Local Deformation for Animation of Implicit Surfaces

Agata Opalach, Marie-Paule Cani

Local Deformations for Animation of Implicit Surfaces

Agata Opalach   Marie-Paule Cani-Gascuel
iMAGIS *- GRAVIR / IMAG
BP 53, F-38041 Grenoble cedex 09, France
Agata.Opalach@imag.fr, Marie-Paule.Gascuel@imag.fr

Abstract

Implicit surfaces are well suited to modelling organic forms that consist of an internal skeleton and deformable flesh smoothly blended around it. An implicit surface can represent such an object's geometric "skin" that deforms according to the motion of the skeleton. We propose a new, simple and efficient method for calculating local deformations to be applied to implicit surfaces during collisions with other objects or between different parts of one object. We discuss applications of this method to deformable object simulation, character animation and interactive sculpture.

1 Introduction

Of growing popularity in computer graphics, implicit surfaces [16] are particularly well suited to modelling organic forms [11, 2, 14] since they generate a smooth surface around skeletons of arbitrary geometry and topology. Animation of such forms is well established [1, 6, 8] and is normally performed in a layered model, in which the internal skeleton is used to specify the global behaviour of an object and an implicit "skin" is moved and geometrically deformed according to the motion of the skeleton. A parametric surface approximating the implicit surface can be used as an additional final layer to accelerate the rendering of objects and facilitate texture mapping [15, 14].

As stressed by Dave Forsey [10], the capability of surfaces to respond to collisions with other objects in a scene by deforming locally in the contact zone is an important aspect of realism, whether the animation is calculated by a physics-based model or not. This is in particular true for organic forms for which local deformations resembling flesh behaviour need to be created. In this case we expect an object that is compressed by a collision to "inflate" locally around the contact zone to compensate the volume loss due to compression. Simulating such effect during collisions using finite elements is an expensive process [7]. Implicit surfaces offer an alternative solution: collision detection is performed efficiently due to a simple inside/outside test associated with the implicit formulation of objects, precise contact surface is created between two

*iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble et Université Joseph Fourier.

colliding objects and the amount of surface compression imposed during a collision can be used to calculate the forces to be applied to the underlying skeleton in the following animation step [6, 3]. Finally, assuming that local deformations around the contact zone are a purely visual effect without influence on the motion [9], a geometric model can be used to calculate them.

The method for creating local dilation of the surface around the contact zone proposed in [6] is complex and costly. Thus, such deformations could not be calculated during interactive animation and were added only during the final rendering stage [5]. Moreover, it required the use of expensive off-line rendering methods like ray tracing to create animation sequences.

This paper proposes a new model, simple and easy to control, for generating local deformations that visually emphasise the occurrence of collisions. Section 2 summarises the principles of implicit surfaces and their use for modelling and animation of deformable objects. Our method of local deformation of objects around the contact zone, including direction-dependent dilation, is presented in section 3. Section 4 discusses three possible applications of this work: simulation of collisions between deformable objects, character animation and interactive sculpting for object modelling.

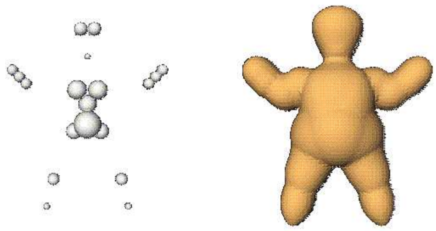

Figure 1: An example of a skeleton (point-primitives) and the generated surface

2 Implicit surfaces in animation

2.1 Skeleton-based implicit surfaces

An *implicit surface* is a surface defined as a set of points $P = (x, y, z)$ that verify an equation of type $f(P) = c$, where $f : R^3 \rightarrow R$ is a given function, called *potential function* and c is a real constant, called *isovalue*.

Skeleton-based implicit surfaces offer an intuitive definition of a potential function: a user specifies a set of geometric primitives (points, line or curve segments, surface or volume elements) that represent the internal *skeleton* of an object. The potential created by a primitive at a given point P is defined by a function $f_i(P)$ decreasing with the distance from the primitive and becoming zero at a certain distance R, called a *radius of influence*. The potential function for the modelled object is then defined as a sum of contributions from all its primitives:

$$f(P) = \sum f_i(P)$$

The resulting surface coats the skeleton and may have several disjoint components depending on the chosen potential functions f_i and the distance between the primitives (see figure 1).

