Level Set Surface Editing Operators

Ken Museth* David E. Breen* Ross T. Whitaker† Alan H. Barr*

*Computer Science Department
California Institute of Technology

†School of Computing
University of Utah

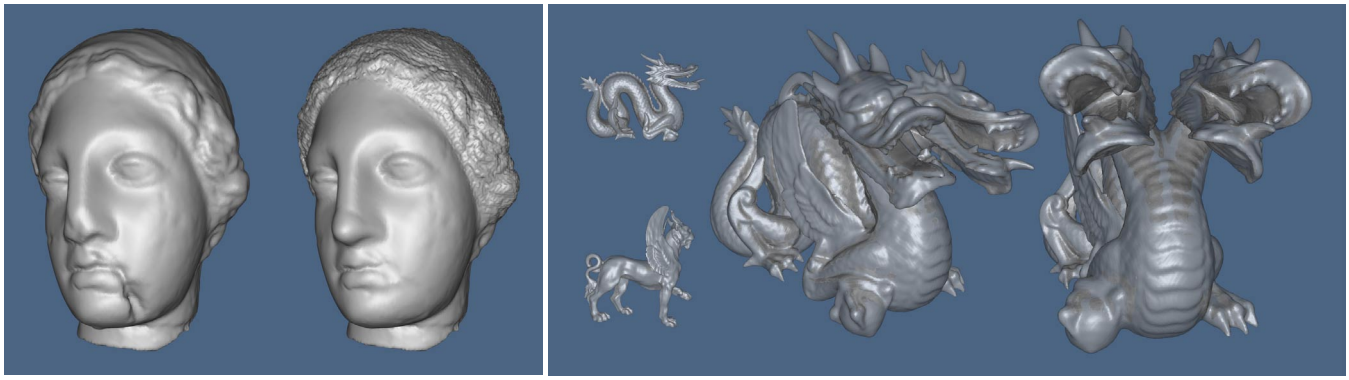


Figure 1: Surfaces edited with level set operators. Left: A damaged Greek bust model is repaired with a new nose, chin and sharpened hair. Right: An edited model is constructed from models of a griffin and dragon (small figures), producing a two-headed, winged dragon.

Abstract

We present a level set framework for implementing editing operators for surfaces. Level set models are deformable implicit surfaces where the deformation of the surface is controlled by a speed function in the level set partial differential equation. In this paper we define a collection of speed functions that produce a set of surface editing operators. The speed functions describe the velocity at each point on the evolving surface in the direction of the surface normal. All of the information needed to deform a surface is encapsulated in the speed function, providing a simple, unified computational framework. The user combines pre-defined building blocks to create the desired speed function. The surface editing operators are quickly computed and may be applied both regionally and globally. The level set framework offers several advantages. 1) By construction, self-intersection cannot occur, which guarantees the generation of physically-realizable, simple, closed surfaces. 2) Level set models easily change topological genus, and 3) are free of the edge connectivity and mesh quality problems associated with mesh models. We present five examples of surface editing operators: blending, smoothing, sharpening, openings/closings and embossing. We demonstrate their effectiveness on several scanned objects and scan-converted models.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Surface and object representations; I.3.4 [Computer Graphics]: Graphics Utilities—Graphics Editors;

Keywords: Deformations, geometric modeling, implicit surfaces, shape blending.

1 Introduction

The creation of complex models for such applications as movie special effects, graphic arts, and computer-aided design can be a time-consuming, tedious, and error-prone process. One of the solutions to the model creation problem is 3D photography [Bouguet and Perona 1999], i.e. scanning a 3D object directly into a digital representation. However, the scanned model is rarely in a final desired form. The scanning process is imperfect and introduces errors and artifacts, or the object itself may be flawed.

3D scans can be converted to polygonal and parametric surface meshes [Edelsbrunner and Mücke 1994; Bajaj et al. 1995; Amenta et al. 1998]. Many algorithms and systems for editing these polygonal and parametric surfaces have been developed [Cohen et al. 2001], but surface mesh editing has its limitations and must address several difficult issues. For example, it is difficult to guarantee that a mesh model will not self-intersect when performing a local editing operation based on the movement of vertices or control points, producing non-physical, invalid results. See Figure 2. If self-intersection occurs, it must be fixed as a post-process. Also, when merging two mesh models the process of clipping individual polygons and patches may produce errors when the elements are small and/or thin, or if the elements are almost parallel. In addition while it is not impossible to change the genus of a surface mesh model [Biermann et al. 2001], it is certainly difficult and requires significant effort to maintain the consistency/validity of the underlying vertex/edge connectivity structure.

1.1 New Surface Editing Operators

In order to overcome these difficulties we present a level set approach to implementing operators for locally and globally editing closed surfaces. Level set models are deformable implicit surfaces that have a volumetric representation [Osher and Sethian 1988]. They are defined as an iso-surface, i.e. a level set, of some implicit function ϕ. The surface is deformed by solving a partial differential equation (PDE) on a regular sampling of ϕ, i.e. a volume dataset. To date level set methods have not been developed for adaptive grids, a limitation of current implementations, but not of the mathematics. It should be emphasized that level set methods do

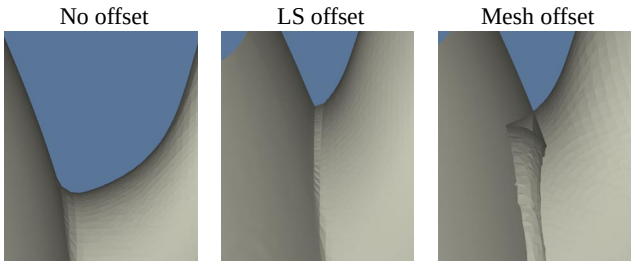


Figure 2: (left) A cross-section of the teapot model near the spout. (middle) No self-intersection occurs, by construction, when performing a level set (LS) offset, *i.e.* dilation, of the surface. (right) Self-intersections may occur when offsetting a mesh model.

not manipulate an explicit closed form representation of ϕ, but only a sampling of it. Level set methods provide the techniques needed to change the voxel values of the volume in a way that deforms the embedded iso-surface to meet a user-defined goal. The user controls the deformation of the level set surface by defining a speed function $\mathcal{F}(\boldsymbol{x}, \ldots)$, the speed of the level set at point $\boldsymbol{x}$ in the direction of the normal to the surface at $\boldsymbol{x}$. Therefore all the information needed to deform a level set model may be encapsulated in a single speed function $\mathcal{F}()$, providing a simple, unified computational framework.

We have developed a number of surface editing operators within the level set framework by defining a collection of new level set speed functions. The cut-and-paste operator (Section 5.1) gives the user the ability to copy, remove and merge level set models (using volumetric CSG operations) and automatically blends the intersection regions (See Section 5.2). Our smoothing operator allows a user to define a region of interest and smooths the enclosed surface to a user-defined curvature value. See Section 5.3. We have also developed a point-attraction operator. See Section 5.4. Here, a regionally constrained portion of a level set surface is attracted to a single point. By defining line segments, curves, polygons, patches and 3D objects as densely sampled point sets, the single point attraction operator may be combined to produce a more general surface embossing operator. As noted by others, the opening and closing morphological operators may be implemented in a level set framework [Sapiro et al. 1993; Maragos 1996]. We have also found them useful for performing global blending (closing) and smoothing (opening) on level set models. Since all of the operators accept and produce the same volumetric representation of closed surfaces, the operators may be applied repeatedly to produce a series of surface editing operations. See Figure 11.

