

# An implicit formulation for precise contact modeling between flexible solids

Marie-Paule Cani

## ► To cite this version:

Marie-Paule Cani. An implicit formulation for precise contact modeling between flexible solids. 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93), ACM, Aug 1993, Anaheim, United States. pp.313–320, 10.1145/166117.166157 . inria-00537550

**HAL Id: inria-00537550**

**<https://hal.inria.fr/inria-00537550>**

Submitted on 18 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An Implicit Formulation for Precise Contact Modeling between Flexible Solids

Marie-Paule Gascuel  
iMAGIS, LIENS, CNRS URA 1327  
Ecole Normale Supérieure, 45 rue d'Ulm  
75005 Paris, France

## Abstract

This paper presents an implicit deformable model, based on iso-surfaces of potential fields generated by skeletons, that provides elegant and unified formulations for both geometric parameters such as shape or deformation and physical properties such as rigidity. The model is especially designed to improve collision and contact processing for non-rigid objects. In particular, it generates and maintains exact contact surfaces during interactions.

Keywords: animation, simulation, deformation, implicit surface, collision detection, collision response.

## 1 Introduction

Dynamic animation systems based on simplified physical laws have drawn a lot of attention during the past few years. One of the reasons why they seem so attractive is their ability to respond automatically to collisions. Nevertheless, contrary to rigid solid animation where complete analytical solutions have been found [1], modeling interactions between deformable objects still remains a challenge. In particular, none of the models proposed up to now generates an exact contact surface between interacting flexible solids.

This paper presents a new, continuous model for deformable material based on an implicit formulation which unifies the description of geometry and of physical properties of solids. Well adapted to the simulation of local deformations, the model is especially designed to improve collision and contact processing for non-rigid objects. In addition to an efficient collision detection mechanism, it generates and maintains exact contact surfaces during interactions. These surfaces are then used for the calculation of reaction forces.

Compact, efficient, easy to implement and to control, our implicit deformable model would be a particularly convenient tool in character animation where locally deformable flesh must be simulated.

### 1.1 Previous approaches

Flexible models in Computer Graphics result from either nodal approaches (which include finite elements [6], finite differences [12, 13], and systems using elementary masses [9, 7]) or global approaches [11, 14]. The latter optimize the animation by approximating deformations by particular classes of global transformations. Well adapted to the animation of homogeneous blocks of elastic material, they would not, however, be convenient to use when simulating a material subject to local deformations (a sponge for instance), or when modeling non homogeneous complex objects like those used in character animation (typically, deformable coating over rigid skeletons).

Collisions between flexible objects are a complex phenomenon. In particular, they are not instantaneous and do not conserve energy. Among the solutions used to cope with this problem in Computer Graphics, penalty methods [10] are probably the most widely spread. They don't generate any contact surface between interacting flexible solids but use instead the amount of local interpenetration to find a force that pushes the objects apart. A different solution consists in using the relative stiffnesses of solids to find correct deformed shapes in contact situations. Here, response is computed by integrating deformation forces within the contact areas. But combined with a deformable model based on spline surfaces controlled by discrete spring systems [5], this method does not generate exact contact surfaces. A third approach [2] extends the analytical interaction processing used for rigid solids [1] to a global deformable model [14]. Contact surfaces are approximated by discrete sets of contact points which, as the authors emphasize, is somewhat unsatisfactory.

In all these methods, the lack of a contact surface between interacting flexible solids generates local interpenetration and imperfectly deformed shapes. The extent of these artifacts is exacerbated by the lasting quality of soft collisions, and forbids any correct evaluation of reaction forces.

### 1.2 Overview

This paper presents a new deformable model which improves interaction processing for flexible solids. Our main point is the use of isopotential implicit surfaces generated by "skeletons" to model the objects. Developed up to now as a tool for free form modeling [4, 3], this formalism has not been used as a way to

model physical properties<sup>1</sup>. It leads to a concise formulation for both geometric parameters, such as shape and deformation, and physical properties, such as rigidity and elastic behavior. The deformable model is continuous and provides easy modeling of local deformations.

The associated method for collision detection and response is an improved version of [5]. The inside/outside functions associated with implicit solids greatly reduces the computational cost of collision detection. The model generates and maintains exact contact surfaces between interacting objects. Opposite compression forces are respectively applied to the solids along contact surfaces, so a correct integration of response forces can be calculated.

Section 2 describes the implicit deformable model, and explains how to design both homogeneous and non-homogeneous flexible solids. The processing of interactions is detailed in Section 3, including the particular cases of multiple collisions and of interactions with rigid solids. Section 4 discusses implementation. Section 5 focuses on the possibilities for future research opened by our method.

