Adaptive Sampling of Implicit Surfaces for Interactive Modeling and Animation

Mathieu Desbrun
Nicolas Tsingos
Marie-Paule Gascuel

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

email: Mathieu.Desbrun@imag.fr

Abstract

This paper presents a new adaptive sampling method for implicit surfaces that can been used in both interactive modeling and animation. The algorithm samples implicit objects generated by skeletons and efficiently maintains this sampling, even when their topology changes over time such as during fractures and fusions. It provides two complementary modes of immediate visualization: displaying "scales" lying on the surface, or a piecewise polygonization. The sampling method is particularly well suited to efficiently avoid "unwanted blending" between different parts of an object. Moreover, it can be used for partitioning the implicit surface into local bounding boxes that will accelerate collision detection during animations and ray-intersections during final rendering.

1 Introduction

Implicit formulations of surfaces [Bar81, WMW86b] have drawn a lot of attention during the past few years. Particularly convenient for modeling smooth objects of any topology and geometry such as objects with holes and branchings [WW89, BW90], they offer a compact storage of complex shapes and are well suited to direct ray-tracing algorithms [Bli82]. They appear as very promising for animating deformable solids, and have successfully been used in descriptive animation systems [WMW86a, OM93], for animating metamorphosis [Wyv93], and for physically-based animation [PW89, TM91, Gas93, DG94, OM94] where their inside/outside function accelerates collision detection.

However, interactive modeling and animation with implicit surfaces has been limited by the difficulty of maintaining sample points on the surfaces at interactive rates. These points are necessary for visualizing implicit objects, and are also needed for detecting collisions and integrating response forces during physically-based animations [PW89, Gas93]. Insuring an adequate repartition of samples on the surface is important for both applications. But the problem is difficult since the surface topology may change over time: deformations, possible appearance of holes, fractures, and fusions at interactive rates requires being able to efficiently create and suppress sample points in the adequate ar-

[1]iMAGIS is a joint project between CNRS, INRIA, Institut National Polytechnique de Grenoble, and Université Joseph Fourier.

eas. This paper presents a new solution to this specific problem, and presents some interesting side effects of the method.

Background

Most of the existing techniques for sampling implicit surfaces are based on spatial partitioning polygonization. The first algorithm of this kind, known as *marching cubes*, was introduced in [WMW86b], and later developed for medical imagery [LC87]. It consists in using an uniform grid to discretize space, find a cube that intersects the surface, and follow the surface by examining intersections with neighboring cubes. A set of polygons that approximate the surface is associated with each intersecting cube. More recent algorithms accelerate this process by the use of octrees, and focus on topological ambiguities [Blo88, WG92, NB93]. However, polygonization methods have a major drawback: they are compute-intensive so they cannot be performed interactively.

A completely different approach consists in scattering seed-points in space, and leaving them migrate to the implicit surface along the gradient direction [BW90]. Here, computations are much more efficient since no polygonization is computed.

Providing an uniform repartition of sample points on an implicit surface can be done by first using one of the previous methods for sampling the surface and then connecting these points with interaction forces. A physically-based simulation of this "particle system" leads to an equilibrium state where the repartition of samples is very good [dFdMGTV92]. A further improvement consists in introducing fissionable particles of adaptive size (where the size corresponds to the radius of the non-penetration area around a particle). After each deformation of the implicit surface, large particles are used for very quickly invading low sampled areas, reach equilibrium, and then subdivide until the desired precision is obtained. A constrained optimization process is used for maintaining particles on the surface at each iteration [WH94]. However, interactively adding and suppressing skeletons is impossible in the current implementation of this method: the number and nature of the degrees of freedom for the set of implicit primitive must be known in advance. This can be a problem for interactive design applications.

Designing a sampling method that is convenient for both interactive modelling and animation should rely on two major criteria:

- First of all, interactivity must be ensured. In particular, the method should take benefits of temporal coherence for accelerating the sampling of a surface that progressively moves and deforms. This does not seem easy to achieve with spatial partitioning polygonalyzations.

