material for Animation and Collision Processing

Mathieu Desbrun
Marie-Paule Gascuel

iMAGIS / IMAG[1] - INRIA
BP 53, F-38041 Grenoble cedex 09, France

email: Mathieu.Desbrun@imag.fr

Abstract

This paper presents a method for modeling and animating highly deformable material such as clay, dough or mud. The animated objects are inelastic: they absorb deformations and seldom come back to their initial shape. Their surfaces are smooth and fit with the objects they are in contact with. Moreover, their topology may change during animation sequences. In particular, they can break into pieces.

The new model presented here is an hybrid one. Each object is composed of an elastic coating over a discrete inelastic layer. The elastic material is defined by an implicit formulation that is particularly convenient for modeling objects whose topology changes over time. Well-suited to automatic collision detection and response, the elastic layer generates and maintains exact contact surfaces between interacting objects. Response forces computed along contact surfaces are transmitted to the internal structure that deforms accordingly. The whole simulation is produced at interactive rates.

Keywords: animation, simulation, deformation, soft material, implicit surface, adaptive sampling, collision detection, collision response.

1 Introduction

In traditional animation systems, specifying motion and successive shapes of deformable material interacting with the simulated world requires a great amount of specialized knowledge from the animator. Recently, several physically-based models have been proposed for the automatic computation of motion and deformations. Most of them are restricted to elastic material, which comes back to its rest shape after any deformation.

Highly deformable objects that can simulate various behaviors from quasi-solid to quasi-liquid, such as those made of mud, dough or clay, are difficult to animate with current tools. In particular, maintaining well-defined and smooth surfaces on objects whose topology may change over time remains a challenge.

[1] IMAG is a Research institute affiliated with CNRS, Institut National Polytechnique de Grenoble, and Université Joseph Fourier.

This paper presents a new method combining deformations of surface and topologic changes. Based on an implicit formulation, our model maintains exact contact surfaces between inelastic objects.

Previous Approaches

A few models for inelastic deformations have been developed in Computer Graphics. All of them belong to the class of nodal approaches: deformations are expressed by displacement of elementary nodes representing mass points or sample points inside the material.

Some of the approaches are derived from elastic models based on discretized continuous equations that are extended for modeling inelastic deformations. A layered model [TF88] represents inelastic material as an elastic component at rest with respect to a reference component used for computing motion. Inelasticity is modeled by allowing the reference component to progressively absorb some of the deformations of the elastic layer. Here, topology changes are limited. Fractures can be modeled, but only at predefined fracture points. Another system [TPF89] models blocks of flexible material that can be heated until they melt. A lattice of mass points linked by springs of variable stiffnesses is used at the solid state, while particle interaction laws govern motion of mass points that have been disconnected from the network (each spring disappears at a certain heat). The main drawback of these hybrid models is the non-unified view of the different behaviors. Specific laws are developed for each behavior to model, but there is no general method. For instance, freezing the material back to a solid state would be quite difficult using the last approach.

Another solution is the direct use of discrete elementary masses, each of them being animated separately according to a set of interaction laws with the others [LJF+91]. These physically-based "particle systems" give an unified view of multiple behaviors, from elasticity to plasticity and from quite stiff objects to quasi-liquid states [Ton91, LJR+91]. In addition, fractures and other tolopogical changes can be generated very easily.

Methods for collision detection and response have to be provided together with a deformable model. Detecting interpenetrations between objects represented by lattices of elementary nodes basically requires $O(n^2)$ steps, where n is the number of nodes. Response to collisions between deformable solids and rigid objects is often computed from penalty methods [TF88, TPF89, Ton91]. A force proportional to the amount of interpenetration is applied to the nodes that have penetrated a rigid object. When all the objects of the scene are represented by sets of elementary masses [LJF+91, LJR+91], interactions between objects are just a particular kind of interaction between particles: a repulsion force is produced between two mass points from different objects that become too close (otherwise, this force is zero). Computing interaction forces between n mass points from different objects normally takes $O(n^2)$ steps, but the use of adequate data structures can reduce this complexity to $O(n \log(n))$ [Hou92].