## 2 Implicit Deformable Solids

Our aim is to simulate damped material where deformations due to collisions remain local. Rather than considering the general Lagrange equations of motion for non-rigid objects (as in [12]), we use the same approximation as in [13, 5]: the mass distribution of solids is considered to be constant, so motion is calculated using rigid body equations which lead to a more efficiently computed animation. More precisely, deformable solids are split into two layers:

- A rigid component which obeys the rigid body equations of motion. Its mass distribution corresponds to the object's rest shape.
- A deformable layer at rest relative to the rigid layer.

This section presents a new model for the deformable layer based on implicitly defined isopotential surfaces generated by skeletons. We first review the definition of these surfaces.

### 2.1 Implicit surfaces

Implicit surfaces such as "distance surfaces" [4] and "convolution surfaces" [3] allow the free form design of shapes through the manipulation of "skeletons" that generate potential fields. Very simple to define and to control, they constitute a good alternative to traditional implicit surfaces defined by analytical equations. An implicit surface  $S$  generated by a set of skeletons  $S_i (i = 1..n)$  with associated "field functions"  $f_i$  is defined by:

$$S = \{P \in \mathbb{R}^3 / f(P) = 1\} \text{ where } f(P) = \sum_{i=1}^n f_i(P)$$

<sup>1</sup> Flexible solids have been described by superquadrics [11, 8], another kind of implicit surface. Contrary to the approach developed here, the choice of an implicit geometric description of objects was not closely related to the way physical properties were modeled.

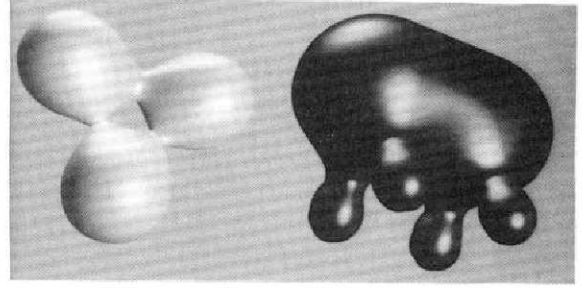


Figure 1: Isopotential objects generated by skeletons.

This surface surrounds the solid defined by  $f(P) \geq 1$ , which can have several disconnected components. Normal vectors are directed along the field's gradient.

The skeletons  $S_i$  can be any geometric primitive admitting a well defined distance function, such as: points, curves, parametric surfaces, or volumes. The field functions  $f_i$  are monotonically decreasing functions of the distance to the associated skeleton [4]. For convolution surfaces [3], they are given by integrals of exponential contributions from each point of the skeleton. In order to optimize the computations, these functions usually have a restricted scope of influence. Examples of isopotential surfaces are shown in Figure 1.

An implicit surface can easily be deformed by introducing a deformation term  $g$  in its implicit representation e.g.  $f(P) + g(P) = 1$ . In the remainder of this paper, only this type of deformations are considered.

### 2.2 Defining elastic material with potential fields

The method is based on the following observation: the set of points  $P$  satisfying  $f(P) = 1$  (where  $f$  is the field function) is sufficient to define a surface. This set of points being fixed, the variation of  $f$  around the isosurface can be used to model physical properties. The next section explains how to express stiffness with field functions in a way which yields a very simple correspondence between applied forces and resulting deformations.

#### Correspondence between forces and deformations

A deformable model is defined by a correspondence between forces and deformations. In computer graphics, this correspondence has been given by both linear [13, 6, 14] and non-linear [12] elasticity. In non-linear models, the stiffness  $k$  is not only a function of the point  $P$  you consider, but may also depend on its current location inside the solid. The applied force during a displacement of  $P$  from  $X_0 = (x_0, y_0, z_0)$  to  $X(P) = (x(P), y(P), z(P))$  is:

$$R(P) = \int_{X_0}^{X(P)} k_P(Y) dY \quad (1)$$

To improve generality, implicit deformable solids should be capable of exhibiting both linear and non-linear behaviors.

In practice, the correspondence between forces and deformations will be used during the collision process, to integrate the reaction forces colinear to normal vectors along contact surfaces between solids. For this application, defining solids with exact elastic properties at each point  $P$  along the principal deformation direction

(or “radial direction”) defined by the normal vector  $N(P)$  is sufficient.

To express exact non-linear elasticity in radial directions, we let  $dR(Y)$  be a small radial force and  $dY$  the resulting small radial displacement. From equation (1) they must satisfy:  $k_P(Y)dY = dR(Y)$ . If we express deformations by variations in the field function, it yields:

$$df(Y) = Grad(f, Y) \cdot dY = Grad(f, Y) \cdot \frac{dR(Y)}{k_P(Y)} \quad (2)$$