1.2 Benefits and Issues

Performing surface editing operations within a level set framework provides several advantages and benefits. Many types of surfaces may be imported into the framework as a distance volume, a volume dataset that stores the signed shortest distance to the surface at each voxel. This allows a number of different types of surfaces to be modified with a single, powerful procedure. By construction, the framework always produces non-self-intersecting surfaces that represent physically-realizable objects, an important issue in computer-aided design. Level set models easily change topological genus, and are free of the edge connectivity and mesh quality problems associated with deforming and modifying mesh models. Additionally, some reconstruction algorithms produce volumetric models [Curless and Levoy 1996; Whitaker 1998; Zhao et al. 2001] and volumetric scanning systems are increasingly being employed in a number of diverse fields. Therefore volumetric models are becoming more prevalent and there is a need to develop powerful editing operators that act on these types of models directly.

There are implementation issues to be addressed when using level set models. Given their volumetric representation, one may be concerned about the amount of computation time and memory needed to process level set models. Techniques have been developed to limit level set computations to only a narrow band around the level set of interest [Adalsteinsson and Sethian 1995; Whitaker 1998; Peng et al. 1999] making the computational complexity proportional to the surface area of the model. We have also developed computational techniques that allow us to perform the narrow band calculations only in a portion of the volume where the level set is actually moving. Additionally, fast marching methods have been developed to rapidly evaluate the level set equation under certain circumstances [Tsitsiklis 1995; Sethian 1996]. Memory usage has not been an issue when generating the results in this paper. The memory needed for our results (512 MB) is available on standard workstations and PCs. We have implemented our operators in an interactive environment that allows us to easily edit a number of complex surfaces. Additionally, concerns have been raised that volume-based models cannot represent fine or sharp features. Recent advances [Frisken et al. 2000; Kobbelt et al. 2001] have shown that is is possible to model these kinds of structures with volume datasets, without excessively sampling the whole volume. These advances will also be available for our operators once adaptive level set methods, an active research area, are developed.

1.3 Contributions

The major contributions of our work are the following.

- The introduction of a unified approach to surface editing within a level set framework.
  - Editing operators defined by speed functions.
  - Results produced by solving a PDE.
- The definition of level set speed functions that implement blending, smoothing and embossing surface editing operators.
  - Blending is automatic and is constrained to only occur within a user-specified distance to an arbitrarily complex intersection curve.
  - Smoothing and embossing are constrained to occur within a user-specified region.
  - The user specifies the local geometric properties of the resulting surface modifications.
  - The user specifies if material should be added and/or removed during editing operations.
- The new techniques used to localize level set calculations.
- In Appendix B we present a new, numerically-stable curvature measure for level set surfaces.

2 Previous Work

Three areas of research are closely related to our level set surface editing work; volumetric sculpting, mesh-based surface editing/fairing and implicit modeling. Volumetric sculpting provides methods for directly manipulating the voxels of a volumetric model. CSG Boolean operations [Hoffmann 1989; Wang and Kaufman 1994] are commonly found in volume sculpting systems, providing a straightforward way to create complex solid objects by combining simpler primitives. One of the first volume sculpting systems is presented in [Galyean and Hughes 1991]. [Wang and Kaufman 1995] improved on this work by introducing tools for carving and sawing. More recently [Perry and Frisken 2001] implemented a volumetric sculpting system based on the Adaptive Distance Fields (ADF) [Frisken et al. 2000], allowing for models with adaptive resolution.

Performing CSG operations on mesh models is a long-standing area of research [Requicha and Voelcker 1985; Laidlaw et al. 1986]. Recently CSG operations were developed for multi-resolution subdivision surfaces by [Biermann et al. 2001], but this work did not address the problem of blending or smoothing the sharp features often produced by the operations. However, the smoothing of meshes has been studied on several occasions [Welch and Witkin 1994; Taubin 1995; Kobbelt et al. 1998]. [Desbrun et al. 1999] have developed a method for fairing irregular meshes using diffusion and curvature flow, demonstrating that mean-curvature based flow produces the best results for smoothing.

There exists a large body of surface editing work based on implicit models [Bloomenthal et al. 1997]. This approach uses implicit surface representations of analytic primitives or skeletal offsets. The implicit modeling work most closely related to ours is found in [Wyvill et al. 1999]. They describe techniques for performing blending, warping and boolean operations on skeletal implicit surfaces. [Desbrun and Gascuel 1995] address the converse problem of preventing unwanted blending between implicit primitives, as well as maintaining a constant volume during deformation.

Level set methods have been successfully applied in computer graphics, computer vision and visualization [Sethian 1999; Sapiro 2001], for example medical image segmentation [Malladi et al. 1995; Whitaker et al. 2001], shape morphing [Breen and Whitaker 2001], 3D reconstruction [Whitaker 1998; Zhao et al. 2001], and recently for the animation of liquids [Foster and Fedkiw 2001]

Our work stands apart from previous work in several ways. We have not developed volumetric modeling tools. Our editing operators act on surfaces that happen to have an underlying volumetric representation, but are based on the mathematics of deforming implicit surfaces. Our editing operators share several of the capabilities of mesh-based tools, but are not hampered by the difficulties of maintaining vertex/edge information. Since level set models are not tied to any specific implicit basis functions, they easily represent complex models to within the resolution of the sampling. Our work is the first to utilize level set methods to perform user-controlled editing of complex geometric models.

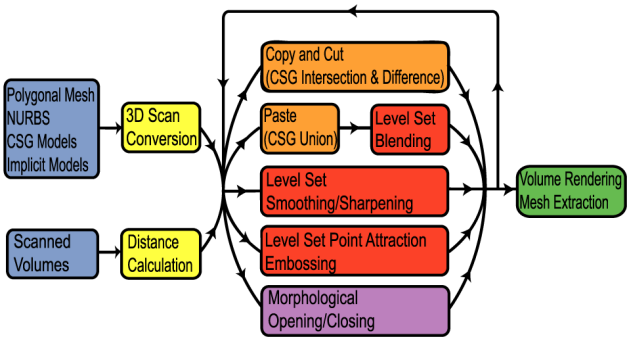


Figure 3: Our level set surface editing operators (red) fit into a larger editing framework. The pipeline consists of: input models (blue), pre-processing (yellow), CSG operations (orange), local LS operators (red), global LS operators (purple) and rendering (green).

3 Overview of the Editing Pipeline

The level set surface editing operators should be viewed as components of a larger modeling framework. The pipeline for this framework is presented in Figure 3. The red components contain the level set speed functions that we have developed for localized surface editing. The remaining components contain the data and operations needed for level set modeling, input models (blue), preprocessing (yellow), CSG operations (orange), global LS operators (purple) and rendering (green). The pipeline provides the context for the details of our speed functions.

3.1 Input and Output Models

We are able to import a wide variety of closed geometric models into the level set environment. We represent a level set model as an iso-surfaces embedded in a distance volume. Frequently we only store distance information in a narrow band of voxels surrounding the level set surface. As illustrated in Figure 3 we have developed and collected a suite of scan conversion methods for converting polygonal meshes, CSG models [Breen et al. 2000], implicit primitives, and NURBS surfaces into distance volumes. Additionally many types of scanning processes produce volumetric models directly, e.g. MRI, CT and laser range scan reconstruction. These models may be brought into our level set environment as is or with minimal pre-processing. We frequently utilize Sethian's Fast Marching Method [Sethian 1996] to convert volume datasets into distance volumes. The models utilized in this paper and their original form are listed in Table 1.

Table 1: Native representations of the input models and dimensions of the corresponding scan converted distance volumes.

Model	Representation	Dimensions
Dragon	volumetric reconstruction	$356 \times 161 \times 251$
Griffin	volumetric reconstruction	$312 \times 148 \times 294$
Greek bust	polygonal reconstruction	$221 \times 221 \times 161$
Human head	polygonal reconstruction	$256 \times 246 \times 193$
Utah teapot	NURBS surface	$156 \times 232 \times 124$
Supertoroid	implicit primitive	$91 \times 91 \times 31$

In the final stage of the pipeline we can either volume render the surface directly or render a polygonal mesh extracted from the volume. While there are numerous techniques available for both approaches, we found extracting and rendering Marching Cubes meshes [Lorensen and Cline 1987] to be satisfactory.