2.2 Animating a layered model

The implicit surface formulation is well suited to animation, since when the skeleton moves, the generated surface follows it by definition. Various models were proposed to specify the motion of the skeleton, from a purely keyframe-based systems [1] to physics-based animation [6]. A hybrid animation approach can also be applied using both keyframes and dynamics [13].

2.3 Modelling collisions

Implicit surfaces are particularly advantageous for collision detection: a given point P is inside an implicit object if and only if $f(P) \geq c$ (the *inside/outside test*). This results in a simple detection of interpenetrations with obstacles, which is achieved by performing the inside/outside test for a given implicit object on the discretisation points of an obstacle.

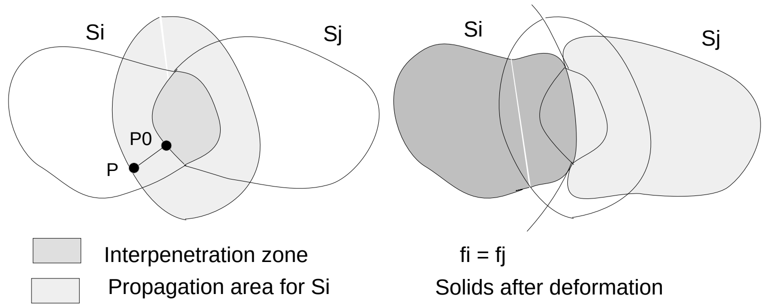

Figure 2: Deformations during a collision between two deformable objects

The deformation method proposed in [6] is as follows. Let f_1 and f_2 be the potential functions of the two objects. When an interpenetration is detected:

1. Negative compressing potentials equal respectively to $c - f_2$ and $c - f_1$ are added to potentials f_1 and f_2 of the two objects in the *interpenetration zone* (see figure 2). It results in a local compression and the creation of a contact surface given by the points satisfying the equation $f_1 = f_2$.

2. At the same time, the positive dilating potentials are applied in the *propagation zone*, defined around the interpenetration zone. These terms create local object dilation that appears around the contact surface. They are calculated so that the C^1 continuity of the deformed surface is preserved.

3. For a physics-based model, the reaction forces R_i are calculated on the contact surface. They reflect the elastic characteristics of the materials in the direction normal to the contact surface (see [6]):

$$R_1 = -R_2 = (f_2 - c)N_1$$

Since only the contact forces calculated in step 3 influence the motion of objects, purely geometric method is used to calculate the deformations in step 2. Nevertheless, the C^1 continuity constraint is difficult to satisfy: to find the deformed potential at a given point P in the propagation zone, the closest point P_0 on the interpenetration zone is found in the direction of the gradient of the surface of the other object. Then, the gradient of the compressing potential at P_0 is calculated:

$$\nabla(c - f_2)(P_0) = -\nabla(f_2)(P_0) \tag{1}$$

Finally, this direction is used as the derivative at zero for a dilating potential function. For a given point P the value of this function depends on both the distance to P_0 and the chosen size of the propagation zone.

This algorithm is extremely costly due to the search for the closest point P_0. It is therefore not applicable in a system for interactive animation. The following section proposes a new, simple and efficient method for generating the same type of local deformation.

3 A new model for local deformation

Modelling compression in the interpenetration zone (step 1 above) is relatively intuitive since the potential of each object is used to hollow out the others. We are now looking for a similarly simple formulation that will express the dilation in terms of the potentials of the objects in collision.

To ensure the C^1 continuity of the deformed surfaces, their normals have to vary continuously, in particular on the border of the interpenetration zone. Thus, at all points P_0 lying on this border, the gradient of the dilation potential has to be equal to the gradient of the compression potential. Moreover, the sought dilation potential and its derivatives have to be zero at the outside border of the propagation zone, where the dilated part joins the undeformed object.

We have chosen to express the dilation potential to be added in the propagation zone to the potential f_1 of the object 1 as:

$$b_1(f_2(P))$$

where b_1 is a function defined on the interval $[c_1, c]$ ($c_1 < c$) that satisfies:

$$b_1(c) = 0 \quad b_1'(c) = -1 \quad b_1(c_1) = 0 \quad b_1'(c_1) = 0$$

From equation 1, the gradient of the compressing potential is equal to $-\nabla f_2(P_0)$. Thus, the tangent continuity constraint is satisfied at the border of the interpenetration zone since:

$$\nabla(b_1(f_2))(P_0) = b_1'(f_2(P_0))\nabla f_2(P_0) = b_1'(c)\nabla f_2(P_0) = -\nabla f_2(P_0)$$

The parameter c_1, associated with the object 1, controls the extent of the propagation zone which is limited by the isosurface $f_2(P) = c_1$ of the object 2.