As said previously, we want to use the way the field function varies inside the implicit solid to express physical properties. Let us directly model stiffness with the field’s gradient:

$$\forall Y \quad Grad(f, Y) = -k_P(Y) N(Y) \quad (3)$$

This choice simplifies equation (2) which yields:

$$\int_{X_0}^{X(P)} df(Y) = - \int_{X_0}^{X(P)} (N(P) \cdot dR(Y)) = -N(P) \cdot R(P) \quad (4)$$

where the normal vector  $N(P)$  remains constant during radial deformations. Let  $g(P) = f(X(P)) - f(X_0)$  be the deformation field term associated at equilibrium with the radial force  $R$ . Here, the correspondence formula (4) becomes:

$$g(P) = -N(P) \cdot R(P) \quad (5)$$

We use equation (5) to define the general correspondence between deformations and forces characterizing implicit deformable solids. Used with a field function satisfying (3), this correspondence gives exact elastic properties in radial directions, but it associates no deformation at all with forces lying in the local tangent plane. Again, this is not a problem since the formula will only be used for computing the radial component of compression forces due to collisions.

### Modeling stiffness with field functions

Let  $S$  be an object defined by a single skeleton and  $P_S$  a point of this skeleton. The field function along the segment between  $P_S$  and the closest point of the surface can be expressed as a function  $f(r)$  of the distance  $r(P) = d(P, P_S)$ . From equation (3), the local stiffness at point  $P$  satisfies:

$$k(P) N(P) = -f'(r(P)) Grad(r, P).$$

But  $Grad(r, P) = (P - P_S) / \|P - P_S\| = N(P)$ , so:

$$k(P) = -f'(r(P))$$

The resulting geometric representation of stiffness (the opposite of the field function’s slope) facilitates the control of the simulated material. The user does not need to be a specialist in mathematical physics to easily design linear and non-linear elastic models as those of Figure 2. The field functions currently implemented are given in Appendix A.

### Homogeneity of the solids

When an object is generated by several skeletons, different field functions  $f_i$  can be associated with each one allowing non-homogeneous objects to be readily designed. The object behaves

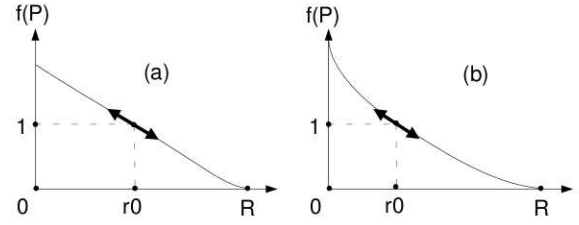


Figure 2: Examples of field functions.

(a) Linear elasticity: stiffness is constant during deformations.

(b) Non-linear elasticity: stiffness increases during compressions.

according to the local stiffness  $k_i$  in a zone influenced by a single skeleton. Otherwise, stiffness contributions from different skeletons blend together:

$$k(P)N(P) = -Grad(f, P) = - \sum Grad(f_i, P)$$

When “distance surfaces” are used, summing stiffness contributions in blending areas can be a problem. The stiffness may pass by a local extremum while varying between values associated with different skeletons. This problem corresponds to the bulge (or the narrowing) in shape which can appear when two fields superimpose [3]. Indeed, there is no reason why  $\|\sum_{j=1}^n Grad(f_j, P)\|$  should take intermediate values between the  $k_i$ . In particular, homogeneous objects are not easy to model with distance surfaces. Giving the same field function to all the skeletons is far from sufficient.

“Convolution surfaces”, for which field functions  $f_i$  are integrals of field contributions from each point of the associated skeleton, solve this problem. With this model, if the same field functions are used for several neighboring skeletons there is no bulge in shape nor in the stiffness function, so complex homogeneous objects can be designed. More generally, stiffness smoothly assumes intermediate values in areas influenced by multiple skeletons as does the field’s gradient.

### 2.3 Animation of implicit deformable solids

Implicit deformable solids are especially suitable for a precise modeling of interactions. While penalty methods directly use the degree of interpenetration between objects to evaluate response forces, our model completely suppresses interpenetrations by introducing an intermediate “contact modeling” step between the detection and the response to collisions. The general animation algorithm is the following: At each time step,

1. Integrate the equations of motion for the rigid components of the solids by taking external forces  $F$  and torques  $T$  into account:

$$\begin{aligned} \sum F &= m A \\ \sum T &= I \dot{\Omega} + \Omega \wedge I \Omega \end{aligned}$$

where  $I$  is the matrix of inertia of a solid computed from its rest shape.  $A$  represents linear acceleration and  $\dot{\Omega}$  angular acceleration.

2. Displace flexible components from their rest shapes.
3. Treat interactions between objects:



- (a) Detect interpenetrations.
- (b) Model contact by deforming each solid in order to generate contact surfaces.
- (c) Integrate reaction and friction forces. Add them to the set of external actions to be applied to the rigid components at the next time step.

4. Display the objects with their new deformed shapes.

This algorithm is used with an adaptive time step. As with penalty methods, overly deep interpenetrations generate overly large response forces (resulting, in the modeling contact phase, in excessive deformation of the objects). When this situation is detected, the system recomputes the objects positions using a smaller time interval.