4 Level Set Surface Modeling

The Level Set Method, first presented in [Osher and Sethian 1988], is a mathematical tool for modeling surface deformations. A deformable (*i.e.* time-dependent) surface is implicitly represented as an iso-surface of a time-varying scalar function, $\phi(\boldsymbol{x}, t)$. A detailed description of level set models is presented in Appendix A.

4.1 LS Speed Function Building Blocks

Given the definition

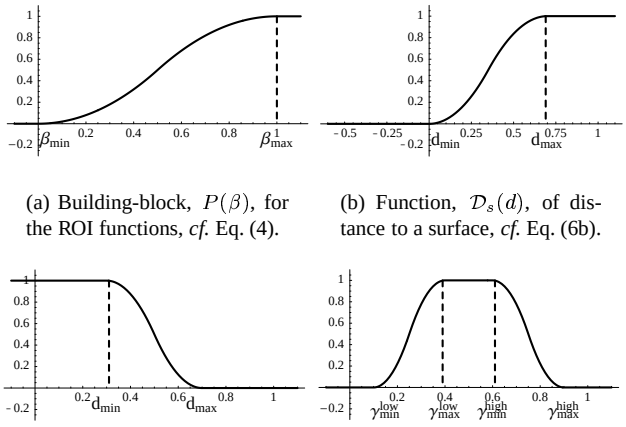
$$\mathcal{F}(\boldsymbol{x}, \boldsymbol{n}, \phi, \dots) \equiv \boldsymbol{n} \cdot \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}, \tag{1}$$

the fundamental level set equation, Eq. (12), can be rewritten as

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \, \mathcal{F}(\boldsymbol{x}, \boldsymbol{n}, \phi, \dots) \tag{2}$$

where $\mathrm{d}\boldsymbol{x}/\mathrm{d}t$ and $\boldsymbol{n} \equiv -\nabla\phi/|\nabla\phi|$ are the velocity and normal vectors at $\boldsymbol{x}$ on the surface. We assume a positive-inside/negative-outside sign convention for $\phi(\boldsymbol{x}, t)$, *i.e.* $\boldsymbol{n}$ *points outward*. Eq. (1) introduces the speed function $\mathcal{F}$, which is a user-defined scalar function that can depend on any number of variables including $\boldsymbol{x}$, $\boldsymbol{n}$, ϕ and its derivatives evaluated at $\boldsymbol{x}$, as well as a variety of external data inputs. $\mathcal{F}()$ is a *signed* scalar function that defines the motion (*i.e.* speed) of the level set surface in the direction of the local normal $\boldsymbol{n}$ at $\boldsymbol{x}$.

The speed function is usually based on a set of geometric measures of the implicit level set surface and data inputs. The challenge when working with level set methods is determining how to combine the building blocks to produce a local motion that creates a desired global or regional behavior of the surface. The general



(a) Building-block, $P(\beta)$, for the ROI functions, *cf.* Eq. (4).

(b) Function, $\mathcal{D}_s(d)$, of distance to a surface, *cf.* Eq. (6b).

(c) Function, $\mathcal{D}_p(d)$, of distance to point-set, *cf.* Eq. (6a).

(d) Function, $\mathcal{C}(\gamma)$, of geometric measure, γ, *cf.* Eq. (7).

Figure 4: Graph of region-of-influence (ROI) functions used to define the speed functions for our local level set operations, *cf.* Eq. (3).

structure for the speed functions used in our surface editing operators is

$$\mathcal{F}(\boldsymbol{x}, \boldsymbol{n}, \phi) = \mathcal{D}_q(d)\mathcal{C}(\gamma)\mathcal{G}(\gamma) \tag{3}$$

where $\mathcal{D}_q(d)$ is a distance-based cut-off function which depends on a distance measure d to a geometric structure q. $\mathcal{C}(\gamma)$ is a cut-off function that controls the contribution of $\mathcal{G}(\gamma)$ to the speed function. $\mathcal{G}(\gamma)$ is a function that depends on geometric measures γ derived from the level set surface, e.g. curvature. Thus, $\mathcal{D}_q(d)$ acts as a region-of-influence function that regionally constrains the LS calculation. $\mathcal{C}(\gamma)$ is a filter of the geometric measure and $\mathcal{G}(\gamma)$ provides the geometric contribution of the level set surface. In general γ is defined as zero, first, or second order measures of the LS surface.

4.2 Regionally Constraining LS Deformations

Most of our surface operators may be applied locally in a small user-defined region on the edited surface. In order to regionally restrict the deformation during the level set computation, a technique is needed for driving the value of $\mathcal{F}()$ to zero outside of the region. This is accomplished in three steps. The first step involves defining the region of influence (ROI), *i.e.* the region where $\mathcal{F}()$ should be non-zero. This is done by either the user interactively placing a 3D object around the region, or by automatically calculating a region from properties of the surface. Both cases involve defining a geometric structure that we refer to as a "region-of-influence (ROI) primitive". The nature of these primitives will vary for the different LS operations and will explicitly be defined in Section 5. The second step consists of calculating a distance measure to the ROI primitive. The final step involves defining a function that smoothly approaches zero at the boundary of the ROI.

We define a region-of-influence function $\mathcal{D}_q(d)$ in Eq. (3), where d is a distance measure from a point on the level set surface to the ROI primitive q. The functional behavior of $\mathcal{D}_q(d)$ clearly depends on the specific ROI primitive, q, but we found the following piecewise polynomial function to be useful as a common speed function building block:

$$P(\beta) = \begin{cases} 0 & \text{for } \beta \leq 0 \\ 2\beta^2 & \text{for } 0 < \beta \leq 0.5 \\ 1 - 2(\beta - 1)^2 & \text{for } 0.5 < \beta < 1 \\ 1 & \text{for } \beta \geq 1. \end{cases} \tag{4}$$

$P(\beta)$ and its derivatives are continuous and relatively inexpensive to compute. See Figure 4(a). Other continuous equations with the same basic shape would also be satisfactory. We then define

$$\mathcal{P}(d; d_{min}, d_{max}) \equiv P\left(\frac{d - d_{min}}{d_{max} - d_{min}}\right) \tag{5}$$

where d_{min} and d_{max} are user-defined parameters that define the limits and sharpness of the cut-off. Let us finally define the following region-of-influence functions

$$\mathcal{D}_p(d) = 1 - \mathcal{P}(d; d_{min}, d_{max}) \tag{6a}$$
$$\mathcal{D}_s(d) = \mathcal{P}(d; 0, d_{max}) \tag{6b}$$

for a point-set, p, and a closed surface, s.

In Eq. (6a) d denotes the distance from a point on the level set surface to the closet point in the point-set p. In Eq. (6b) d denotes a signed distance measure from a point on the level set surface to the implicit surface s. The signed distance measure does not necessarily have to be Euclidean distance - just a monotonic distance measure following the positive-inside/negative-outside convention. Note that $\mathcal{D}_p(d)$ is one when the shortest distance, d, to the point-set is smaller than d_{min}, and decays smoothly to zero as d increases to d_{max}, after which it is zero. $\mathcal{D}_s(d)$, on the other hand, is zero everywhere outside, as well as on, the surface s ($d \leq 0$), but one inside when the distance measure d is larger than d_{max}.

An additional benefit of the region-of-influence functions is that they define the portion of the volume where the surface cannot move. We use this information to determine what voxels should be updated during the level set deformation, significantly lowering the amount of computation needed when performing editing operations. This technique allows our operators to be rapidly computed when modifying large models.