None of these models generates any contact surfaces between highly deformable objects. This is mainly due to the fact there is no surface defined. An object is just a set of nodes or of mass points, and there is no information

telling which points define "the object surface" after it undergoes topological changes. However, displaying objects at a macroscopic level rather than just as a set of points is often needed. Some of the authors display implicit surfaces generated by the mass points [TPF89, Ton91]. But as these surfaces are not considered for collision detection, this can result in visual anomalies such as local interpenetrations and bouncing before contact.

Overview

This paper presents a new model for highly deformable material that provides exact surface contact during interactions. We use an hybrid representation enabling a coherent use of internal and external structures. Each object is composed of discrete internal elements linked by interaction laws, and coated with some elastic material. Based on an implicit formulation particularly convenient for modeling topological changes, the elastic flesh defines a soft surface for each block of material. This surface is used for detecting collisions (which can be done efficiently thanks to the implicit inside-outside functions), computing deformed shapes, and integrating response forces. These forces are then propagated into the internal structure, producing inelastic deformations and possibly fractures, in the subsequent time steps.

Section 2 briefly reviews the model for implicit elastic material introduced in [Gas93]. Section 3 explains how to extend this model for defining an elastic coating around a discrete layer. In particular, volume variations of the implicit flesh must be carefully controlled during a progressive compression of the internal structure. This leads to coherent choices for the internal and external parameters. Section 4 details our method for collision detection and response for the hybrid model. Computing sample points for collision processing is a difficult task, since the topology of the objects changes over time. Therefore, we propose an efficient adaptive sampling algorithm. Section 5 presents our results and discusses work in progress.

2 Implicit Elastic Material

This section briefly describes the continuous elastic model developed in [Gas93]. Specifically designed for precise contact processing, this model uses isopotential implicit surfaces generated by "skeletons" to model both the geometry and the physical properties of the objects.

Implicit Surfaces

An implicit surface S generated by a set of skeletons $S_i (i = 1..n)$ with associated "field functions" f_i is defined by: $S = \{P \in I\!R^3 \ / \ f(P) = 1\}$ where $f(P) = \sum_{i=1}^{n} f_i(P)$ [2]. The skeletons S_i can be any geometric primitive admitting a well defined distance function. The field functions f_i are decreasing functions of the

[2] Throughout this paper, we always use the value 1 as the "isovalue" defining the isosurface of an object.

distance to the associated skeleton (we call e the distance for which the field is 1, k the slope in e, and R the radius of influence of the skeleton; see Figure 1). The surface surrounds a volume defined by $f(P) \geq 1$, and normal vectors $N(P)$ are directed along the gradient of f.

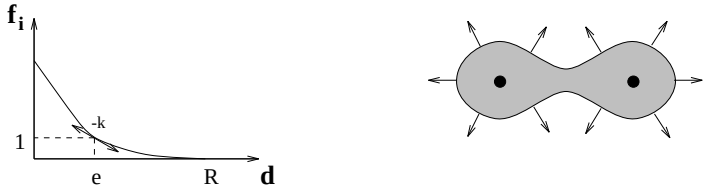

Figure 1: Field generated by two points, and the resulting implicit surface.

