Animating Soft Substances with Implicit Surfaces

Mathieu Desbrun, Marie-Paule Cani

Animating Soft Substances with Implicit Surfaces

Mathieu Desbrun *Marie-Paule Gascuel*
iMAGIS*/ IMAG

Abstract

This paper presents a hybrid model for animation of soft inelastic substance which undergo topological changes, e.g. separation and fusion and which fit with the objects they are in contact with. The model uses a particle system coated with a smooth iso-surface that is used for performing collision detection, precise contact modeling and integration of response forces. The animation technique solves three problems inherent in implicit modeling. Firstly, local volume controllers are defined to insure constant volume deformation, even during highly inelastic processes such as splitting or fusion. Secondly, we avoid unwanted distance blending between disconnected pieces of the same substance. Finally, we simulate both collisions and progressive merging under compression between implicit surfaces that do not blend together. Parameter tuning is facilitated by the layered model and animation is generated at interactive rates.

Keywords: implicit surface, physics-based animation, inelasticity.

1 Introduction

Most deformable models in Computer Graphics are dedicated to visco-elastic deformation: objects deform under an external force field and then progressively come back to their rest shape. Animating highly deformable inelastic substances, such as clay, dough or mud, is a more challenging problem. These substances are characterized by a smooth surface that fits with the objects it is in contact with and can undergo any topological change. One can make a hole in it, split a block of substance into several pieces and even merge two pieces together by compressing them strongly against each other. During all these deformations the total volume remains approximately constant. This paper presents an integrated set of methods for simulating these behaviors.

1.1 Previous inelastic models

Contrary to elastic objects, the shape of inelastic bodies depends on the entire history of applied forces. Terzopoulos et al. [10] use two layers to simulate this behavior: an inelastic reference component, that computes motion and absorbs large scale deformations, and an elastic layer that represents the difference between the current and reference shapes. The model handles visco-elasticity, plasticity and fractures. However, since the lattice used for discretizing equations has a fixed topology, the model is restricted to very structured inelastic objects.

Other models [5, 11, 12] use physically-based particle systems for modeling a wide range of behaviors, including visco-elasticity, plasticity, collisions, separation and fusion. The change from stiff material to the quasi-liquid state is achieved by adapting the interaction laws between particles. However, visualizing the surface of a substance during animation is difficult since particles can change their positions during deformations. Therefore, a fixed set of "boundary particles" cannot be used for surface representation. One solution is to display an iso-surface generated by the set of particles [11, 12]. Nevertheless, since this surface is only introduced for rendering and is not considered for collision detection, visual anomalies such as local inter-penetrations with obstacles or bouncing before contact may occur.

1.2 Overview

This paper presents a new model for interactive animation of smooth soft substances which fit with other objects during contact, can be split into pieces and may merge when disconnected components are compressed against each other. The model ensures volume preservation, performs collision detection and models precise contact surface and local deformation during collisions. A precise description of the surface of an object is maintained throughout the animation and can be used for final high quality rendering.

Implicit surfaces seem to be the best surface representation for smooth bodies that deform over time and may change their topology [13]. Our basic idea, introduced in [2], is to combine particle systems and implicit surfaces during the animation. Controlled by the particles as in [11, 12], the implicit surface is animated according to the implicit elastic model of [4] that gives it the ability to detect collisions, to deform locally for exact contact modeling, and to compute precise integration of response forces. These forces are transmitted to the particles to be subsequently integrated.

However, the direct use of this model generates a number of anomalies. This paper presents novel and general methods for controlling volume variation, avoiding unwanted blending effects, and simulation of both collisions and progressive fusion under compression of disconnected pieces of the same substance.

Section 2 reviews the hybrid model for soft substances introduced in [2], and discusses its limitations. Section 3 presents our new method for generating constant-volume deformations. An algorithm for performing separation without subsequent distance blending is detailed in Section 4. Section 5 explains how to process either collision or progressive fusion between surfaces that do not blend, according to the amount of compression forces.

2 A Hybrid Model for Soft Substances

2.1 Combining particles and implicit surfaces

The hybrid model we use for modeling soft inelastic bodies is composed of two layers (see [2]). Motion and large scale deformations are governed by an inelastic reference component made of particles, while an elastic implicit layer gives the current shape of an object and is used to compute local deformation during collisions.