4.3 Limiting Geometric Property Values

We calculate a number of geometric properties from the level set surface. The zero order geometric property that we utilize is shortest distance from the level set surface to some ROI primitive. The first order property is the surface normal, $\boldsymbol{n} \equiv -\nabla\phi/|\nabla\phi|$. Second order information includes a variety of curvature measures of the LS surface. In Appendix B we outline a new numerical approach to deriving the mean, Gaussian and principle curvatures of a level set surface. Our scheme has numerical advantages relative to traditional central finite difference schemes for computing the second order derivatives. We found the mean curvature to be a useful second order measure [Evans and Spruck 1991].

Another desirable feature of our operators is that they allow the user to control the geometric properties of surface in the region being edited. This feature is implemented with another cut-off function, $\mathcal{C}()$, within the level set speed function. $\mathcal{C}()$ allows the user to slow and then stop the level set deformation as a particular surface property approaches a user-specified value. We reuse the cut-off function, Eq. (5), defined in the previous section, as a building block for $\mathcal{C}()$. We define

$$\mathcal{C}(\gamma) = \begin{cases} \mathcal{P}(\gamma; \gamma_{min}^{low}, \gamma_{max}^{low}) & \text{for } \gamma \leq \overline{\gamma} \\ 1 - \mathcal{P}(\gamma; \gamma_{min}^{high}, \gamma_{max}^{high}) & \text{for } \gamma > \overline{\gamma} \end{cases} \tag{7}$$

where $\overline{\gamma} \equiv (\gamma_{max}^{low} + \gamma_{min}^{high})/2$. The four parameters γ_{min}^{low}, γ_{max}^{low}, γ_{min}^{high}, and γ_{max}^{high} define respectively the upper and lower support of the filter, see Figure 4(d).

4.4 Constraining the Direction of LS Motions

Another important feature of the level set framework is its ability to control the direction of the level set deformation. We are able

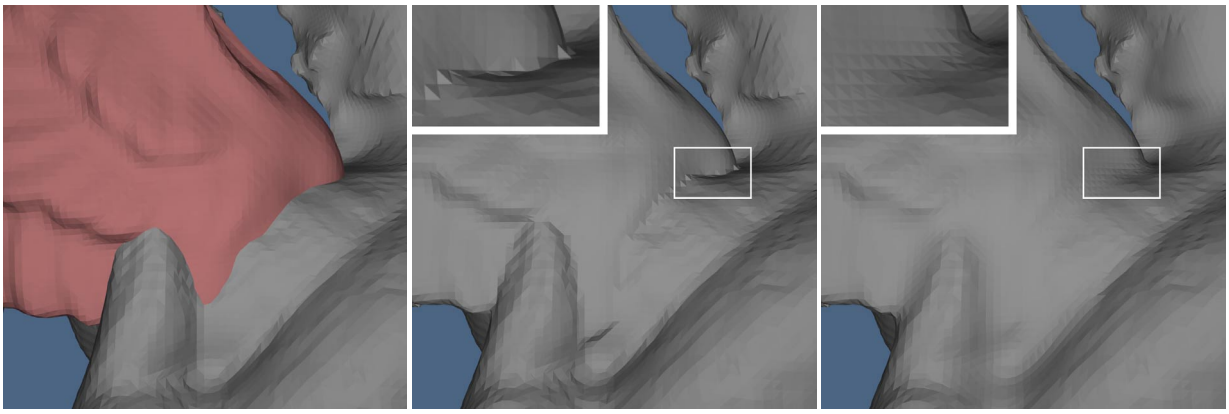


Figure 5: Left: Positioning the (red) wing model on the dragon model. Middle: The models are pasted together (CSG union operation), producing sharp, undesirable creases, a portion of which is expanded in the box. Right: Same region after automatic blending based on mean curvature. The blending is constrained to only move outwards. The models are rendered with flat-shading to highlight the details of the surface structure.

to restrict the motion of the surface to only add or remove material during the level set editing operations. At any point the level set surface can only move in the direction of the local surface normal. Hence, we can simply redefine the speed function as $\min(\mathcal{G}, 0)$ to remove material (inward motion only) and $\max(\mathcal{G}, 0)$ to add material (outward motion only). In the case of curvature driven speed functions this produces min/max flows [Sethian 1999]. Of course no restriction on the direction of the motion need be imposed.

5 Definition of Surface Editing Operators

Given the building blocks described in the previous section, the level set surface editing operators outlined in Figure 3 may be defined. We begin by defining the well-known CSG operations that are essential to most editing systems. We then define the new level set speed functions that implement our surface editing operators by combining the geometric measures with the region-of-influence and cut-off functions.

5.1 CSG Operations

Since level set models are volumetric, the constructive solid geometry (CSG) [Hoffmann 1989] operations of union, difference and intersection may be applied to them. This provides a straightforward approach to copy, cut and paste operations on the level set surfaces. In our level set framework with a positive-inside/negative-outside sign convention for the distance volumes these are implemented as min/max operations [Wang and Kaufman 1994] on the voxel values as summarized in Table 2. Any two closed surfaces represented as signed distance volumes can be used as either the main edited model or the cut/copy primitive. In our editing system the user is able to arbitrarily scale, translate and rotate the models before a CSG operation is performed.

Table 2: Implementation of CSG operations on two level set models, A and B, represented by distance volumes V_A and V_B with positive inside and negative outside values.

Action	CSG Operation	Implementation
Copy	Intersection, $A \cap B$	$\text{Min}(V_A, V_B)$
Paste	Union, $A \cup B$	$\text{Max}(V_A, V_B)$
Cut	Difference, $A - B$	$\text{Min}(V_A, -V_B)$

5.2 Automatic Localized LS Blending

The surface models produced by the CSG paste operation typically produces sharp and sometimes jagged creases at the intersection of the two surfaces. We can dramatically improve this region of the surface by applying an automatic localized blending. The method is automatic because it only requires the user to specify a few parameter values. It is localized because the blending operator is only applied near the surface intersection region. One possible solution to localizing the blending is to perform the deformation in regions near both of the input surfaces. However, this naive approach would result in blending the two surfaces in *all* regions of space where the surfaces come within a user-specified distance of each other, creating unwanted blends. A better solution, and the one we use, involves defining the region of influence based on the distance to the *intersection curve* shared by both input surfaces. A sampled representation of this curve is the set of voxels that contains a zero distance value (within some sub-voxel value ϵ) to both surfaces. We have found this approximate representation of the intersection curve as a point-set to be sufficient for defining a shortest distance d for the region-of-influence function, $\mathcal{D}_p(d)$, *cf.* Eq. (3). Representing the intersection curve by a point-set allows the curve to take an arbitrary form - it can even be composed of multiple curve segments without introducing any complication to the computational scheme.

The blending operator moves the surface in a direction that minimizes a curvature measure, $\mathcal{K}$, on the level set surface. This is obtained by making the speed function, $\mathcal{G}$, Eq. (3), proportional to $\mathcal{K}$, leading to the following blending speed function:

$$\mathcal{F}_{blend}(\boldsymbol{x}, \boldsymbol{n}, \phi) = \alpha\, \mathcal{D}_p(d)\, \mathcal{C}(\mathcal{K})\, \mathcal{K} \tag{8}$$

where α is a user-defined positive scalar that is related to the rate of convergence of the LS calculation, $\mathcal{D}_p(d)$ is defined in Eq. (6a) where d is the shortest distance from the level set surface to the intersection curve point set, and $\mathcal{C}(\mathcal{K})$ is given by Eq. (7) where $\mathcal{K}$ is one of the curvatures define in Appendix B. Through the functions $\mathcal{D}_p$ and $\mathcal{C}$ the user has full control over the region of influence of the blending (d_{min} and d_{max}) and the upper and lower curvature values of the blend (γ_{min}^{low}, γ_{max}^{low} and γ_{min}^{high}, γ_{max}^{high}). Furthermore we can control if the blend adds or removes material, or both as described in Section 4.4.