The next section details the three steps of the interaction processing module and studies extensions to multiple collisions and to interactions with rigid implicit solids.

### 3 Interactions between Implicit Solids

#### 3.1 Interpenetration detection

We use axis-parallel bounding boxes to quickly cull most non-intersecting cases. Afterwards, we benefit from the implicit representation of the objects, as in [11]. For each pair of solids, sample points associated with one of them are tested against the inside/outside function of the other. This is done, of course, only for the sample points located inside the second solid's bounding box. As the list of solids interacting together is the only information needed for the modeling contact step, detection is stopped for a given pair of solids as soon as an interpenetration point is found.

The method used for computing sample points at each time step is detailed in Section 4. As will be shown, the detection process can be optimized by starting detection in the neighborhood of points (if any) that most penetrated the other object during the last time step.

#### 3.2 Modeling contact

Once detected, an interpenetration must be suppressed by deforming each object according to the set of interacting solids. Deforming objects involves generating contact surfaces as well as modeling the transverse propagation of deformations (see Figure 3). Rather than simulating local interactions inside the objects, the system directly computes deformed shapes at equilibrium using a model for damped propagation. Deformations outside a given "propagation area", an offset of the interpenetration zone, are ignored.

##### Interpenetration areas: Generating contact surfaces

In addition to being a very natural model to express physical properties, the implicit surface formalism is also convenient for generating exact contact surfaces. See Figure 4.

Suppose that two objects  $S_i$  and  $S_j$  interact locally. We are looking for new terms  $g_{ji}$  and  $g_{ij}$  to add to their respective field functions  $f_i$  and  $f_j$  in the interpenetration zone ( $g_{ji}$  represents the action of object  $j$  on object  $i$ ). After deformation, the objects will

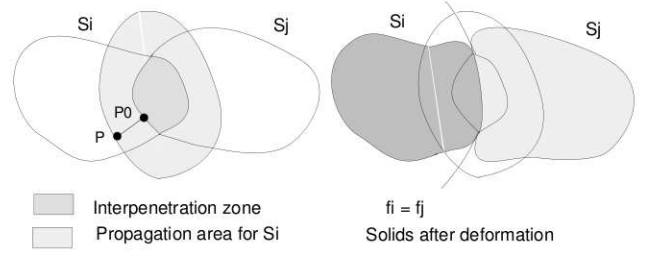


Figure 3: Modeling contact consists in applying different deformation fields in the interpenetration zone and in the "propagation area" associated with each solid (view in cross section).

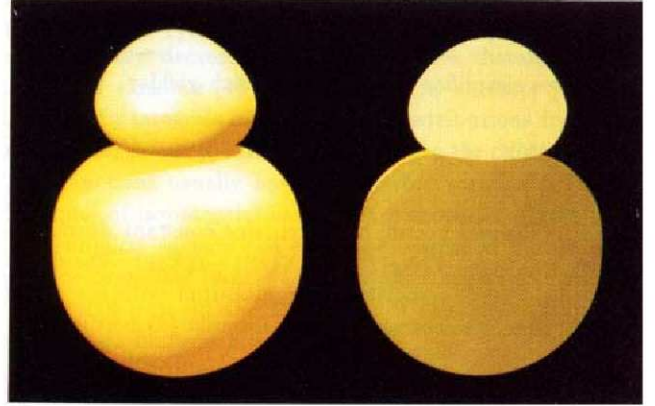


Figure 4: (left) Contact between two colliding objects. (right) View in cross section showing the exact contact modeling.

be defined in this area by:

$$f_i(P) + g_{ji}(P) = 1 \quad (6)$$

$$f_j(P) + g_{ij}(P) = 1 \quad (7)$$

The deformation fields  $g_{ji}$  and  $g_{ij}$  must be negative (they model local compression of the objects) and locally generate a contact surface, thus equations (6) and (7) must have common solutions. In order to give the new contact surface exactly the same border as the interpenetration area, deformation fields must satisfy  $g_{ij}(P) = g_{ji}(P) = 0$  in points  $P$  where  $f_i(P) = f_j(P) = 1$ . Moreover, the contact surface generated must fit with the local rigidities of colliding objects. So opposite forces must be applied by the two compressed objects on each point of this surface. Adding extra skeletons to generate deformation field terms would be inconvenient; Indeed, we prefer to directly use  $S_j$ 's skeleton to deform  $S_i$  and vice versa. Consequently, the deformation field terms of an interpenetration area are defined by:

$$g_{ji}(P) = 1 - f_j(P)$$

$$g_{ij}(P) = 1 - f_i(P)$$

With this choice, all the properties needed are verified: deformation field terms are negative in the interpenetration zone, and generate a contact surface defined by:

$$f_i(P) = f_j(P) \quad (8)$$

Let  $P \in S_i$  be a point of the contact surface,  $N_i(P)$  be  $S_i$ 's unit normal vector, and  $N_j(P) = -N_i(P)$ .