Reference component: As in [5, 12], we model inelasticity with a particle system, i.e. a set of mass points P_i subject to both attraction/repulsion forces F_{int} and fluid friction forces F_{fr} depending on

*iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble and Université Joseph Fourier.
Address: BP 53, F-38041 Grenoble cedex 09, France
Email: [Mathieu.Desbrun|Marie-Paule.Gascuel]@imag.fr.

local particle density. In our system, the forces applied by particle P_1 on particle P_2 are:

$$F_{int}(P_1 \to P_2) = \lambda \left(\left(\frac{r_0}{r} \right)^8 - \left(\frac{r_0}{r} \right)^4 \right) \frac{P_2 - P_1}{r^2} \quad (1)$$

$$F_{fr}(P_1 \to P_2) = \mu(r) \, ||\dot{P}_1 - \dot{P}_2|| \, (\dot{P}_1 - \dot{P}_2) \quad (2)$$

where $r = ||P_2 - P_1||$, λ is a parameter for regulating the stiffness of a material, $\dot{P}_i$ is the speed vector of particle P_i, and μ is a decreasing continuous function with a restricted scope of influence.

External elastic layer: Implicit surfaces such as distance surfaces [1, 13] are particularly suitable for animating deformable bodies capable of splitting and fusion. We use them as a coating over a particle system: each particle generates a field f_i, a smooth decreasing function of the distance with a restricted scope of influence, and the surface of an object is defined as the set of points P such as $f(P) = \sum f_i(P) = s$, s being a given isovalue.

The implicit elastic model of [4] is used to animate the implicit surface and for collision detection and response. This model defines a simple correspondence between applied forces and deformation, the force at a particular point being given by the local variation of the field value. Exact contact modeling is performed during collisions by adding deformation terms to the fields defining objects: for objects defined by $f_1 = s$ and $f_2 = s$, the respective values of these terms are $s - f_2$ and $s - f_1$. This generates an exact contact surface of equation $f_1 = f_2$ where opposite normal compression forces $F_{1\to2} = -F_{2\to1} = (s - f_1)N_2$ are applied.

At each time step, animation is computed as follows:

1. Compute the new position of each particle by integrating the associated equation of motion from the set of applied forces.
2. After a pre-detection with bounding boxes, use the implicit surface generated by the particles for detecting collisions (test the sample points of an object against the field of another one).
3. Avoid inter-penetration by generating exact contact surfaces between colliding objects, and compute response forces.
4. Distribute response forces between particles that contribute to surface generation in contact area. These forces will be used at the next time step.

2.2 Problems to be solved

Despite its capability of defining smooth substances that fit with other objects during contact, this hybrid model generates several anomalies.

Due to the implicit coating, a piece of substance may undergo very significant volume variation during deformations, especially during separation and fusion. The partial solution proposed in [2] is far from sufficient. Firstly, it is based on the choice of a specific field function[2], which is very restrictive since both the shape of an object and its stiffness are controlled by this function. Moreover, it gives good results near equilibrium states only, so it is of no use for animating large scale deformations and topological changes.

The second problem concerns the irreversibility of soft substance splitting. Two pieces coming back close to each other should not produce the same intermediate shapes than when they were disconnected, i.e. they should not blend before contact [9]. Unfortunately, since they are components of the same implicit body, their surfaces locally inflate and merge. This artifact is related to the well-known "unwanted blending problem" [6, 14], but the difficulty is intensified in this case, since the desired blending properties for a soft substance are changing with its topology.

[2]This function is $f_i(P) = (r_0/r)^3$, where r is the distance $d(P, P_i)$ and where r_0, introduced in equation (1), is the rest distance between particles.

Finally, both collision and progressive fusion between soft bodies are to be produced, depending on their physical properties and on the amount of compression forces that press them against each other.

The following sections present solutions to these three problems.

3 Constant Volume Deformation

Constant volume deformation of flexible models have already drawn some attention [7, 8]. In the case of objects discretized into lattices of fixed topology, the problem can be solved by using constrained optimization techniques based on Lagrange multipliers. To the authors knowledge, no solution has been proposed in the case of soft substances, although volume variations may increase due to topological changes such as separation or fusion.

This section presents a general method for controlling volume of objects defined using implicit surfaces. This method is well adapted to the soft substances we are modeling, but can also be applied to any other way of animating implicit surfaces.