- Another important criterion is the legibility the sampling algorithm will offer for implicit surface visualization. Here polygonization techniques seem much more convenient than methods that compute a mere set of sample points. Computing points without any polygonization can be combined with the visualization of "scales", flat primitives centered on the sample points and oriented according to the surface tangent plane [WH94]. This gives a reasonably good idea on the shape of the object. However, it still doesn't provide any opaque visualization of implicit surfaces, which can be a problem for designing complex shapes and for displaying animations of colliding objects.

Overview

This paper presents a new and efficient algorithm for computing and maintaining sample points on implicit surfaces whose topology may change over time. The method takes benefits of temporal coherence, which is important in both animation and modeling fields (a designer drags primitives during a modelling process). Computed at interactive rates, the set of sample points enables an opaque display of the implicit surface thanks to a piecewise polygonization. Moreover, the method is well suited to the elimination of the unwanted blending problem (ie. modeling implicit objects with different parts that do not blend together), and enables to cover the surface with local bounding boxes that can be used for accelerating collision detection and ray-intersection algorithms.

Section 2 details the sampling algorithm. Section 3 focuses on interactive visualization of the surfaces, including opaque display. Sections 4 and 5 show how one can take advantage of the sampling method for accelerating unwanted blending avoidance and intersection computations. We conclude in Section 6.

2 An Efficient Adaptive Sampling Algorithm

Principle of the method

As pointed out in the introduction, performing a polygonal partitioning in real time seems impossible. A good solution for taking advantage of temporal coherence is to compute a set of points that migrate to the surface, as in the scattering method [BW90]. However, there is a severe drawback in recomputing seeds along the gradient direction: the repartition of sample points on the implicit surface would become far from uniform after several deformations, and a topological change such as a fracture would require scattering points again, with the problem that there is no easy way to detect the fracture. So our principles are the following:

- Rather than creating and suppressing particles as in [WH94], we associate a fixed set of points, called "seeds" to each skeleton (or primitive) generating the iso-surface. This set is quickly re-computed when skeletons are enlarged during interactive modelling so that the sampling density remains constant.
- Each seed migrates to the surface along an axis that is fixed in the skeleton/primitive's local coordinate system. The use of a fixed axis rather than the gradient direction as in [BW90] has two benefits. Firstly, we don't need any intermediate gradient evaluation during seeds migrations.

Secondly, a rigid motion of an implicit object during an animation will not require any seed re-computation, since they are stored in local referentials.

- During deformations and topological changes, a very simple validation/invalidation process insures a convenient, although non-uniform, repartition of seed points on the surface.

The remainder of the paper describes the method in the terminology of implicit surfaces generated by skeletons [Bli82, NHK+85, WMW86b, BW90, BS91]. However, the method can be generalized to any other kind of blended implicit primitives, by associating a specific initialization procedure with each of them.

In our terminology, the skeletons S_i can be any convex geometric primitive with a well defined distance function. S_i generates a scalar field f_i that decreases with the distance. The surface is an iso-surface for the sum of skeleton contributions:

$$\mathit{Surface} = \{P \;/\; f(P) = iso\} \quad \text{with: } f(P) = \sum f_i(P)$$

where iso is a given isovalue. The surface normal is given by the field's gradient.

Initialization

The first step of the sampling algorithm consists in generating a set of axes that are well distributed around each skeleton: these axes will be used as seeds migration directions during the interactive modeling or animation process.

Any method could be used for initializing the axes. We are currently using the following process (see Figure 1) since we only use convex primitives:

1. We first extend the skeleton's bounding box by the thickness $e_i = f_i^{-1}(iso)$, that represents the thickness of the implicit volume when the skeleton is used alone.

2. We calculate a regular grid on the surface of this box that fits the desired sampling density, specified by the user.

3. We project each grid vertex on the skeleton to find the directions of the axis. A seed is initialized on each axis at the distance e from the skeleton.

This process does not provide an exact uniform repartition of axis directions around a skeleton, but is has the advantage of being simple. Anyway, a uniform axis repartition around a skeleton would not lead to a uniform sampling of the surface, due to the blending between several skeleton contributions.