Let  $R_i(P) = ||R_i(P)||N_i(P)$  be the radial force applied by  $S_j$  at  $P$ . The correspondence (5) between forces and deformations yields :  $g_{ji}(P) = -R_i(P).N_i(P) = -||R_i(P)||$ , so:  $R_i(P) = -g_{ji}(P)N_i(P)$ . From equation (8), opposite forces are then applied by the objects along the contact surface:

$$R_i(P) = (1 - f_j(P)) N_j(P) = (1 - f_i(P)) N_j(P) = -R_i(P)$$

### Deformations in “propagation areas”

We want to optimize the contact modeling process by directly computing deformed shapes in contact positions rather than simulating local interactions inside the flexible material. Designing a purely geometric layer is justified here: only deformations along contact surfaces will be used for computing response forces. The use of geometric propagation will not affect the motion at all. Moreover, we wish to model damped material where deformations outside given “propagation areas” can be neglected. Providing the user with a set of intuitive parameters, such as the thickness of the propagation areas around interpenetration zones or the way deformations are attenuated, offers a simple and efficient control of the simulated material.

More precisely, the user controls  $S_i$ ’s propagation field term  $p_{ji}(P)$  (due to the collision with  $S_j$ ) through two additional parameters in  $S_i$ ’s description:

- A thickness value  $w_i$  giving the size of the offset were deformations propagate around an interpenetration zone. Deformations will be neglected outside this area.
- An “attenuation value”  $\alpha_i$  giving the ratio between the maximal value desired for  $p_{ji}$  and the current maximal compression term in the interpenetration area.

Because of the parameter  $\alpha_i$ , the size of the bulge due to propagation of deformations will first increase during a collision, while the solid is progressively compressed, and then decrease back to zero when the colliding objects move off.

The propagation field  $p_{ji}$  must be positive within the propagation area in order to model a local expansion of the solid, compensating for the compression due to collision. To preserve the shape’s first order continuity<sup>2</sup>  $p_{ji}$  and its derivative must become zero at the exterior limit of the propagation area, and have the same value and gradient vector as the contact term  $g_{ji}$  in the border of the interpenetration zone. Let  $P$  be a point within the propagation area, and  $P_0$  the closest point of  $S_j$  in  $S_j$ ’s gradient direction (see Figure 3). To satisfy the conditions just listed, we define  $p_{ji}$  along the line  $(P_0, P)$  by:

$$p_{ji}(P) = a_{k,a0,w_i}(d(P, P_0))$$

where  $k = ||Grad(f_j, P_0)||$ ,  $a_0$  is the maximal propagation value equal to  $\alpha_i$  times the maximal compression field value, and  $a_{k,a0,w}(x)$  is the piecewise polynomial function shown in Figure 5. An exact formula is given in Appendix B.

With this choice, all the conditions on  $p_{ji}$  can be verified. Let us prove that  $Grad(p_{ji}, P_0) = Grad(g_{ji}, P_0)$ .

$$Grad(p_{ji}, P) = a'_{k,a0,w_i}(d(P, P_0)) N_0$$

<sup>2</sup> The method would be easy to extend to higher order continuity by considering constraints over higher order derivatives, and by using more complex attenuation functions.

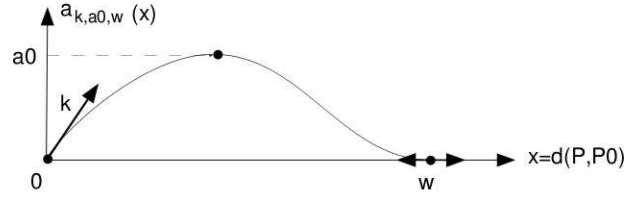


Figure 5: Attenuation function defining the propagation field.

With the value of  $a_{k,a0,w_i}$ ’s derivative in zero, we obtain:

$$Grad(p_{ji}, P_0) = -Grad(f_j, P_0) = Grad(g_{ji}, P_0)$$

Expansion of the objects in propagation areas must not produce new interpenetrations. To best avoid this situation, we insure that each deformed object does not cross the median surfaces of equation  $f_i(P) = f_j(P)$  (see Figure 3). In other words,  $p_{ji}(P)$  must be less than or equal to  $1 - f_j(P)$  throughout the propagation area. If the problem does occur, the system truncates the propagation term and issues a warning that a smaller value should be chosen for  $\alpha_i$ .

### 3.3 Computation of response forces

#### Radial reaction forces

The reaction forces directed along normal vectors are given by the correspondence (5) between forces and deformations. They are numerically integrated along contact surfaces (this process is detailed in Section 4). Because of our choice for the contact surface, the principle that opposite reactions occur on two colliding objects is verified.