Automatic blending is demonstrated in Figure 5. A wing model is positioned relative to a dragon model. The two models are pasted together and automatic mean curvature-based blending is applied to smooth the creased intersection region.

5.3 Localized LS Smoothing/Sharpening

The smoothing operator smooths the level set surface in a user-specified region. This is accomplished by enclosing the region of interest by a geometric primitive. The "region-of-influence prim-

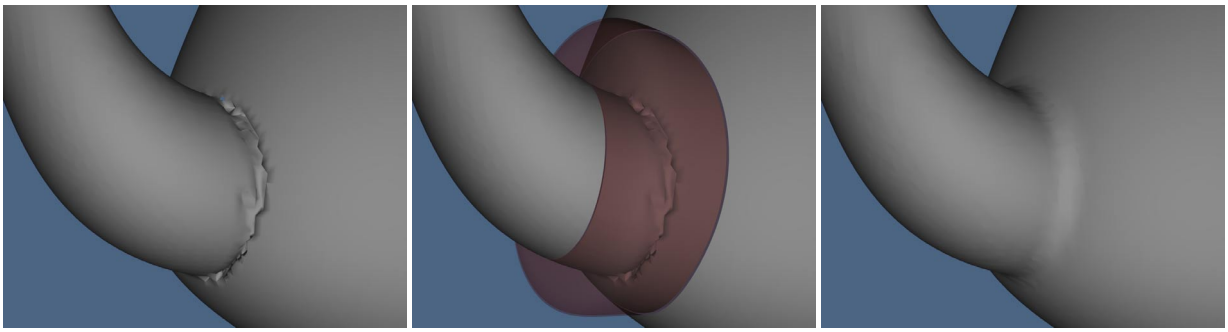


Figure 6: (left) Scan conversion errors near the teapot spout. (middle) Placing a (red) superellipsoid around the errors. (right) The errors are smoothed away in 15 seconds. The surface is constrained to only move outwards.

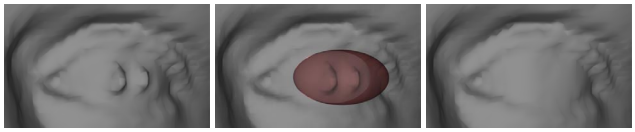


Figure 7: Regionally constrained smoothing. Left: Laser scan reconstruction with unwanted, pointed artifacts in the eye. Middle: Defining the region to be smoothed with a (red) superellipsoid. Right: Smoothing the surface within the superellipsoid. The surface is constrained to only move inwards.

itive" can be any closed surface for which we have signed inside/outside information, e.g. a level set surface or an implicit primitive. We use superellipsoids [Barr 1981] as a convenient ROI primitive, a flexible implicit primitive defined by two shape parameters. The surface is locally smoothed by applying motions in a direction that reduces the local curvature. This is accomplished by moving the level set surface in the direction of the local normal with a speed that is proportional to the curvature. Therefore the speed function for the smoothing operator is

$$\mathcal{F}_{smooth}(\boldsymbol{x}, \boldsymbol{n}, \phi) = \alpha \mathcal{D}_s(d)\mathcal{C}(\mathcal{K})\mathcal{K}. \tag{9}$$

Here d denotes the signed value of the monotonic inside/outside function of the ROI primitive s evaluated at $\boldsymbol{x}$. As before, $\mathcal{C}_s(d)$ ensures that the speed function smoothly goes to zero as $\boldsymbol{x}$ approaches the boundary of the ROI primitive.

Figure 7 demonstrates our smoothing operator applied to a laser scan reconstruction. Unwanted artifacts are removed from an eye by first placing a red superellipsoid around the region of interest. A smoothing operator constrained to only remove material is applied and the spiky artifacts are removed. Figure 6 demonstrates our smoothing operator applied to a preliminary 3D scan conversion of the Utah teapot. Unwanted artifacts are removed from the region where the spout meets the body of the teapot by first placing a superellipsoid around the region of interest. A smoothing operator constrained to only add material is applied and the unwanted artifacts are removed. In our final, artificial smoothing example in Figure 9 a complex structure is completely smoothed away. This examples illustrates that changes of topological genus and number of disconnected components are easily handled within a level set framework during smoothing.

We obtain a sharpening operator by simply inverting the sign of α in Eq. (9) and applying an upper cut-off to the curvature in $\mathcal{C}()$ in order to maintain numerical stability. The sharpening operator has been applied to the hair of the Greek bust in Figure 1.

5.4 Point-Set Attraction and Embossing

We have developed an operator that attracts and repels the surface towards and away from a point set. These point sets can be samples of lines, curves, planes, patches and other geometric shapes, *e.g.* text. By placing the point sets near the surface, we are able to emboss the surface with the shape of the point set. Similar to the smoothing operator, the user encloses the region to be embossed with a ROI primitive *e.g.* a superellipsoid. The region-of-interest function for this operator is $\mathcal{D}_s(d)$, Eq. (6b).

Figure 8: Left: Three types of single point attractions/repulsions using different ROI primitives and γ values. Right: Utah teapot embossed with 7862 points sampling the "SIGGRAPH 2002" logo.

First, assume that all of the attraction points are located outside the LS surface. $\boldsymbol{p}_i$ denotes the closest attraction point to $\boldsymbol{x}$, a point on the LS surface. Our operator only allows the LS surface to move towards $\boldsymbol{p}_i$ if the unit vector, $\boldsymbol{u}_i \equiv (\boldsymbol{p}_i - \boldsymbol{x})/|\boldsymbol{p}_i - \boldsymbol{x}|$, is pointing in the same direction as the local surface normal $\boldsymbol{n}$. Hence, the speed function should only be non-zero when $0 < \boldsymbol{n} \cdot \boldsymbol{u}_i \leq 1$. Since the sign of $\boldsymbol{n} \cdot \boldsymbol{u}_i$ is reversed if $\boldsymbol{p}_i$ is instead located inside the LS surface we simply require $\gamma = -\text{sign}[\phi(\boldsymbol{p}_i, t)]\boldsymbol{n} \cdot \boldsymbol{u}_i$ to be positive for any closest attraction point $\boldsymbol{p}_i$. This amounts to having only positive cut-off values for $\mathcal{C}(\gamma)$. Finally we let $\mathcal{G} = -\alpha\phi(\boldsymbol{p}_i, t)$ since this will guarantee that the LS surface, $\boldsymbol{x}$, actually stops once it reaches $\boldsymbol{p}_i$. The following speed function implements the point-set attraction operator:

$$\mathcal{F}_{point}(\boldsymbol{x}, \boldsymbol{n}, \phi) = -\alpha \mathcal{D}_s(d)\mathcal{C}(-\text{sign}[\phi(\boldsymbol{p}_i, t)]\boldsymbol{n} \cdot \boldsymbol{u}_i)\phi(\boldsymbol{p}_i, t), \tag{10}$$

where d is a signed distance measure to a ROI primitive evaluated at $\boldsymbol{x}$ on the LS surface, and p_i is the closest point in the set to $\boldsymbol{x}$. The shape of the primitive and the values of the four positive parameters in Eq. (7) define the footprint and sharpness of the embossing. See Figure 8, left. Point-repulsion is obtained by making α negative. Note that Eq. (10) is just one example of many possible point-set attraction speed functions.