Modeling Elasticity with Field Functions

The set of points verifying $f(P) = 1$ being fixed, the variation of f around the isosurface can be used to model physical properties. Therefore, stiffness can be directly stored in the field's gradient by choosing f such as: $\vec{\nabla} f(Y) = -k_P(Y)\,\vec{N}(Y)$, where $k_P(Y)$ is stiffness at point P when this point has been moved to position Y. Elasticity can then be expressed in terms of field functions:

$$df(Y) = \vec{\nabla} f(Y) \,.\, d\vec{Y} = \vec{\nabla} f(Y) \,.\, \frac{d\vec{R}(Y)}{k_P(Y)} = \vec{N}(P) \,.\, d\vec{R}(Y)$$

If the small applied force $d\vec{R}$ is radial, the normal vector remains unchanged during deformation, and integrating the last equation yields a very simple correspondence between applied force and deformation : $g(P) = \vec{N}(P) \,.\, \vec{R}(P)$, where $g(P) = \int df(Y)$ is the deformation field term. This correspondence will be used during animations for computing radial response forces from deformations of colliding objects.

Collision and Contact Processing

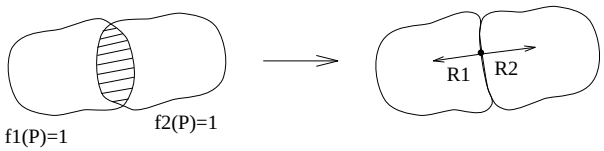

Figure 2: Deformation and response during a collision.

In [Gas93], implicit elastic material is attached to a rigid component used for computing motion (variations of the inertia tensor due to deformations are neglected). The implicit layer processes collisions and deformations, and provides the rigid component with response forces to be applied at the next time step. Collision processing is performed in three steps (see Figure 2) :

1. Detect interpenetrations between objects. After a predetection using bounding boxes, use the implicit inside/outside functions by testing one solid's sample points against the other's field function.

2. Generate exact contact surfaces by deforming the solids. Apply a negative compression field $g(P)$ in the interpenetration area, and a positive dilatation field in the propagation zone in order to efficiently model the transversal propagation of deformations in damped material.

3. Compute the response forces and the torques to be applied to the rigid components at the next time step. Our method for generating exact contact surfaces takes the objects' local stiffnesses into account, so opposite compression forces $\vec{R}(P) = g(P)\vec{N}(P)$ are applied to both sides of the contact surfaces. These forces are integrated together with fluid friction forces.

3 A Hybrid Model for Highly Deformable Material

As emphasized in the introduction, a simple and unified way of modeling from stiff to very soft behaviors and for simulating material capable of breaking into pieces is to use physically-based particle systems. However, the main drawback of these systems is the lack of methods for defining a soft surface for the objects, which could be used for collision processing and for display.

The main point of our approach is to adapt the elastic material presented in Section 2 to an implicit coating around a generalized particle system. In other words, each particle interacting with the others through long range attraction forces and short range repulsion forces will be seen as a geometric skeleton contributing to the implicit surface of an object. During an animation, the relative motions of the skeletons will automatically produce adequate deformations and topological changes of the objects' surfaces. An important benefit of the elastic implicit coating will be to provide a good model for detecting and processing collisions.

Animation of the Internal Structure

As explained previously, skeletons defining an implicit surface may be any geometric primitive admitting a well defined distance-function (points, segments, curves, triangles, simple volumes..). To model a block of highly deformable material, we choose the number of skeletons, their shapes, and the flesh thickness around each skeleton according to the shape and size of the macroscopic components that we want to be unbreakable during any animation. Our approximation consists in considering, while computing motion, that the mass of the implicit body is entirely concentrated in the skeletons. Each skeleton is represented as a rigid body, with its associated mass and inertia tensor, animated by integrating over time rigid body equations of motion (or point-wise dynamics if the skeleton is a point). Skeletons are linked together by adequate interaction forces, constituting a generalized physically based particle system.

We use Lennard-Jones interaction forces as in [Ton91, Jim93]. The force to be applied to the center of a skeleton S_1 interacting with another S_2 is :

$$\vec{F} = K \left(\frac{r_0^n}{r^{n+1}} - \frac{r_0^m}{r^{m+1}} \right) \frac{\vec{r}}{r} \qquad (1)$$

where $\vec{r} = S_1 - S_2$ and $r = ||\vec{r}||$ The remaining parameters of Equation (1) can be used as follows: r_0 represents the rest distance between the skeletons, K controls the softness of the system (when K increases, the global behavior is stiffer). The assumption $n = 2m$ is commonly used. We are currently experimenting highly deformable material with values $m = 4, n = 8$.