Seeds migration

After each deformation due to interactive modeling or animation, seeds migrate along their axis to reach the iso-surface. This is done by finding an inner and an outer point along the axis (which can be done very efficiently due to temporal coherence), and then using any root finding method, as for instance Regula Falsi [PTVF92], for finding the new seed position.

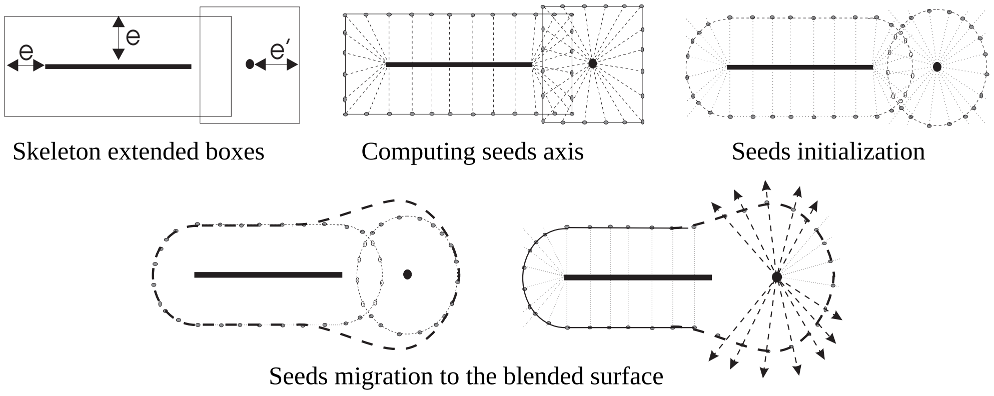

Figure 1: From initialization to migration: different stages of sampling

Validation/Invalidation process

When several skeletons are blending together, seeds that penetrate into an area already sampled by another skeleton are not useful for the current surface sampling (see Figure 1). However, these seeds should not be suppressed, since they may be needed later if the object happens to break into pieces. We choose to invalidate them for a while, according to the following criterion:

A seed located at P is invalid if the field value $f_i(P)$ generated by its associated skeleton s_i is smaller than another field contribution.

That is to say:

$$\{\exists j \neq i \mid f_j(P) > f_i(P)\}$$

Thanks to this criterion, only some of the seeds points migrate to the surface at each time step, which saves computations, and their repartition is sufficient for our applications, although not uniform (see Figure 2). During deformations, invalid seeds may be validated again, so the sampling process efficiently adapts to geometrical and topological changes.

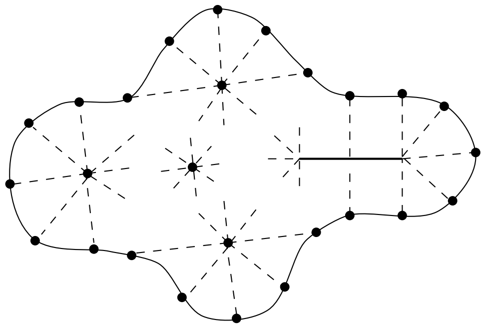

Figure 2: Valid seeds on an object defined by 5 skeletons.

Discussion

We obtain interactive sampling even for quite complex scenes as the one depicted in Figure 5. Since our method does not compute and apply sets of interaction forces on each sample point, it appears to to be more efficient than physically based sampling methods. Moreover, our algorithm enables the addition or suppression of skeletons during interactive modeling without perturbating the sampling process, since each skeleton provides its own set of seed points. However, we stated earlier that this method does not result in a uniform samples distribution over the surface. Although sufficient for our applications, the distribution cannot be compared with the impressive result of [WH94] that uses particle system simulations and constrained optimization techniques to perfect samples distribution.

The following section describes a method for opaque visualization that is a specific benefit of our approach.

3 Interactive Visualization

Although well-suited to ray tracing, implicit surfaces cannot be rendered interactively at a high quality level. Cruder representations based on sample points need to be defined. We are currently using two complementary modes of interactive display that give good idea of the surface's shape.