In Figure 8, right, the Utah teapot is embossed with 7862 points that have been acquired by scanning an image of the SIGGRAPH 2002 logo and warping the points to fit the shape of the teapot.

5.5 Global Morphological Operators

The new level set operators presented above were designed to perform localized deformations of a level set surface. However, if the user wishes to perform a global smoothing of a level set surface, it is advantageous to use an operator other than $\mathcal{F}_{smooth}$. For

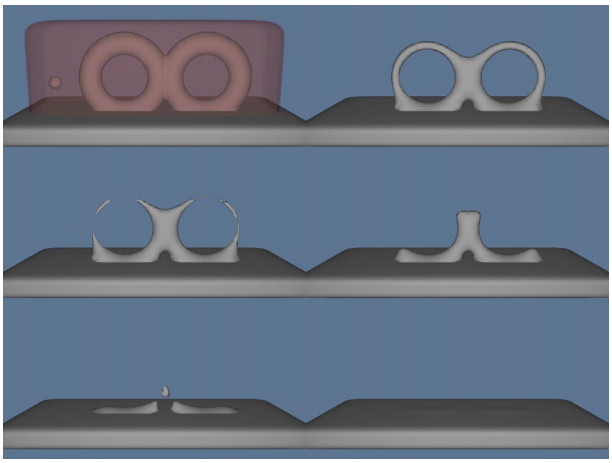


Figure 9: Changes in topological genus and the number of disconnected components are easily handled within a level set framework during smoothing. The superellipsoid defines the portion of the surface to be smoothed. The surface is constrained to move only inwards.

a global smoothing the level set propagation is computed on the whole volume, which can be slow for large volumes. However, in this case morphological opening and closing operators [Serra 1982] offer faster alternatives to global smoothing of level set surfaces. While we are not the first to explore morphological operators within a level set framework [Sapiro et al. 1993; Maragos 1996], we have implemented them and find them useful. Morphological openings and closings consist of two fundamental operators, dilations D_ω and erosions E_ω. Dilation creates an offset surface a distance ω outward from the original surface, and erosion creates an offset surface a distance ω inwards from the original surface. The morphological opening operator O_ω is an erosion followed by a dilation, i.e. $\mathrm{O}_\omega = \mathrm{D}_\omega \circ \mathrm{E}_\omega$, which removes small pieces or thin appendages. A closing is defined as $\mathrm{C}_\omega \phi = \mathrm{E}_\omega \circ \mathrm{D}_\omega \phi$, and closes small gaps or holes within objects. Morphological operators may be implemented by solving a special form of the level set equation, the Eikonal equation, $\partial\phi/\partial t = \pm|\nabla\phi|$, up to a certain time t, utilizing Sethian's Fast Marching Method [Sethian 1996]. The value of t controls the offset distance from the original surface of $\phi(t = 0)$. Figure 10 contains a model from a laser scan reconstruction that has been smoothed with an opening operator with ω equal to 3.

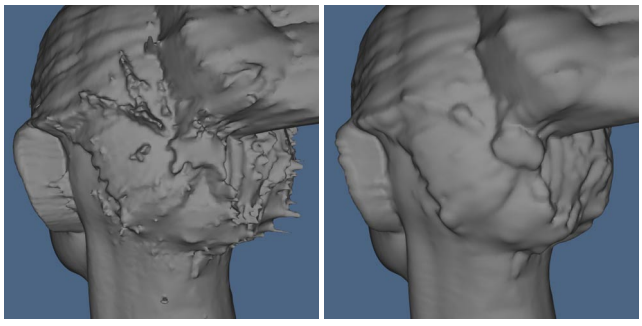


Figure 10: Applying a morphological opening to a laser scan reconstruction of a human head. The opening performs global smoothing by removing protruding structures smaller than a user-defined value.

5.6 Editing Session Details

Figure 11 contains a series of screen shots taken of our level set modeling program while constructing the two-headed winged dragon. The first shows the original dragon model loaded into the system. A cylindrical primitive is placed around its head and it is cut off. The model of the head is duplicated and the two heads are positioned relative to each other. Once the user is satisfied with their orientation, they are pasted together and an automatic blending is performed at the intersection seam. The combined double head model is positioned over the cropped neck of the dragon body. The double head is pasted and blended onto the body. The griffin model is loaded into the LS modeling system. A primitive is place around one of its wings. The portion of the model within the primitive is copied, being stored in a buffer. Several cutting operations are used to trim the wing model (not shown). The double-headed dragon model is loaded, and the wing is positioned, pasted and blended onto it. A mirror copy of the wing model is created. It is also positioned, pasted and blended onto the other side of the double-headed dragon. We then added a loop onto the dragon's back as if designing a bracelet charm. This is accomplished by positioning, pasting, and blending a scan-converted supertoroid, producing the final model seen in the bottom right.

The Greek bust model was repaired by copying the nose from the human head model of Figure 10, and pasting and blending the copied model onto the broken nose. A piece from the right side of the bust was copied, mirrored, pasted and blended onto the left side of her face. Local smoothing operators were applied to various portions of her cheeks to clean minor cracks. Finally, the sharpening operator was applied within a user-defined region around her hair.

Table 3: Typical operator execution times on a R10K 250MHz MIPS processor.

Operation	Objects	sub-volume	Time
Paste	wing on dragon	$316 \times 172 \times 215$	33 sec.
Blend	wing on dragon	$82 \times 48 \times 63$	98 sec.
Smooth	teapot spout	$60 \times 55 \times 31$	15 sec.
Opening	human head	$256 \times 246 \times 193$	22 sec.
Emboss	single point	$21 \times 29 \times 29$	1.5 sec.

Table 4: Parameters used in examples. γ_{min}^{high} and γ_{min}^{high} are only used during sharpening. Their values are 0.8 and 0.9. No upper limit is placed on γ in the other examples.

Example	d_{min}	d_{max}	γ_{min}^{low}	γ_{max}^{low}
Wing Blending	7	9	0.04	0.06
Eye Smoothing	0.9	1	0.04	0.07
Spout Smoothing	0.9	1	0.1	0.13
Hair Sharpening	0.9	1	0.01	0.013
Teapot Embossing	0.9	1	0.8	0.9

6 Conclusion and Future Work

We have presented an approach to implementing surface editing operators within a level set framework. By developing a new set of level set speed functions automatic blending, localized smoothing and embossing may be performed on level set models. Additionally we have implemented morphological and volumetric CSG operators to fill out our modeling environment. All of the information needed to deform a level set surface is encapsulated in the speed function, providing a simple, unified computational framework. The level set framework offers several advantages. By construction, self-intersection cannot occur, which guarantees the generation of physically-realizable, simple, closed surfaces. Additionally, level set models easily change topological genus, and are free of the edge connectivity and mesh quality problems associated with mesh models.