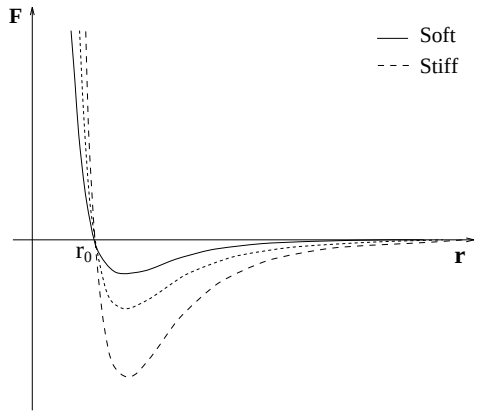

Figure 3: Interaction laws for three different values of K.

The use of these interaction forces is not sufficient for modeling viscous behaviour as particles may oscillate indefinitely around a rest position. We avoid this problem by adding damping forces to the model. The damping force applied to a particle should depend on the local density of particles around it. To achieve this, for each particle we compute a damping force with respect to every other particle that interacts with it. Damping in a viscous medium is usually proportional to the square of the object speed. Here, only the relative speed of the two interacting skeletons will produce damping. The total damping force applied on a skeleton S of speed vector $\vec{V}$, and interacting with the skeletons S_i of speed vectors $\vec{V_i}$, is given by:

$$\vec{D} = \sum_i \delta_i(\text{dist}(S, S_i))\, ||\vec{V_i} - \vec{V}||\, (\vec{V_i} - \vec{V})$$

where δ_i is a damping coefficient based on the distance between S and S_i. In practice we use a simple function with only two parameters: r_1 controlling the area of influence and α providing the maximum influence, i.e.:

$$\delta_i(r) = \begin{cases} \alpha.\frac{(r-r_1)^2}{r_1^2} & \text{if } r \in [0, r_1] \\ 0 & \text{otherwise} \end{cases}$$

These damping forces offer easy tuning of the viscosity of the deformable materials.

Coherent Volume Variations

Simply animating the skeletons of an implicit solid while using an arbitrary field function to coat them would not produce good results. If no particular

care is taken in the choice of parameters, applying compression forces to internal structures could dramatically increase the volume. Figure 4 shows what would happen if we model highly deformable material with a quadratic field function as used in [Gas93], where all skeletons were motionless in the object local coordinate system.

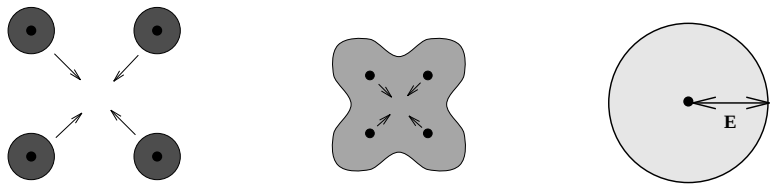

Figure 4: Compression of the internal layer should not increase the volume!

Studies of materials that conserve their volume during deformations has already been performed in Computer Graphics [PB88]. To model more general behaviors, a certain amount of volume variation can be allowed. Compression forces applied to the internal structure of the material in Figure 4 should produce a compression of the total volume, but never a dilatation.