#### Friction and damping forces

To model both tangential friction in contact areas and damping due to the progressive compression of the solids, we include a friction coefficient  $\lambda_i$  in the description of each object. When a collision occurs, the friction and damping force  $F_i$  at a point  $P$  of the contact surface between  $S_i$  and  $S_j$  is expressed by:

$$F_i(P) = \lambda_i \lambda_j (V_j(P) - V_i(P)) \quad (9)$$

where  $V_i(P)$  (respectively  $V_j(P)$ ) is the speed of  $P$ , a point on the surface of the solid  $S_i$  (respectively  $S_j$ ). Like radial reaction forces, friction forces are numerically integrated along contact surfaces.

Figure 6 shows the action of response forces during a few steps of an animation.

#### From collisions to lasting contacts

The deformed shapes generated during the contact modeling step can be conveniently used for lasting contacts and equilibrium states, because opposite forces are applied to each side of a contact surface. In Equation (9),  $F_i$ ’s tangential component represents friction due to the different tangential speeds of the solids at a contact point, while the normal component models the loss of energy due to the progressive deformation of the solids. The energy consumed over time enables colliding objects to settle into lasting contact situations, and then into resting stable states without unwanted oscillations. Figure 7 is an example of equilibrium state between four solids.



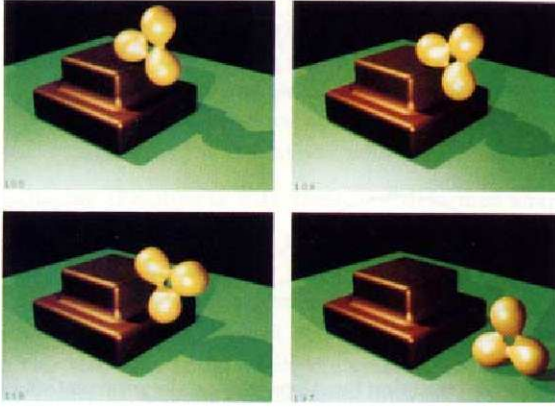


Figure 6: Flexible clover falling on a quite rigid staircase.



Figure 7: An equilibrium state between a rigid floor, a flexible vaulting horse, and two soft balls.

### 3.4 Multiple interactions

An important benefit of our model is that, in multiple interaction situations such as in Figure 7, the resulting shapes and reaction forces are *completely independent* of the order in which objects, or pairs of objects, are considered.

When an object interacts with several others, its compression field term (which produces the contact forces) is defined as a sum of terms due to the different collisions. No propagation term must be added in an interpenetration zone with another object, so, in practice, we always use a procedural method to compute field values. To evaluate the field generated by a deformed object  $S_0$  at a given point  $P$ :

1. Compute the initial field value  $f_0(P)$ .
2. For each object  $S_i$  interacting with  $S_0$ , if  $P$  lies inside  $S_i$ , add the contact deformation term  $1 - f_i(P)$ .
3. If  $P$  was not lying inside any of the  $S_i$ , compute and sum all non-zero propagation terms at  $P$ . Truncate this sum if needed (as explained at the end of Section 3.2) before adding it to the field value.

If an intersection area is detected between more than two solids as in Figure 8, several negative compression terms are simultaneously added in this area. This leaves a small space between the solids,

whose shapes remain  $C^1$  continuous (generating multiple contact points would produce singularities).

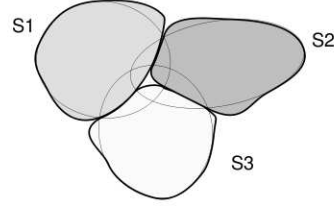


Figure 8: Deformation of 3 intersecting solids (cross sections).

### 3.5 Interactions with rigid implicit solids

Another important issue for our model is its ability to simulate interactions between flexible and rigid objects<sup>3</sup>, like the vaulting horse and the floor in Figure 7.

Suppose an interpenetration has been detected between a rigid solid  $S_j$  and a flexible object  $S_i$ . The deformation field term applied to  $S_i$  in the contact area must make  $S_i$  exactly fit  $S_j$ 's shape; i.e., the solutions of  $f_i(P) + g_{ji}(P) = 1$  must be points satisfying  $f_j(P) = 1$ . Moreover, the deformation field term must be negative in the interpenetration zone given by  $f_i(P) \geq 1, f_j(P) \geq 1$ . Then, we define  $g_{ji}(P)$  by:

$$g_{ji}(P) = (1 - f_j(P)) + (1 - f_i(P))$$

The usual formula is used for the attenuation function in  $S_i$ 's propagation area. Simply,  $S_j$ 's gradient vector (used to define the slope of the attenuation function) is now replaced by  $-(\text{Grad}(f_i + f_j, P_0))$  so that the bulge will exactly fit  $S_j$ 's normal vectors at the border of the contact surface. When this is done, response forces corresponding to  $S_i$ 's deformation are integrated along the contact surface, and opposite forces are applied to the rigid solid  $S_j$  according to the principle that opposite reactions occur on two colliding objects.