Several issues still must be addressed to improve our work. Currently level set implementations are based on uniform samplings of

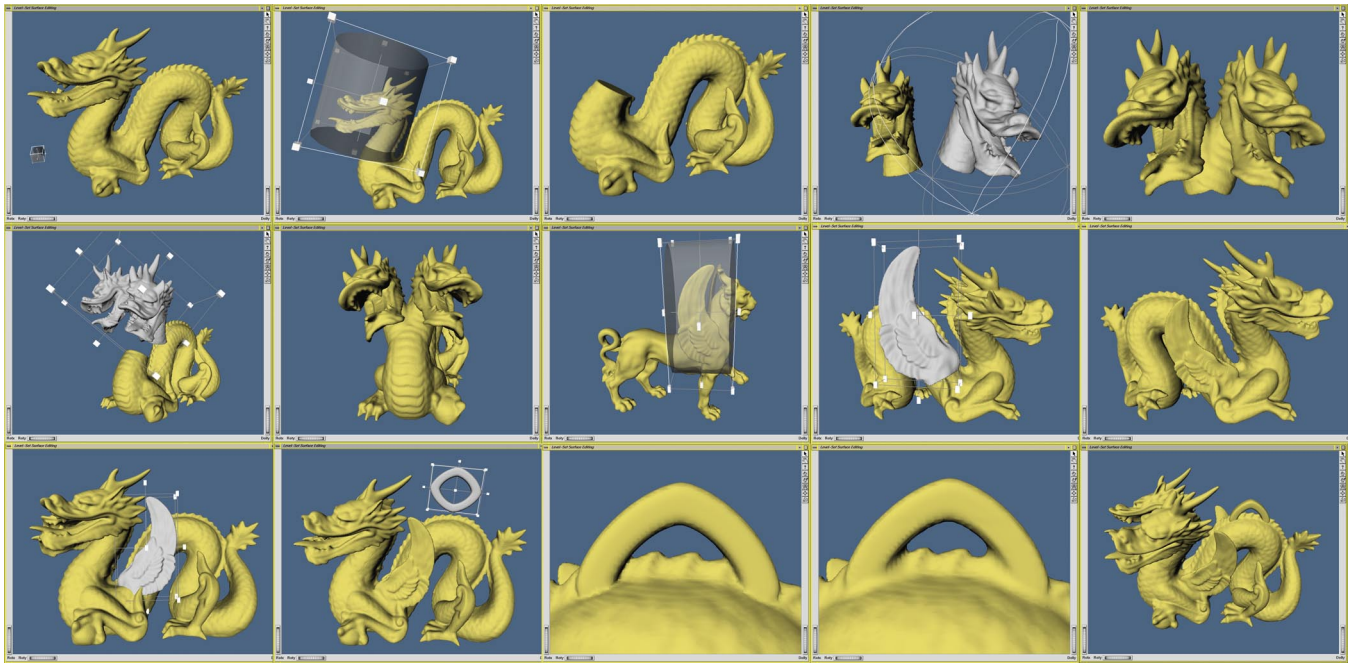


Figure 11: Series of operations used to create the winged two-headed dragon of Figure 1. First the head is cut off, pasted and blended back onto the body. Next a wing is copied from a different model and blended onto one side of the dragon. The same wing is then mirrored and blended onto the other side. Finally a scan converted supertoroid is blended onto the dragon's back to form the loop of a bracelet charm.

space, a fact that effectively limits the resolution of the objects that can be modeled. The development of adaptive level set methods would allow our operators to be applied to adaptive distance fields. It is possible to shorten the time needed to edit level set surfaces. Incrementally updating the mesh used to view the edited surface, utilizing direct volume rendering hardware, parallelizing the level set computations, and exploring multiresolution volumetric representations will lead to editing operations that require only a fraction of a second, instead of tens of seconds.

We have presented five example level set surface editing operators. Given the generality and flexibility of our framework many more can be developed. We intend to explore operators that utilize Gaussian and principal curvature, extend embossing to work directly with lines, curves and solid objects, and ones that may be utilized for general surface manipulations, such as dragging, warping, and sweeping.

7 Acknowledgements

We would like to thank Mathieu Desbrun for his helpful suggestions, and Cici Koenig and Katrine Museth for helping us with the figures. The Greek bust and human head models were provided by Cyberware Inc. The dragon and griffin models were provided the Stanford Computer Graphics Laboratory. The teapot model was provided by the University of Utah's Geometric Design and Computation Group. This work was financially supported by National Science Foundation grants ASC-89-20219, ACI-9982273 and ACI-0083287.

References

ADALSTEINSSON, D., AND SETHIAN, J. 1995. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 269–277.

AMENTA, N., BERN, M., AND KAMVYSSELIS, M. 1998. A new voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH '98*, 415–421.

BAJAJ, C., BERNARDINI, F., AND XU, G. 1995. Automatic reconstruction of surfaces and scalar fields from 3D scans. In *Proc. SIGGRAPH '95*, 109–118.

BARR, A. 1981. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications 1*, 1, 11–23.

BIERMANN, H., KRISTJANSSON, D., AND ZORIN, D. 2001. Approximate Boolean operations on free-form solids. In *Proc. SIGGRAPH 2001*, 185–194.

BLOOMENTHAL, J., ET AL., Eds. 1997. *Introduction to Implicit Surfaces*. Morgan Kaufmann, San Francisco.

BOUGUET, J.-Y., AND PERONA, P. 1999. 3D photography using shadow in dual space geometry. *International Journal of Computer Vision 35*, 2 (Nov/Dev), 129–149.

BREEN, D., AND WHITAKER, R. 2001. A level set approach for the metamorphosis of solid models. *IEEE Trans. on Visualization and Computer Graphics 7*, 2, 173–192.

BREEN, D., MAUCH, S., AND WHITAKER, R. 2000. 3D scan conversion of CSG models into distance, closest-point and color volumes. In *Volume Graphics*, M. Chen, A. Kaufman, and R. Yagel, Eds. Springer, London, 135–158.

COHEN, E., RIESENFELD, R., AND ELBER, G. 2001. *Geometric Modeling with Splines*. AK Peters, Natick, MA.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH '96*, 303–312.

DESBRUN, M., AND GASCUEL, M. 1995. Animating soft substances with implicit surfaces. In *Proc. SIGGRAPH 95 Conference*, 287–290.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. SIGGRAPH '99*, 317–324.

DO CARMO, M. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ.

EDELSBRUNNER, H., AND MÜCKE, E. 1994. Three-dimensional alpha shapes. *ACM Trans. on Graphics 13*, 1, 43–72.

EVANS, L., AND SPRUCK, J. 1991. Motion of level sets by mean curvature, I. *Journal of Differential Geometry*, 635–681.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proc. SIGGRAPH 2001*, 23–30.

FRISKEN, S., PERRY, R., ROCKWOOD, A., AND JONES, T. 2000. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *SIGGRAPH 2000 Proceedings*, 249–254.

GALYEAN, T., AND HUGHES, J. 1991. Sculpting: An interactive volumetric modeling technique. In *Proc. SIGGRAPH '91*, 267–274.

HOFFMANN, C. 1989. *Geometric and Solid Modeling*. Morgan Kaufmann, San Francisco.

KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. 1998. Interactive multi-resolution modeling on arbitrary meshes. In *Proc. SIGGRAPH '98*, 105–114.

KOBBELT, L. P., BOTSCH, M., SCHWANECKE, U., AND SEIDEL, H.-P. 2001. Feature sensitive surface extraction from volume data. In *Proc. SIGGRAPH 2001*, 57–66.

LAIDLAW, D., TRUMBORE, W., AND HUGHES, J. 1986. Constructive solid geometry for polyhedral objects. In *Proc. SIGGRAPH '86*, vol. 20, 161–170.

LORENSEN, W., AND CLINE, H. 1987. Marching Cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH '87*, 163–169.

MALLADI, R., SETHIAN, J., AND VEMURI, B. 1995. Shape modeling with front propagation: A level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence 17*, 2, 158–175.

MARAGOS, P. 1996. Differential morphology and image processing. *IEEE Trans. on Image Processing 5*, 6 (June), 922–937.

OSHER, S., AND FEDKIW, R. 2001. Level set methods: An overview and some recent results. *Journal of Computational Physics 169*, 475–502.

OSHER, S., AND SETHIAN, J. 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics 79*, 12–49.

PENG, D., MERRIMAN, B., OSHER, S., ZHAO, H.-K., AND KANG, M. 1999. A PDE-based fast local level set method. *Journal of Computational Physics 155*, 410–438.

PERRY, R., AND FRISKEN, S. 2001. Kizamu: A system for sculpting digital characters. In *Proc. SIGGRAPH 2001*, 47–56.