Wyvill *et al.* [WMW86] previously pointed out the fact that two skeletons in the same place should create a sphere with twice the volume of a one-skeleton sphere. The choice of their *isovalue* ensures this constraint only when two point-skeletons merge. Let us express this problem more generally, when any number of particles are concentrating in one point (see Figure 4). Suppose that n punctual skeletons are initially in positions where their field functions do not blend together, and that they move to the same location. Initially composed of n spheres of radius $e = f^{-1}(1)$, the object blends into a single larger sphere of radius $E = (n.f)^{-1}(1) = (f)^{-1}(1/n)$. We write:

$$\frac{4}{3}\pi E^3 \leq n \cdot \frac{4}{3}\pi e^3 \quad \text{which is equivalent to:} \quad f^{-1}(\frac{1}{n}) \leq \sqrt[3]{n} \cdot e$$

so that the global volume will decrease during this process. This last equation must be verified for every n smaller than the number of particles in our system: this gives us a set of constraints for f. Combining the last equation with the fact that f is a decreasing function and taking $x = \sqrt[3]{n}.e$ yields:

$$\forall x \geq e \quad f(x) \leq \left(\frac{e}{x}\right)^3 .$$

In particular, the continous function $f(x) = (e/x)^3$ verifies all the constraints, and so maintains volume between the configurations of Figure 4(a) and Figure 4(c).

However, even if the volume is constant or decreases between the two extreme configurations of Figure 4, a maximum may appear in an intermediate position. Let us consider an implicit solid generated by two punctual skeletons. Figure 5 represents the volume[3] variations as a function of the distance between

[3] As the formula expressing volume cannot be integrated analytically, we have approximated volume by discretizing space into small voxels.

the skeletons. For the field $f(x) = (e/x)^3$, there is a maximum near $d = 2e$, where e is the radius of the implicit spheres associated with each skeleton, which represents a 19% increase of the volume value. This result is still better than the 43% increase produced by the quadratic field of [Gas93]. Moreover, even if a field function admits a maximum value for volume, coherent choices for the internal and external parameters of our hybrid model can produce correct behaviors. In Figure 5, if the distance between the skeletons oscillates around the equilibrium position $d = e$, the volume will, as expected, decrease during a compression and increase during a small dilatation of the internal structure. Except during fractures, distances between skeletons in a block of deformable material do not vary much during a simulation: the block reaches an equilibrium state, and skeletons oscillate near rest positions. So in practice, we choose in Equation (1) the rest distance r_0 between the skeletons to be equal to the middle value e of the "validity interval" for which coherent compressions and dilatations are produced.

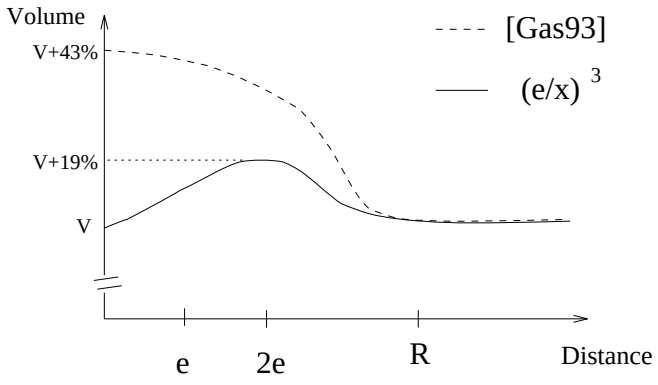

Figure 5: Volume as a function of the distance between two skeletons.

4 Animation and Collision Processing

The general animation algorithm for highly deformable material can be described as follows. At each time step:

1. Compute new positions for the skeletons, by integrating the associated equations of motion from the set of externally applied forces.

2. Detect collisions, avoid interpenetrations by generating exact contact surfaces between objects, and compute response forces.

3. Display the resulting deformed shapes.

The first phase requires smaller time steps than the one used for display, so deformations and response forces are calculated every Δt, whereas integration of particles movement are performed every $dt = \Delta t/k$.

Our method for collision detection and response is similar to the algorithm given in Section 2. However, as objects may break into pieces during simulations, bounding boxes can become too large, i.e. inefficient, and finding sample

points at each time step becomes a more serious challenge. Moreover, performing these operations very efficiently is essential for computing the animation at interactive rates. Another problem that is introduced when modeling highly deformable material is defining how response forces should be transmitted into the internal structure of the solids. The next two sections address these problems.