Displaying scales

Visualizing a mere set of points is far from sufficient for giving a good idea of a surface. In [WH94], discs are placed at sample points with their normals aligned to the surface normals. Our system offers the same kind of representation: any graphic primitives - called scales for the analogy with fish scales - are placed on valid seed positions and oriented according to the surface normal. This gives a good idea of the object shape while demanding very few extra calculation and display time.

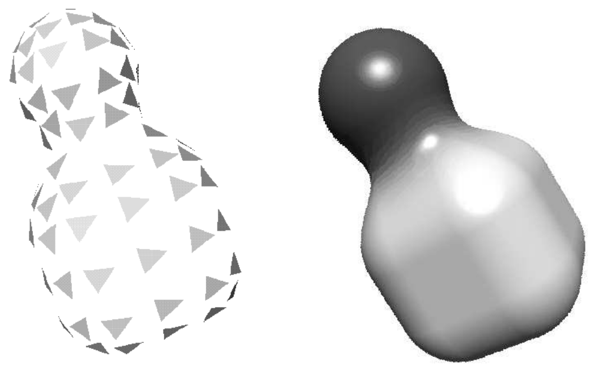

Figure 3: The same object with scales or ray-traced.

However, this visualization mode presents a major drawback in the case of complex scenes, and particularly for those, very frequent in animation, that represent objects in contact: dissociating objects that are in the foreground and objects in the background can become very difficult. In these situations, an opaque visualization is more convenient.

Using a piecewise polygonization

We propose a second visualization mode, that is specific to our sampling technique, and offers an opaque visualization of the surface without any extra computation.

Rather that computing a Delaunay triangulation at each time step for connecting valid seeds from different skeletons, which would be compute intensive, we display a piecewise polygonization. During the sampling initialization process, a closed set of polygons constructed on seed points is associated with each skeleton. The topology of these meshes is given by the grids used for computing migration axis. During surface deformations, each mesh, based on both valid and invalid seeds, surrounds the implicit volume where a given skeleton is preponderant. The resulting opaque visualization is shown in Figure 4. The fact that the polygonization used is piecewise generates artifacts such as discontinuities in the surface in the area where two skeletons blend together, but this is not a problem since the user can always switch to scales display to get a better idea of normal vectors in this particular area. Another example of visualization with the piecewise polygonization is given in Figure 5. Polygons are displayed with transparency in order to show the skeletons generating the implicit surfaces.

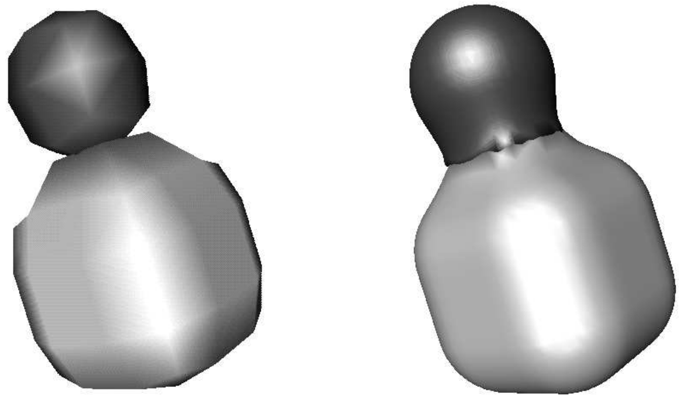

Figure 4: Two piecewise opaque visualizations with different sampling densities for an object generated by two skeletons: a point and a box.

4 Avoiding Unwanted Blending

The unwanted blending case is a difficult problem known for a long time [WW89]. For modeling implicitly defined characters, for instance, arms should blend with shoulders, but not with another part of the body (Figure 6).