## 4 Implementation

Our modeling and animation system for implicit deformable solids is implemented in C++ on an SGI Indigo workstation. The current implementation uses distance surfaces which provide us with analytical expressions of normal vectors.

### 4.1 Optimizing the animation process

One of the main problems raised by implicit isosurfaces is the search for efficient ways to discretize objects. The animation process uses discretizations three times by animation step: for collision detection, for integrating response forces, and for displaying objects. Despite recent improvements in adaptive octree techniques, spatial partitioning polygonizations remain quite expensive. Using this type of algorithm at each time step would prevent any interactive computation and display of the animation.

<sup>3</sup> Analytical solutions such as those developed in [1] should be used for interactions between pairs of rigid objects.

Fortunately, the solids only deform locally during animations and return to their rest shapes. Their topology never changes (otherwise, our hybrid model with its invariant matrix of inertia would be invalid). Consequently, the objects need not be completely re-sampled at each time step.

Before an animation is calculated, sample points and the associated normal vectors are precomputed from the object's rest shape, and stored relative to the local coordinate system. Then, at each animation step:

- The sample points are positioned according to the current position and orientation of the solid's rigid component.
- They are used to detect collisions. To benefit from temporal coherence, tests for interpenetration are first performed in the neighborhood of points  $P_j$  which penetrated most deeply into the other object at the previous time step<sup>4</sup>.
- Finally, the sample points in deformed areas are recomputed, before display, by using a linear search algorithm along the undeformed normal direction. The use of this direction insures that the points will come back to their initial positions after any deformation.

To improve in efficiency, response forces in contact areas are integrated during this process. If a point is located in an interpenetration zone, the field function computes the deformation term, which is equal to the local reaction force. As soon as a point of the contact surface is found, this force (plus the friction force term), multiplied by the area of an elementary surface  $ds$ , is added to the sum of external actions applied to the object. In the current implementation,  $ds$  is approximated by an average value computed from the size of the discretization voxels.

## 4.2 Rendering

The implicit formalism provides us with exact high level descriptions of deformed solids, even if only a few sample points are used during the computations. During animations, we directly save the parameters needed to compute the deformed field functions defining the objects (including stiffness, scope of influence, attenuation parameters, and the current list of colliding objects). To give an idea of required disc space, the file describing the implicit objects of Figure 7 take less than 1 Kbyte, while the storage of the associated sample points with their normal vectors and the list of triangles takes more than 1200 Kbyte.

Once the objects are stored, any method could be used for rendering, including computing polygonizations with an arbitrary precision. We currently use direct ray-tracing on implicit surfaces, implemented as an extension to the public domain renderer *Rayshade* (by C. Kolb). If a ray intersects an implicit solid's bounding box, we first look for a seed point along the ray which is located inside the solid. If we find one, an intersection point is computed by binary search. Testing if a point is inside or outside is done by evaluating the solid's potential field. Normal vectors at the intersection points are analytically computed from the field gradient.

<sup>4</sup> We use a continuation method for sampling, so starting detection a few points before  $P_j$  in the list of sample points is sufficient.

## 5 Conclusion

This paper presents a novel way to model deformable solids. The implicit formalism provides a compact and unified formulation for both geometric and physical properties, and enables to keep a continuous high level representation of the objects. The model offers simple and quite general control of the simulated material. One can experiment with field function curves to adjust stiffness variations when objects are compressed, or with different ways to propagate deformations due to collisions. All the parameters are easy to understand, even for a non-specialist.

Well adapted to local deformations, the system is especially designed for a precise modeling of interactions. It generates exact contact surfaces between solids which facilitates a precise evaluation of reaction forces. The model applies to sudden collisions, lasting contacts, and equilibrium situations. It gives an elegant solution to the multiple collision problem, producing new deformed shapes and response forces which are independent of the order in which collisions are detected. The model can be generalized to treat interactions between flexible and rigid objects.

During animations, discretized representations of the objects are displayed at interactive rates, while a compact storage of their implicit description is performed. This description, which still defines curved contact surfaces, is used for producing subsequent, high-quality images.

## Future work

Implementing convolution surfaces [3] would facilitate the design of complex objects composed of homogeneous materials. To improve generality, these surfaces should be extended to non-exponential potential fields.

At present, deformed shapes only depend on the set of external forces currently applied to the solids, so deformations disappear as soon as there is no longer contact between objects. Modeling visco-elastic or elasto-plastic behaviors in addition to pure elasticity would be a good extension. Moreover, some objects of the real world conserve their volume during deformations while others are partially compressible. Modeling the propagation of deformations according to a compressibility parameter would provide easier control.