REQUICHA, A., AND VOELCKER, H. 1985. Boolean operations in solid modeling: Boundary evaluation and merging algorithms. *Proceedings of the IEEE 73*, 1, 30–44.

RUDIN, L., OSHER, S., AND FATEMI, C. 1992. Nonlinear total variation based noise removal algorithms. *Physica D 60*, 259–268.

SAPIRO, G., KIMMEL, R., SHAKED, D., KIMIA, B., AND BRUCKSTEIN, A. 1993. Implementing continuous-scale morphology via curve evolution. *Pattern Recognition*, 9, 1363–1372.

SAPIRO, G. 2001. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, Cambridge, UK.

SERRA, J. 1982. *Image Analysis and Mathematical Morphology*. Academic Press, London.

SETHIAN, J. 1996. A fast marching level set method for monotonically advancing fronts. In *Proceedings of the National Academy of Science*, vol. 93 of 4, 1591–1595.

SETHIAN, J. 1999. *Level Set Methods and Fast Marching Methods*, second ed. Cambridge University Press, Cambridge, UK.

TAUBIN, G. 1995. A signal processing approach to fair surface design. In *Proc. SIGGRAPH '95*, 351–358.

TSITSIKLIS, J. 1995. Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automatic Control 40*, 9, 1528–1538.

WANG, S., AND KAUFMAN, A. 1994. Volume-sampled 3D modeling. *IEEE Computer Graphics and Applications 14*, 5 (September), 26–32.

WANG, S., AND KAUFMAN, A. 1995. Volume sculpting. In *1995 Symposium on Interactive 3D Graphics*, ACM SIGGRAPH, 151–156.

WELCH, W., AND WITKIN, A. 1994. Free-form shape design using triangulated surfaces. In *Proc. SIGGRAPH '94*, 247–256.

WHITAKER, R., AND XUE, X. 2001. Variable-conductance, level-set curvature for image denoising. In *Proc. IEEE International Conference on Image Processing*, 142–145.

WHITAKER, R., BREEN, D., MUSETH, K., AND SONI, N. 2001. Segmentation of biological datasets using a level-set framework. In *Volume Graphics 2001*, M. Chen and A. Kaufman, Eds. Springer, 249–263.

WHITAKER, R. 1998. A level-set approach to 3D reconstruction from range data. *Int. Jrnl. of Comp. Vision October*, 3, 203–231.

WYVILL, B., GALIN, E., AND GUY, A. 1999. Extending the CSG tree. warping, blending and Boolean operations in an implicit surface modeling system. *Computer Graphics Forum 18*, 2 (June), 149–158.

ZHAO, H.-K., OSHER, S., AND FEDKIW, R. 2001. Fast surface reconstruction using the level set method. In *Proc. 1st IEEE Workshop on Variational and Level Set Methods*, 194–202.

A Level Set Models

A deformable (*i.e.* time-dependent) surface, $\mathcal{S}(t)$, is implicitly represented as an iso-surface of a time-varying[1] scalar function, $\phi(\boldsymbol{x}, t)$, embedded in 3D, *i.e.*

$$\mathcal{S}(t) = \{\boldsymbol{x}(t) \mid \phi(\boldsymbol{x}(t), t) = k\}, \tag{11}$$

[1]Our work uses the dynamic level set equation, which is more flexible than the corresponding stationary equation, $\phi(\boldsymbol{x}) = k(t)$, see [Sethian 1999] for more details.

where $k \in \Re$ is the iso-value, $t \in \Re^+$ is time, and $\boldsymbol{x}(t) \in \Re^3$ is a point in space on the iso-surface. It might seem inefficient to implicitly represent a surface with a 3D scalar function; however the higher dimensionality of the representation provides one of the major advantages of the LS method: the flexible handling of changes in the topology of the deformable surface. This implies that LS surfaces can easily represent complicated surface shapes that can, form holes, split to form multiple objects, or merge with other objects to form a single structure.

The fundamental level set equation of motion for $\phi(\boldsymbol{x}(t), t)$ is derived by differentiating both sides of Eq. (11) with respect to time t, and applying the chain rule giving:

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}, \tag{12}$$

where $\mathrm{d}\boldsymbol{x}/\mathrm{d}t$ denotes the speed vectors of the level set surface. A number of numerical techniques by [Osher and Sethian 1988; Adalsteinsson and Sethian 1995] make the initial value problem of Eq. (12) computationally feasible. A complete discussion of the details of the level set method is beyond the scope of this paper. We instead refer the interested reader to [Sethian 1999; Osher and Fedkiw 2001]. However, we will briefly mention two of the most important techniques: the first is the so called "up-wind scheme" which addresses the problem of overshooting when trying to solve Eq. (12) by a simple finite forward difference scheme. The second is related to the fact that one is typically only interested in a single solution to Eq. (12), say the $k = 0$ level set. This implies that the evaluation of ϕ is important only in the vicinity of that level set. This forms the basis for "narrow-band" schemes [Adalsteinsson and Sethian 1995; Whitaker 1998; Peng et al. 1999] that solve Eq. (12) in a narrow band of voxels containing the surface. The "up-wind scheme" makes the level set method numerically robust, and the "narrow-band scheme" makes its computational complexity proportional to the level set's surface area rather than the size of the volume in which it is embedded.

B Curvature of Level Set Surfaces

The principle curvatures and principle directions are the eigenvalues and eigenvectors of the *shape matrix* [do Carmo 1976]. For an implicit surface, the shape matrix is the derivative of the normalized gradient (surface normals) projected onto the tangent plane of the surface. If we let the normals be $\boldsymbol{n} = \nabla\phi/|\nabla\phi|$, the derivative of this is the 3×3 matrix

$$\boldsymbol{N} = \left(\frac{\partial \boldsymbol{n}}{\partial x} \;\; \frac{\partial \boldsymbol{n}}{\partial y} \;\; \frac{\partial \boldsymbol{n}}{\partial z} \right)^T. \tag{13}$$

The projection of this derivative matrix onto the tangent plane gives the shape matrix [do Carmo 1976] $\boldsymbol{B} = \boldsymbol{N}(I - \boldsymbol{n} \otimes \boldsymbol{n})$, where $\otimes$ is the exterior product. The eigenvalues of the matrix $\boldsymbol{B}$ are k_1, k_2 and zero, and the eigenvectors are the principle directions and the normal, respectively. Because the third eigenvalue is zero, we can compute k_1, k_2 and various differential invariants directly from the invariants of $\boldsymbol{B}$. Thus the weighted curvature flow is computing from $\boldsymbol{B}$ using the identities $D = ||\boldsymbol{B}|\!|$, $H = \mathrm{Tr}(\boldsymbol{B})/2$, and $K = 2H^2 - D^2/2$. The choice of numerical methods for computing $\boldsymbol{B}$ is discussed in the following section. The principle curvature are calculated by solving the quadratic

$$k_{1,2} = H \pm \sqrt{\frac{D^2}{2} - H^2}. \tag{14}$$

In many circumstances, the curvature term, which is a kind of directional diffusion, which does not suffer from overshooting, can be computed directly from first- and second-order derivatives of ϕ using central difference schemes. However, we have found that central differences do introduce instabilities when computing flows that rely on quantities other than the mean curvature. Therefore we use the method of *differences of normals* [Rudin et al. 1992; Whitaker and Xue 2001] in lieu of central differences. The strategy is to compute normalized gradients at staggered grid points and take the difference of these staggered normals to get centrally located approximations to $\boldsymbol{N}$. The shape matrix $\boldsymbol{B}$ is computed with gradient estimates from central differences. The resulting curvatures are treated as speed functions (motion in the normal direction), and the associated gradient magnitude is computed using the up-wind scheme.