A solution, that was suggested in [WW89] and further developed in [OM93, GW95], consists in defining a neighboring graph between the different skeletons, and stating that a skeleton's field only blends with contributions from neighboring skeletons. More precisely, we are currently using the following procedure

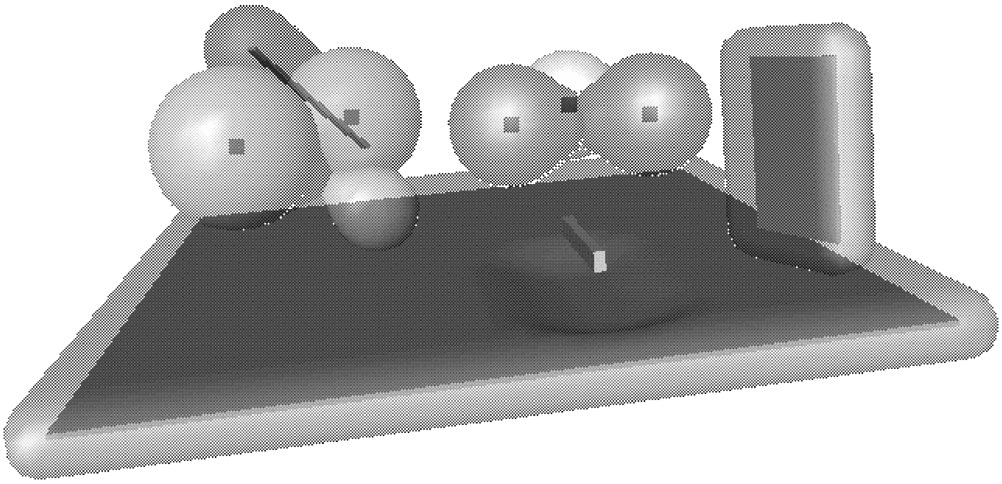

Figure 5: Snapshot from our interactive modeling system: piecewise polygonization generated by our sampling method is displayed with transparency, in order to show the skeletons generating the surface.

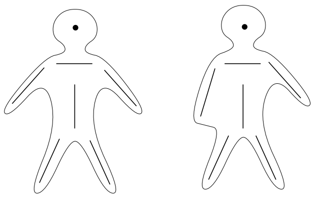

Figure 6: The unwanted blending problem

for computing the field function at point P:

1. Compute all the field contributions at point P,
2. select the field f_i that is preponderant,
3. add the maximal contribution from groups of skeletons that blend together and are all neighbors to S_i in the graph[2],
4. return the resulting value.

Taking unwanted blending into account during interactive modeling and animation requires using this new procedural field function for recomputing sample points. However, computing all the field contributions at every seed point for only using the preponderant and neighboring fields would be compute intensive. Our specific adaptive sampling algorithm provides a much more efficient answer to this problem.

[2]Taking the maximum ensures the C^0 continuity of the surface [GW95]

Sampling surfaces according to neighboring lists

We use the same initial idea as in [WW89, OM93] to cope with the problem: we define the blending graph by associating to each skeleton a list of neighbors to blend with.

Fortunately, it happens that finding the preponderant field is made significantly easier when using our sampling algorithm. Indeed, each seed is sent by a particular skeleton and samples the area where the field of this skeleton is the highest. So we already know which is the preponderant field associated with any valid seed. This makes the computations very efficient: we just have to follow the list of neighboring skeletons and add their contribution. Invalidation tests are made during the same process (the seed is invalidated if one of the neighboring contribution is higher that the first one).

Thanks to this algorithm, we are able to sample complex implicit objects, such as the articulated string depicted in Figure 7: the two extremities of this string do not blend together, but are able to collide during a physically-based simulation.

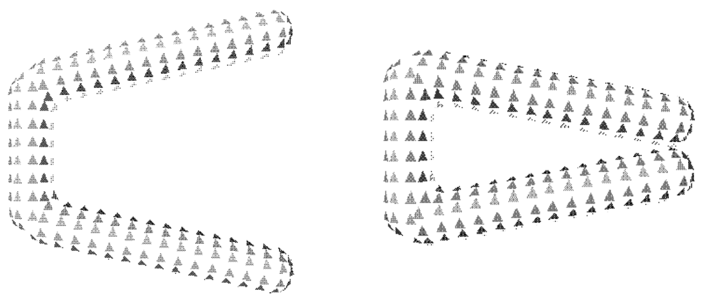

Figure 7: An articulated string whose extremities do not blend together

Benefits for further optimizations

In fact, storing lists of neighboring skeletons that define blending properties can be used for a more general purpose. Even if no unwanted blending property is needed, maintaining neighboring lists is a way to optimize computations.