The implicit deformable model opens new directions for future research, particularly within the area of human simulation. Modeling complex objects such as deformable flesh covering rigid skeletons could not be done with previous global deformation techniques (the skeleton must not be deformed, nor the flesh on the opposite side of the skeleton). Nodal approaches can be used, but they demand complicated databases [6] and involve an expensive numerical simulation of deformations propagating in damped material. Our ability to model local deformations while preserving a compact continuous representation of objects, even when they are non-homogeneous, would be helpful. Moreover, the precise contact modeling presented here should allow human models to interact with the simulated world.



## Acknowledgements

Many thanks to Jean-Dominique Gascuel for his constant support during this research and for implementing direct ray-tracing on implicit surfaces. Thanks to Philippe Limantour, Christophe Vedel, Frederic Asensio and Julien Signes for interesting early discussions, and to Francois Sillion, Alain Chesnais, Dave Forsey, Phil Brock and Jules Bloomenthal for re-reading this paper.

## References

- [1] David Baraff. Dynamic simulation of non-penetrating rigid bodies. *PHD Thesis*, Cornell University, May 1992.
- [2] David Baraff and Andrew Witkin. Dynamic simulation of non-penetrating flexible bodies. *Computer Graphics*, 26(2):303-308, July 1992. Proceedings of SIGGRAPH'92 (Chicago, Illinois, July 1992).
- [3] Jules Bloomenthal and Ken Shoemake. Convolution surfaces. *Computer Graphics*, 25(4):251-256, July 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).
- [4] Jules Bloomenthal and Brian Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, 24(2):109-116, March 1990.
- [5] Marie-Paule Gascuel, Anne Verroust, and Claude Puech. A modeling system for complex deformable bodies suited to animation and collision processing. *Journal of Visualization and Computer Animation*, 2(3), August 1991. A shorter version of this paper appeared in *Graphics Interface'91*.
- [6] Jean-Paul Gourret, Nadia Magnenat Thalmann, and Daniel Thalmann. Simulation of object and human skin deformations in a grasping task. *Computer Graphics*, 23(3):21-29, July 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).
- [7] Annie Luciani, Stephane 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.
- [8] Dimitri Metaxas and Demetri Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. In *CVPR*, pages 337-343. IEEE Computer Society Conference, June 1991. Lahaina, Maui, Hawaii.
- [9] Gavin Miller. The motion dynamics of snakes and worms. *Computer Graphics*, 22(4):169-177, August 1988. Proceedings of SIGGRAPH'88 (Atlanta, August 1988).
- [10] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. *Computer Graphics*, 22(4):289-298, August 1988. Proceedings of SIGGRAPH'88 (Atlanta, August 1988).
- [11] Alex Pentland and John Williams. Good vibrations: Modal dynamics for graphics and animation. *Computer Graphics*, 23(3):215-222, July 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).

- [12] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205-214, July 1987. Proceedings of SIGGRAPH'87 (Anaheim, California, July 1987).
- [13] Demetri Terzopoulos and Andrew Witkin. Physically based model with rigid and deformable components. *IEEE Computer Graphics and Applications*, pages 41-51, December 1988.
- [14] Andrew Witkin and William Welch. Fast animation and control for non-rigid structures. *Computer Graphics*, 24(4):243-252, August 1990. Proceedings of SIGGRAPH'90 (Dallas, Texas, August 1990).

## Appendix A. Equation for the field functions

Field functions currently implemented are parameterized by a scope of influence  $R$ , a thickness value  $r_0$ , and a stiffness value  $k$ :

$$\begin{aligned} f_i(P) &= ar^2 + br + c & \text{if } r \in [0, r_0] \\ f_i(P) &= (r - R)^2(dr + \epsilon) & \text{if } r \in [r_0, R] \\ f_i(P) &= 0 & \text{elsewhere} \end{aligned}$$

$$d = -(k(r_0 - R) + 2)/(r_0 - R)^2, c = (kr_0(r_0 - R) + 3r_0 - R)/(r_0 - R)^2.$$

If linear elasticity is chosen,  $a = 0$ ,  $b = -k$  and  $c = kr_0 + 1$ . If non-linear elasticity is selected, we use  $a = k/(2r_0)$ ,  $b = -2k$ ,  $c = 3kr_0/2 + 1$  (of course any other stiffness variation inside the object would be easy to implement).

## Appendix B. Equation for the attenuation function

The equation we use for the attenuation function is:

$$\begin{aligned} a_{k,a0,w}(r) &= cr^3 + dr^2 + kr & \text{if } r \in [0, w/2] \\ a_{k,a0,w}(r) &= \frac{4a0 + (x - w)^2(4x - w)}{w^3} & \text{if } r \in [w/2, w] \end{aligned}$$

Where  $c = 4(wk - 4a0)/w^3$  and  $d = 4(3a0 - wk)/w^2$ .