Efficient Adaptive Sampling for Topology-Variable Objects

Contrary to elastic implicit solids that only deform locally and then recover their initial shape, blocks of highly deformable material may suffer any topological change during a simulation. A violent collision may create a hole, a block can break into pieces, and several blocks made of the same material can melt[4]. In consequence, pre-sampling the objects before a simulation and then always using the same sample-points, as was done in [Gas93], is no longer possible. Unfortunately, spatial partitioning polygonization is compute-intensive, thus using these techniques at each time step would prevent any interactive computation and display of the animation.

Consequently, we have developed a new approach for an efficient adaptive sampling of highly deformable material. Based on the idea of seed migration introduced in [BW90], our method benefits from temporal coherence between the frames of an animation. The basics are the following: each skeleton sends a set of seed points to sample the surface, with a good spatial distribution around it. A seed, always moving along a fixed axis with respect to its skeleton's local coordinate system, can be efficiently recomputed at each time step from its last position in this coordinate system : very few search steps are required (because of time coherence) to find an inner and an outer point to start a binary search on the field function (see Figure 6).

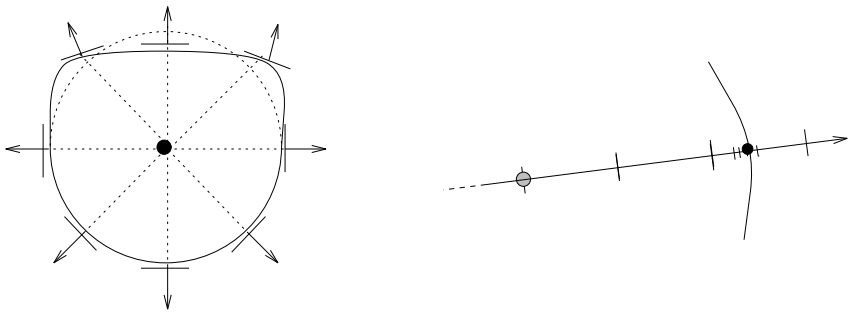

Figure 6: Seed migration during deformation by large steps and binary search.

To increase efficiency and to ensure a reasonable distribution of seed points when several basic volumes melt together, some of the seeds are invalidated at each time step (see Figure 7): during search steps, if a seed enters an area where the field generated by its own skeleton is smaller than another field contribution,

[4] In our current implementation, the user defines one or several blocks of highly deformable material, each one being breakable into several pieces. Collisions are computed between pieces from different materials while components from the same object always "melt" together.

the seed stops and is marked as invalid. An invalid seed may reappear if the topology of the object changes.

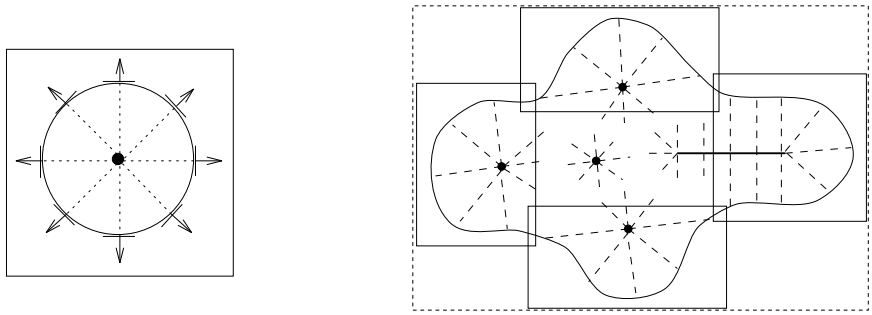

Figure 7: Seed points used for sampling and associated bounding boxes.