3.1 Basic ideas

Our aim was to develop a general method for controlling the volume of an implicit object that does not set any restriction on the choice of the field function, thus allowing for a wide range of shapes and stiffness to be modeled.

First of all, a good way of detecting volume variation must be defined. An implicit volume defined by $V = \int\int\int_{f(P)>=s} dxdydz$ where $P = (x, y, z)$, cannot be computed analytically for arbitrary field functions. Discretizing space into voxels can be used for finding an approximate value. However, this way of detecting volume variation would not give us any chance of solving the problem. As illustrated by the example in Figure 1, reducing the strength of field contributions at step 2 in order to avoid volume variation should only be done in the area where the object has been deformed. Thus, volume should be controlled locally.

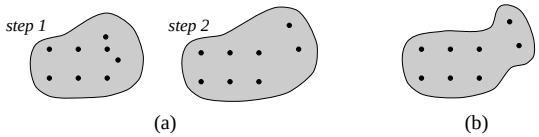

Figure 1: (a) Volume variation. (b) Local volume control in step 2.

Our basic idea is to detect the area where the volume is changing, and then adjust the strength of local fields in this area. Thus, a notion of local volume needs to be defined.

- We call the *territory* Ti of a particle P_i the part of the implicit object where its field contribution is the highest. Territories form a partition of the implicit volume $(f(P) \geq s)$.

$$T_i = \{P \in I\!R \ / \ (f(P) \geq s) \text{ and } (\forall j \ f_i(P) \geq f_j(P))\}$$

- The *local volume* V_i associated with P_i is the volume of T_i.

3.2 Detecting local volume variation

We are looking for an efficient way of approximating local volume. Since deformation is continuous over time, we can take advantage of temporal coherence. This is achieved by maintaining a sampling of territory boundaries throughout the animation.

Each particle sends a fixed number of points called "seeds" to sample its territory boundary, in a set of distributed directions called "seed-axes" that are defined in the particle local coordinates system (see Figure 2). At each animation step, seeds migrate to the surface from their previous position along their axis. They stop either when

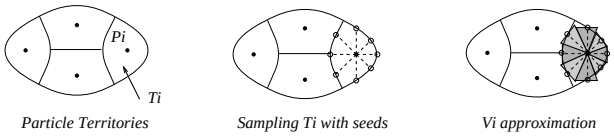

Figure 2: Particle territories and seeds used for volume approximation.

they have reached the isosurface or when the preponderant field f_i becomes smaller than another one.

We approximate V_i by the sum of the volumes of small pyramids defined by seeds (see Figure 2):

$$V_i = \sum_{s \in S_i} K_i \, d(s, P_i)^3$$

where S_i is the set of seeds sent by P_i, and where the factor K_i only depends on the seed repartition chosen for P_i. In practice, K_i can be left out in volume computations, since it is sufficient to maintain $\sum_{s \in S_i} d(s, P_i)^3$ at a constant level.

3.3 Local volume control

We control local volume variation by associating a proportional-derivative controller with each particle. Given the current local volume V_i and the initial value $V_{i,0}$ to maintain, this controller outputs an adequate adjustment of the field function f_i.

For our application, the way of modifying f_i must be chosen carefully since the norm of its gradient gives the local stiffness of an object [4]. In order to adjust the volume of particle territories without modifying the physical properties of the object, we combine the original field function with a translation $\epsilon_{i,t}$. At each time step, the field, originally defined by the decreasing function of the distance $f_i(P) = h_i(d(P, P_i))$, is replaced by: $f_{i,t} = h_i\,(d(P, P_i) - \epsilon_{i,t})$.

In order to produce steady shape variation, we control the time derivative $\dot{\epsilon}_{i,t}$ of the translation parameter rather than its value. The input of the proportional-derivative controller consists of the normalized volume variation $\Delta_{i,t}$ and its time derivative $\dot{\Delta}_{i,t}$:

$$\Delta_{i,t} = \frac{V_{i,t} - V_{i,0}}{V_{i,0}} \qquad \dot{\Delta}_{i,t} = \frac{V_{i,t} - V_{i,t-dt}}{V_{i,0}\, dt}$$

and its output is: $\dot{\epsilon}_{i,t} = \alpha\Delta_{i,t} + \beta\dot{\Delta}_{i,t}$

A simple example of volume control is given in Figure 3. Figures 5 and 6 show the results obtained during fusion, where volume control is essential. Otherwise, very significant and sudden increases of volume would be produced when two soft bodies merge.