After each deformation of an implicit surface, many seed points need to be recomputed. This can be done more efficiently if the list of skeletons that have a non-zero influence in the area is already known. We do this by computing current lists of neighboring skeletons: after each deformation, the skeletons whose influence areas do not overlap are suppressed from neighbors lists. Then current neighboring lists are used for every field computation at seed points, avoiding distance evaluations with distant skeletons.

5 Surface pavement with local bounding boxes

A last benefit of the adaptive sampling algorithm we have developed is to enable the definition of a hierarchy of bounding boxes that can be used for optimizing both collision processing and ray-intersection algorithms.

Local bounding boxes

The basics of the sampling algorithm was to see the implicit objects as an union of areas controlled by specific skeletons. This can be further exploited by associating an axis-parallel bounding box with the *valid seeds* of each skeleton. Computed from the seed positions, these boxes are enlarged by the maximal distance between sample points (a function of the sampling density defined by the user and of the blending properties of the implicit primitives).

These local boxes provide a precise pavement of the implicit surface (they don't necessarily cover the implicit volume), and are used to compute a main bounding box that includes the entire surface. This two-level hierarchy of boxes, depicted in Figure 8, can be used for optimizing intersection tests.

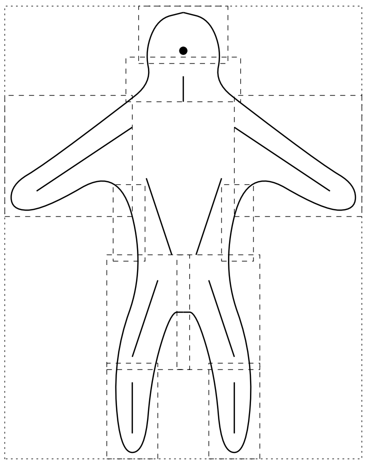

Figure 8: The hierarchy of bounding boxes associated with a surface

Application to collision detection

Local bounding boxes, that better fit the surface than the large one, can be used for optimizing collision detection during physically-based simulations. In particular, they unable to greatly reduce the number of field evaluations (see [DG94]).

The algorithm for detecting inter-penetrations between two implicitly defined solids I_1 and I_2 is: if large bounding boxes B_1 and B_2 intersect, then, for each local box b_1 of I_1 that intersects B_2, if b_1 intersects at least one of I_2's local boxes, then test the seed points that have been used to define b_1 against I_2's field function. An inter-penetration is detected as soon as I_2's field on one of these seeds is greater than the isovalue.

Application to ray-tracing

Final high-quality rendering on an implicit surface can be computed by processing direct ray-tracing, ray intersections with the surface being computed by a Regula Falsi-like method.

In practice, we store the hierarchy of bounding boxes provided by our sampling algorithm together with the parameters that define the implicit surface (ie. positions of the skeletons, field parameters, and neighboring lists defining the blending properties). This accelerates both ray-intersection computations, and field evaluations, since a good candidate to be the "preponderant field" is known when a ray intersects a given local box. This leads to an optimized version of ray-tracing that respects unwanted blending avoidance [Gas95].

6 Conclusion

This paper has presented a new algorithm for adaptively sampling implicit surfaces that deform over time. The method takes benefits of temporal coherence, and provides an opaque visualization of the objects. Applications include both modeling and animation/simulation fields.

Objects are re-sampled at interactive rates after any deformation including addition/suppression of skeletons. This is essential for interactive modelling. Moreover, objects with specific unwanted blending properties can be designed as well, and stored with adequate local bounding boxes for final rendering.

During interactive animations, the opaque piecewise display proves to be very convenient for observing collision and contact situations [Gas93, DG94]. Local boxes associated with each skeleton enable to accelerate collision detection, and also permit collision processing between different parts of the same object, for which non-blending properties have been specified.