Another problem is defining bounding boxes for optimizing collision detection between highly deformable objects. Storing an axis-parallel bounding box in each object's local coordinate system [Gas93] cannot be done, since an object may be composed of several pieces and each skeleton comprising an object has its own local coordinate system. Moreover, using this large box would be inefficient when an object has broken into pieces. Therefore we attach an axis-parallel bounding box to each skeleton. The size of the box is computed from the position of the valid seeds sent by this skeleton so that the object surface – but not necessarily the object volume – is included in the union of skeleton-boxes. Then a large bounding box is computed from all those boxes as shown in Figure 7. Collision detection uses this hierarchy of boxes to reduce the number of field evaluations. If two large boxes from different objects interpenetrate, collisions are detected between pairs of skeleton-boxes. For each colliding pair, only the seed points attached to one box and located inside the other are tested against the other object's field function. A useful extension would be to add to the hierarchy an intermediate box computed from each connected component of an object.

Response to Collisions: Transmitting Forces to the Internal Structure

After the "modeling-contact" step, which works in exactly the same way as in Section 2 (deformation fields generating exact contact surfaces are applied to colliding objects), response forces computed on contact surfaces must be transmitted to the internal structure. However, in contrast to the method in Section 2, there is no-longer one rigid component to which send the forces, but several different skeletons, each contributing to the field value.

If an external force is applied at a point P of the surface (that verifies $\sum_i f_i(P) = 1$) a first approach consists of distributing this force between the skeletons S_i in proportion to their amount of contribution to the normal vector at P. See Figure 8(a). A good way to do this, especially if the applied force is directed along the normal vector at P, is to use the decomposition of the

normal vector at P into the sum of gradient vectors from each skeleton :

$$\vec{N}(P) = \frac{\sum_i \vec{\nabla} f_i(P)}{||\sum_i \vec{\nabla} f_i(P)||}$$

So, if we let $\vec{n_j}(P)$ be: $\vec{n_j}(P) = \vec{\nabla} f_j(P)/||\sum_i \vec{\nabla} f_i(P)||$, a radial force applied on the initial shape $\vec{F} = F\vec{N}(P)$ at P can be split into forces $F\vec{n_j}(P)$ applied to each skeleton S_j, together with the associated torques if needed.

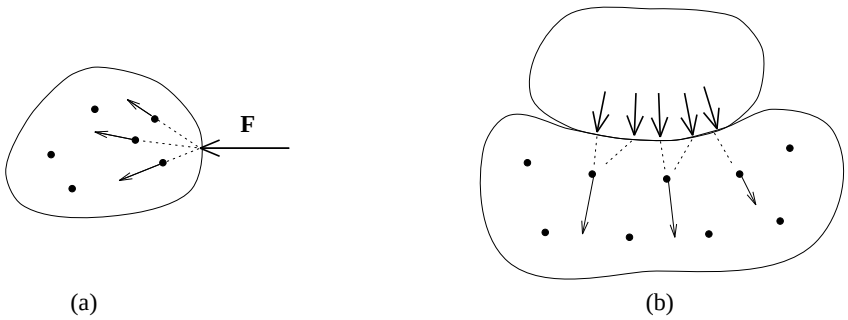

Figure 8: Two different solutions for transmitting forces to the skeletons.

This first method gives a good distribution of punctual forces applied on an object surface without collision, but this is too restrictive. However, in the case of response forces occurring across an entire contact surface after collisions, the algorithm can still be simplified. Our seed points are, in effect, a precise sampling of a smooth contact surface between two deformable objects. So, transmitting the force at a seed point to the skeleton that contributes the most to the potential at this point is quite sufficient. Moreover the invalidation test seen in section 4 eases the computation, because it ensures that the response force calculated on a seed will be transmitted to the skeleton that sent this seed. So a skeleton will be affected proportionally to its contribution to the contact surface sampling. See Figure 8(b). The interactions inside the system of skeletons will propagate the action of the external forces throughout the structure. The simplified algorithm produces good results, as demonstrated by the animations that we describe in the next section.