By definition of b_1, the value and derivatives of the dilation potential are zero at this limit.

In practice, the function shown in figure 3 is used for b_1. It is parameterised by its maximal height h achieved at a given point $m \in (c_1, c)$ and is defined as a union of two cubic functions defined for intervals $[c_1, m]$ and $[m, c]$.

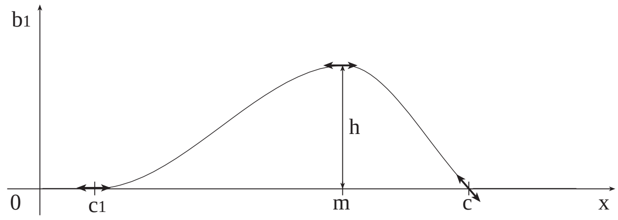

Figure 3: The function that models dilation in the propagation zone

The advantage of this formulation is its low cost: the modification of the potential at a given point P in the propagation zone only requires an evaluation of $f_2(P)$ (already known to determine the zone P belongs to) and its composition with the function b_1 given above. There is no longer the need to find the closest point on the border of the interpenetration zone. The method can therefore be used during interactive animation creation.

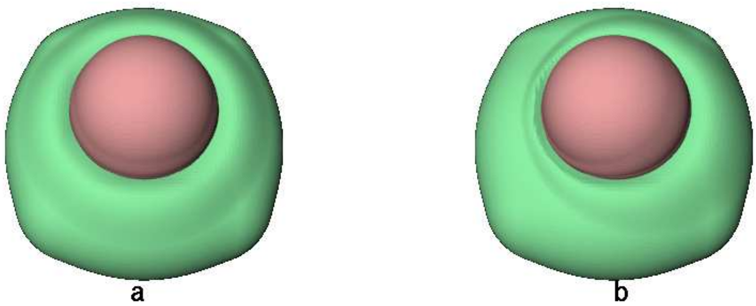

Figure 4: A sphere falls into a deformable block: (a) vertically (b) from the right

3.1 Non-uniform surface deformation

The method proposed assumes that the surface dilation around the interpenetration zone is uniform. However, this is only the case for frontal collisions. The shape of local deformations around the contact surface should depend on the relative velocities of the objects in collision. We expect the surface in front of the arriving object to form a higher bump extending for a short distance and a lower one, extending for a longer distance, behind it.

This effect can be achieved with our approach by varying the parameter c_1 that defines the size of the propagation zone. For a given point P, it will depend on the gradient of the undeformed surface 1, $\nabla f_1(P)$ and the velocity vector of the object 2, D_2. We propose the following simple expression for $c_1(P)$, based on the dot product of the two vectors:

$$c_1(P) = \frac{1}{2} c_0 \left(1 + \frac{D_2 \cdot \nabla f_1(P)}{\|D_2\| \|\nabla f_1(P)\|}\right)$$

where c_0 is a parameter specified for object 1.

Figure 4 shows two animations of a sphere that falls into a block of deformable substance. Figure 4a presents a frontal collision in which the sphere falls vertically. The resulting surface dilation is uniformly distributed around the contact zone. The collision in figure 4b occurs when the sphere arrives from the right. The surface dilation is asymmetric. In both cases local surface deformation preserves C^1 continuity of the objects and visually emphasises the collision.

3.2 Volume variation

During collision an object that is progressively compressed should proportionally enlarge its local dilation and reduce it when the object bounces away. This effect is obtained by calculating the parameter h for the function b_1 as a percentage of the maximal negative potential applied in the interpenetration zone and the size of the propagation zone.

Local dilation during a collision should ideally tend to restore the initial volume of the compressed object. A slight decrease of volume, modelling the particularly compressible character of flesh, is also a desired result. However, the deformation should not increase the global volume of the object, which is the case in figure 5 below. We are currently looking for an expression for h that may resolve this problem.

4 Applications

4.1 Simulation

Figure 5 shows three stages of a collision between two deformable spheres, calculated using an animation and simulation platform *Fabule* [4]. The compression forces applied to objects on the contact surface are passed on to the skeletons (here simple point masses) and integrated for the calculation of the bounce. This example shows the progressive character of the local dilation of objects which increases with compression.

4.2 Character animation

Local deformations proposed are purely geometric and thus can be used in a keyframe-based animation system. In this context, they increase the visual realism of animated forms emphasising the deformations of the flesh.