(a) *Initial positions* (b) *Without control: 41% of volume increase* (c) *Constant volume (error under 3%)*

Figure 3: Volume control during a blending process ($\alpha = 10.0$ and $\beta = 1.0$).

4 Avoiding Unwanted Blending

One of the main difficulties raised by the animation of implicit surfaces is the avoidance of undesirable blending effects between objects. This problem is well known in the case of character animation: arms and legs of a character should not blend together, although both blend with the body. The solution suggested in [14] pre-defines a blending graph where a connection between two skeletons indicates that their field contributions are to be added. Then, the field value at a point P is computed by first finding the skeleton with the highest field contribution at P, and then adding the contributions of neighboring skeletons in the graph[3].

The unwanted blending problem is more difficult in the case of a soft substance splitting into pieces. If two disconnected pieces come back close to each other, they will blend at some distance as in Figure 3 rather than colliding because they are considered to be parts of the same object. Moreover, since the topology of the substance varies over time, a pre-defined blending graph cannot be used: the blending properties of the particles change according to the surface decomposition into connected components.

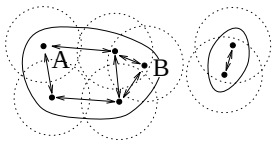

Figure 4: The influence graph and its connected components. Particles A and B lie in the same component, so their fields will blend if they come close to each other.

Therefore, we compute a time-varying blending graph, represented by lists of neighbors called a "blending list" associated with each particle. Processing unwanted blending is achieved by reducing blending lists each time the implicit surface breaks into disconnected components that no longer blend. Before the animation, the blending graph is initialized as a complete graph, where each particle is connected to all others. Then, at each animation step:

1. For each pair of particles, look if their spheres of influence, defined by the radius of influence of their field, intersect. Transitive closure of this relation, that defines an "influence graph", gives the "influence connected components" (see Figure 4).

2. For each particle, remove from its blending list those of the neighbors that are no longer in the same influence component.

3. Use the blending graph and the field function it defines for computing seeds migration. Since seeds sample the territory boundary of a particle, fields can be evaluated very efficiently: we already know which field contribution is the highest, so we just have to add the contributions from the neighbors.

The soft substances animated with this method split into components that no longer blend, as shown in the three first frames in Figure 5. The next section explains how to handle collisions between these components, and to enable fusions according to the amount of compression forces and to the properties of the substance.

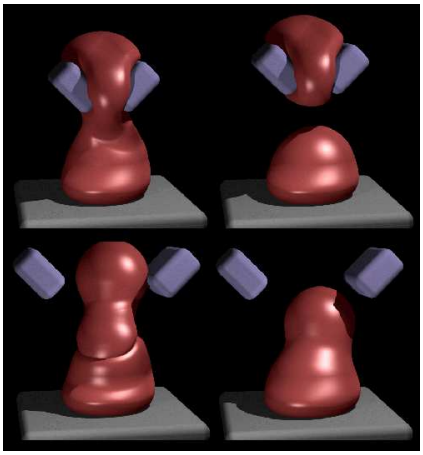

Figure 5: Soft substance made of 9 particles grabbed away by pliers and released.

[3]The blending graph must be defined with care, otherwise surface discontinuities may appear between areas where a given skeleton's contribution is still considered, and zones where it is not a neighbor anymore.

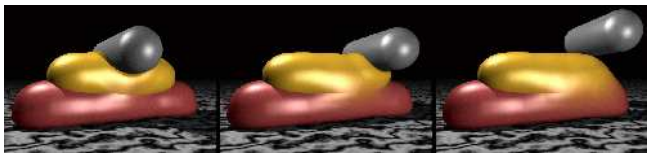

Figure 6: Progressive fusion under compression of two colliding bodies.

5 From Collision to Progressive Fusion

5.1 Collision processing

Instead of processing collisions between pairs of objects as in [2, 4], the collision processing algorithm computes collisions between *each pair of particle territories* such that the fields of the particles do not blend together. As a result, if a soft substance splits into pieces, subsequent collisions will be detected between the pieces. The main problem is to enable collision detection between particle territories. The methods for modeling contact and computing collision response, reviewed in Section 2, do not need to be modified.