5 Results and Concluding Remarks

Animation Sequences

The sequence depicted in Figure 8 has been computed with only 18 point-skeletons for the block of material. This sequence shows the result of weak interaction forces that create a viscous fluid behavior. The second one uses stronger interaction forces, preventing objects from breaking up on impact. These objects contains only 12 and 8 particles respectively (see Figure 9). The benefit of detecting collisions with the exact surface is that this surface does not need to stay very close to the particles. This enables us to generate very smooth objects with few skeletons.

Ceci est la page de figures en couleur.

Conclusion

This paper presents a new method for modeling and animating soft blocks of highly deformable material, capable of breaking into pieces. The model is hybrid: deformable material is composed of several rigid skeletons interacting together, coated with elastic flesh. Our implicit formulation for the flesh offers an analytic definition of the object surface throughout the simulation. Coherent parameter choices for the external and internal structures ensure consistent volume variations during the simulation. Used for precise collision detection, the flesh deforms during contacts with other objects. Reaction forces computed along contact surfaces are transmitted to the internal structure, producing subsequent deformations. Thanks to an efficient adaptive sampling algorithm, the simulation is produced at interactive rates, which facilitates the precise tuning of an animation.

Work in progress includes attempts to further optimize the animation algorithm (in particular by improving the way we recompute and validate seed points), to find a better solution for avoiding volume variations when objects break into pieces or melt, and to implement solid friction between contact surfaces.

Acknowledgements

Many thanks to Nicolas Tsingos for implementing an efficient visualisation of implicit surfaces, to Jean-Dominique Gascuel for his constant support during this research, and to George Drettakis, Kevin Novins, Frederic Cazals and Jean-Christophe Lombardo for rereading this paper.

References

[BW90] Jules Bloomenthal and Brian Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, 24(2):109–116, March 1990.

[Gas93] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics*, pages 313–320, August 1993. Proceedings of SIGGRAPH'93 (Anaheim, California, August 1993).

[Hou92] Donald House. Coupled particles: Theory. *Particle System Modeling, Animation, and Physically-Based Techniques (SIGGRAPH'92 course notes Number 16, Chicago, Illinois)*, 1992.

[Jim93] Stéphane Jimenez. Modélisation et simulation physique d'objets volumiques déformables complexes. Thèse de doctorat, Institut National Polytechnique de Grenoble, November 1993.

[LJF+91] A. Luciani, S. Jimenez, J-L. Florens, C. Cadoz, and O. Raoult. Computational physics: a modeler simulator for animated physical objects. In *Eurographics'91*, Vienna, Austria, September 1991.

[LJR+91] Annie Luciani, Stéphane Jimenez, Olivier Raoult, Claude Cadoz, and Jean-Loup Florens. An unified view of multitude behaviour, flexibility, plasticity, and fractures: balls, bubbles and agglomerates. In *IFIP WG 5.10 Working Conference*, Tokyo, Japan, April 1991.

[PB88] J.C. Platt and A.H. Barr. Constraint methods for flexible models. *Computer Graphics*, 22(4):279–288, August 1988.

[TF88] Demetri Terzopoulos and Kurt Fleischer. Modeling inelastic deformations: Viscoelasticity, plasticity, fracture. *Computer Graphics*, 22(4):269–278, August 1988. Proceedings of SIGGRAPH'88 (Atlanta, August 1988).

[Ton91] D. Tonnesen. Modeling liquids and solids using thermal particles. In *Graphics Interface'91*, pages 255–262, Calgary, Canada, June 1991.

[TPF89] Demetri Terzopoulos, John Platt, and Kurt Fleisher. Heating and melting deformable models (from goop to glop). In *Graphics Interface'89*, pages 219–226, London, Ontario, Canada, June 1989.

[WMW86] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for soft objects. *The Visual Computer*, pages 227–234, August 1986.