Using implicit surfaces for character animation requires the use of a *blending graph*, that specifies the manner of combining the contributions f_i from different parts of a character's skeleton [12, 3]. For example, an arm should be blended with the body at the shoulder while a collision should be detected between a hand and the body.

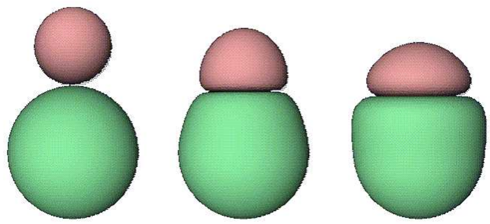

Figure 5: Three stages of a collision between two deformable spheres

Our algorithm allows us to calculate the potential at a point of space and consider the collisions and/or blending between different parts of an object as shown in figure 6.

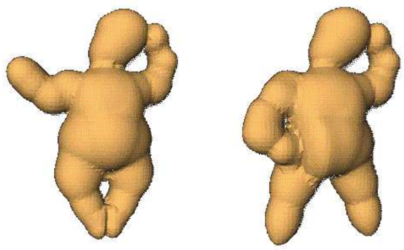

Figure 6: Local deformations during keyframe character animation

4.3 Interactive sculpture

Finally, we are planning to apply the local deformation technique presented in this paper in a system of interactive sculpture. A user will use a number of tools of different forms, modelled with implicit surfaces, to hollow out a deformable block or to leave tool-prints. Local object dilation around the contact zone will ensure C^1 continuity of obtained forms.

In order not to increase the complexity of the modelled object during the process of sculpting (which would be the case if all deformations applied to the model need to be stored), we propose to use a discrete representation of the potential function that defines the deformable block. The discrete values will be progressively updated to model the actions of the tools.

5 Conclusion

We have presented a new method to model local deformations of implicit objects during collisions. This method is well suited to modelling and animation of organic forms for which local dilation appears around contact zones. The deformation is purely geometric which results in an inexpensive algorithm that can be used in an interactive animation system. The work in progress includes achieving better volume control and the use of the technique for interactive sculpture.

References

[1] Thaddeus Beier. Practical uses for implicit surfaces in animation. In *Modeling, Visualizing and Animating with Implicit Surfaces (SIGGRAPH'93 course notes Number 25)*, Anaheim, CA, August 1993.

[2] Jules Bloomenthal. Skeletal design of natural forms. *PHD Thesis*, The University of Calgary, January 1995.

[3] Marie-Paule Cani-Gascuel and Mathieu Desbrun. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):39–50, March 1997.

[4] Jean-Dominique Gascuel. Fabule : un environnement de recherche pour l'animation et la simulation. In *Les Simulateurs, Troisième Séminaire du groupe de travail français Animation et Simulation*, October 1994.

[5] Jean-Dominique Gascuel. Implicit patches: An optimized and powerful ray intersection algorithm for implicit surfaces. In *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, pages 143–159, Grenoble, France, April 1995.

[6] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics*, 27:313–320, August 1993. Proceedings of SIGGRAPH'93 (Anaheim, CA).

[7] 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).

[8] Thierry Guiard-Marigny, Nicolas Tsingos, Ali Adjoudani, Christian Benoît, and Marie-Paule Gascuel. 3d models of the lips for realistic speech animation. In *Computer Animation'96*, pages 80–89, Geneva, Switzerland, June 1996.

[9] Arash Habibi. Visualisation d'objets très déformables : relations mouvement - forme - image. *Thèse de troisième cycle*, INPG, January 1997.

[10] Jason Harrison and David R. Forsey. A kinematic model for collision response. In *Fifth Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994.

[11] Shigeru Muraki. Volumetric shape description of range data using blobby model. *Computer Graphics*, 25(4):227–235, July 1991.

[12] Agata Opalach and Steve Maddock. Implicit surfaces: Appearance, blending and consistency. In *Fourth Eurographics Workshop on Animation and Simulation*, pages 233–245, Barcelona, Spain, September 1993.

[13] Agata Opalach and Steve Maddock. Disney effects with implicit surfaces. In *5th Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994.

[14] Jianhua Shen. Human body modeling and deformation. *PHD Thesis*, EPFL, Lausanne, Switzerland, September 1996.

[15] Russel Turner. Leman: A system for construsting and animating layered elastic characters. In *Computer Graphics- Developments in Virtual Environments*, pages 185–203, Academic Press, San Diego, CA, June 1995.

[16] Brian Wyvill and Jules Bloomenthal, editors. *Modeling and Animating with Implicit Surfaces*, Dallas, USA, August 1990. SIGGRAPH Course Notes No 23.