The data needed for detecting collisions between implicit surfaces are a bounding box around each of these surfaces and a set of points that sample them. These sample points will be tested against the field function of another object. In our implementation, we use the seeds introduced in Section 3 for both sampling the portion of the surface associated with a particle territory T_i and computing a local bounding box around it:

- A seed is said to be "valid" if it reaches the iso-surface during migration. At each time step, the set of valid seeds associated with T_i gives the sample points needed for collision detection.
- The axis-parallel bounding box associated with T_i is computed from the positions of the valid seeds, and enlarged by the maximum distance between neighboring seeds.
- Local bounding boxes are grouped into connected component boxes to optimize collision detection.

Valid seeds can also be used for the interactive opaque display of the substance during the animation [3].

5.2 Fusion under compression

Blocks of soft substance such as clay or dough merge under compression forces that exceed a specified threshold. This behavior can be easily simulated with our model.

The fusion threshold is added as a new parameter in a substance description. Then, each time a collision is computed between two bodies made of the same substance, compression forces computed along the contact surface between two particle territories are compared with the fusion threshold. If the compression exceeds it, each particle adds the other one to its blending list. At the next time step, fields from the two bodies will locally blend in this area, while collisions will still be computed elsewhere (see Figure 6).

As mentioned in Section 4, the use of a blending graph that is not fully connected may produce discontinuities in the implicit surface. A solution, introduced in [6], defines the field value at a point P as the maximum field from groups of particles that blend together. In our current implementation we obtain the same effect by rendering a substance as a set of implicit components representing these groups. Each component includes the area where the groups merge and a union of all components gives the final isosurface. However, neither of these methods avoids tangent discontinuities in the final shape.

6 Conclusions

We have presented a hybrid model for animation of smooth soft substances that maintain their volume, collide and can undergo separation and fusion during animation. Control is facilitated by the layered nature of the model. It combines a particle system that models large scale inelastic deformations, an implicit surface that generates local deformations during contact and a control module that performs local volume preservation. Animation is computed and visualized at interactive rates. The implicit surface parameters are stored at each animation step providing a compact storage of the animation and enabling the direct use of surface/ray intersection algorithms for computing final high-quality images.

The solution developed for local control of implicit volumes offers a general contribution to the field of animation using implicit surfaces. For instance, it can be used in character animation, for setting user-defined volume variation that imitate the contraction and dilatation of muscles. We are currently experimenting with a solution to avoid tangent discontinuities in the implicit surface when an object is modeled using a blending graph.

Acknowledgements

Many thanks to Jean-Dominique Gascuel for his efficient ray-tracing software, to the reviewers for their helpful comments, and to Agata Opalach for carefully re-reading this paper.

References

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

[2] Mathieu Desbrun and Marie-Paule Gascuel. Highly deformable material for animation and collision processing. In *5th Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994.

[3] Mathieu Desbrun, Nicolas Tsingos, and Marie-Paule Gascuel. Adaptive sampling of implicit surfaces for interactive modeling and animation. In *First International Workshop on Implicit Surfaces*, Grenoble, France, April 1995.

[4] 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, CA).

[5] 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.

[6] Agata Opalach and Steve Maddock. Implicit surfaces: Appearance, blending and consistency. In *Fourth Eurographics Workshop on Animation and Simulation*, Barcelona, Spain, September 1993.

[7] John Platt and Alan Barr. Constraint methods for flexible models. *Computer Graphics*, 22(4):279–288, August 1988. Proceedings of SIGGRAPH'88 (Atlanta, Georgia).

[8] Ari Rappoport, Alla Sheffer, Daniel Youlus, and Michel Bercovier. Volume-preserving free-form deformations. In *ACM Solid Modeling'95*, Salt Lake City, Utah, May 1995.

[9] Jean-Paul Smets-Solanes. Surfacic textures for animated implicit surfaces: the 2d case. In *Fourth Eurographics Workshop on Animation and Simulation*, Barcelona, Spain, September 1993.

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

[11] 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, June 1989.

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

[13] Brian Wyvill, Craig McPheeters, and Geoff Wyvill. Animating soft objects. *The Visual Computer*, 2(4):235–242, August 1986.

[14] Brian Wyvill and Geoff Wyvill. Field functions for implicit surfaces. *The Visual Computer*, 5:75–82, December 1989.