Work in progress includes attempts to improve the sampling repartition, which is good but not uniform. Sampling irregularities are not a real problem in modeling applications, but can lead to imprecise results during physically-based simulations, where seed points in contact areas are used for integrating response forces [Gas93]. Also, the method described for initialization can only be used with convex primitives as points, segments or facets. When dealing with more complex skeletons, care must be taken during the selection of axes directions. When possible, a reasonable sampling could be obtained by splitting the skeletons into convex components during sampling initialization.

Acknowledgements

We would like to thank Jean-Dominique Gascuel for the multiple discussions and for his constant help during the implementation. Many thanks to Gershon Elber for carefully re-reading this paper and for his helpful comments.

References

[Bar81] Alan H. Barr. Superquadrics and angle-preserving transforms. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[Bli82] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, pages 235–256, July 1982.

[Blo88] Jules Bloomenthal. Polygonisation of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1988.

[BS91] Jules Bloomenthal and Ken Shoemake. Convolution surfaces. *Computer Graphics*, 25(4):251–256, July 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).

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

[dFdMGTV92] Luiz Henrique de Figueiros, Jonas de Miranda Gomez, Demetri Terzopoulos, and Luiz Velho. Physically-based methods for polygonization of implicit surfaces. In *Graphics Interface'92*, pages 250–257, Vancouver, Canada, May 1992.

[DG94] 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.

[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.

[Gas95] Jean-Dominique Gascuel. Implicit patches: An optimized and powerful ray intersection algorithm for implicit surfaces. In *Implicit Surfaces'95*, Grenoble, France, April 1995.

[GW95] Andrew Guy and Brian Wyvill. Controlled blending for implicit surfaces using a graph. In *1st Eurographics Workshop on Implicit Surfaces*, April 1995.

[LC87] William Lorensen and Harvey Cline. Marching cubes: a high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987. Proceedings of SIGGRAPH'87 (Anaheim, California, July 1987).

[NB93] Paul Ning and Jules Bloomenthal. An evaluation of implicit surface tilers. *IEEE Computer Graphics and Applications*, 13(6), November 1993.

[NHK+85] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Objects modeling by distribution function and a method of image generation (in japanese). *The Transactions of the Institute of Electronics and Communication Engineers of Japan*, J68-D(4):718–725, 1985.

[OM93] Agata Opalach and Steve Maddock. Implicit surfaces: Appearance, blending and consistency. In *Fourth Eurographics Workshop on Animation and Simulation*, Barcelona, Spain, 1993.

[OM94] Agata Opalach and Steve Maddock. Disney effects with implicit surfaces. In *5th Eurographics Workshop on Animation and Simulation*, Oslo, Norway, September 1994.

[PTVF92] William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. *Numerical Recipes in C, second edition.* Cambridge University Press, New York, USA, 1992.

[PW89] 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).

[TM91] Demetri Terzopoulos and Dimitri Metaxas. Dynamic 3-D models with local and global deformations : deformable super quadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(7):703–714, July 1991.

[WG92] J. Wilhelms and A. Van Gelder. Octree for faster isosurface generation. *Transactions on Graphics*, 11(3):210–227, July 1992.

[WH94] Andrew Witkin and Paul Heckbert. Using particles to sample and control implicit surfaces. *Computer Graphics*, pages 269–278, July 1994. Proceedings of SIGGRAPH'94.

[WMW86a] B. Wyvill, C. McPheeters, and G. Wyvill. Animating soft objects. *The Visual Computer*, pages 235–242, August 1986.

[WMW86b] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for soft objects. *The Visual Computer*, pages 227–234, August 1986.

[WW89] Brian Wyvill and Geoff Wyvill. Field functions for implicit surfaces. *The Visual Computer*, 5:75–82, December 1989.

[Wyv93] Brian Wyvill. Metamorphosis of implicit surfaces. *Modeling, Visualizing, and Animating with Implicit Surfaces (SIGGRAPH'93 course notes Number 25, Anaheim, CA, USA)*, August 1993.