Constructing Implicit Shape Models from Boundary Data

Luiz Velho[†‡]

Demetri Terzopoulos[‡]

Jonas Gomes[†]

[‡] University of Toronto – Department of Computer Science

Toronto, Ontario, M5S-1A4

[†] IMPA – Instituto de Matemática Pura e Aplicada

Estrada Dona Castorina, 110, Rio de Janeiro, Brazil, 22460-320

Contact:	Luiz Velho
Address:	IMPA – Instituto de Matemática Pura e Aplicada
	Estrada Dona Castorina, 110
	Rio de Janeiro, Brazil, 22460
Phone:	(5521) 294-9032, ext. 105
Fax:	(5521) 512-4115
E-mail:	lvelho@visgraf.impa.br

Abstract

We develop a method to construct an implicit shape model that approximates a solid object from its characteristic function. The method is based on multiresolution edge detection and reconstruction using dyadic wavelets. It synthesizes edges at multiple resolutions such that the reconstructed function preserves the shape boundary and the gradient of the function gives the normal vector field at the boundary. Several examples and applications of the procedure are given. One of the applications consists in the use of the procedure to obtain a rough estimate of the tubular neighborhood of a shape. We also apply this procedure to obtain an approximate, but accurate, conversion from a parametric to an implicit surface.

1 Introduction

In this paper we exploit multiscale edges as a characteristic of the implicit function that is closely related to its level surfaces. We develop a method to generate smooth implicit functions from edge information.

One of the possible choices for implicit model of a solid object is based on pseudo-metric functions. This type of function vary smoothly and has as one of its isosurfaces the boundary of the solid. In general, there are many functions satisfying these properties, and the extra degree of freedom can be used to provide control over a prescribed tubular neighborhood of the surface bounding the object.

The problem of constructing such a function is important because it gives an implicit model for which we have explicit information about the variations of the implicit function. This knowledge is crucial to robust computation with implicitly defined objects.

1.1 Modeling and Vision

One of the most general implicit representations is the volumetric description. In this form, the implicit function is usually defined by a three dimensional array of samples, or a 3D image. Therefore, it makes sense to extend the machinery developed for image analysis to the context of the implicit shape representation.

The method developed in this paper is heavily influenced by image processing techniques used in vision. One of the problems in vision is the reconstruction of a continuous image function from a discrete set of sample values. This problem is related in many ways with general implicit shape models. If we are given a volumetric representation, it may be necessary to find a continuous interpolant that describes it. More generally, if we have a function that classifies points of the ambient space in relation to a shape, it may be desirable to generate a smooth implicit function describing the shape.

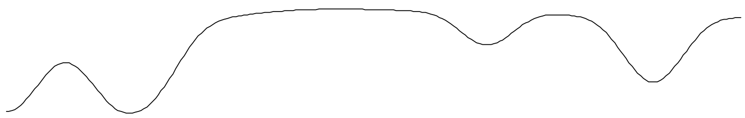
1.2 From the Characteristic Function to an Implicit Model

A characteristic function is a classification function that can be used to define a shape. It has value one at points belonging to the shape and value zero at all other points of the ambient space. The characteristic function can be used to create a smooth implicit function that also defines the shape. The basic idea is to perform a selective smoothing of the characteristic function such that the shape (and its boundary) is preserved.

The shape boundary corresponds to points where the characteristic function is discontinuous, and in an implicit model it is a level set of the implicit function. Our goal is to transform the first description into the second. This can be done by convolving the characteristic function with a smooth kernel. The main difficulty is how to perform such smoothing operation while keeping the boundary invariant.

A one-dimensional example may illustrate the problem: we start with a one dimensional shape, whose characteristic function is shown in Figure 1(a). This function is then smoothed by a space invariant low-pass filter. The resulting function is shown in Figure 1(b). Note that some of the boundary points have been displaced.

(a)



(a)

Figure 1: Smoothing of a Binary Function

If a space variant low-pass filter is employed, it is possible to obtain a smooth implict function without any displacement of boundary points. In this paper we develop a method that performs this type of smoothing to create a implicit model of a shape.

1.3 Related Work

Some of the traditional image reconstruction methods used in vision could be applied to the problem of constructing implicit shape models from boundary data. These methods, based on regularization techniques, reconstruct a hypersurface that is a function graph from sparse data [1]. More precisely, these techniques generate a function $f : \mathbf{R}^n \to \mathbf{R}$ (whose graph is a hypersurface in $\mathbf{R}^{n+1}$) that is constrained to have as one of its isocontours $f^{-1}(c)$ the input surface $\mathcal{S}$. There are several numerical algorithms to implement these methods [2], [3].

Here we take a different approach, and propose to exploit the multiscale edges corresponding to the boundary of a solid shape. Our method employs a wavelet algorithm [4] to generate a smooth implicit function. It essentially synthesizes a tubular neighborhood corresponding to the bounding surface of a solid object. The benefits of this choice are that it establishes a direct relationship with the implicit model characterization and provides better control over the properties of the resulting function.

2 Implicitly Defined Shapes

A shape can be defined in implicit form by a scalar function of the ambient space. In this section we analyze implicit shape models and the main characteristics of the implicit function.

2.1 Implicit Models

A hypersurface $\mathcal{S} \subset \mathbf{R}^n$ has an implicit description if there is a function $f : U \subset \mathbf{R}^n \to \mathbf{R}$, $\mathcal{S} \subset U$, and a value $c \in \mathbf{R}$, such that $\mathcal{S} = f^{-1}(c)$. That is,

$$\mathcal{S} = \{p \in U : f(p) = c\}.$$

The function f is called an *implicit function*, U is the domain of f, and c is called the *level* of $\mathcal{S}$. In general we require f to be differentiable of class C^k, $k \geq 1$. The implicit surface $\mathcal{S}$ is the set of points which satisfy the implicit equation $f(p) - c = 0$.

If the gradient vector field grad(f) of the implicit function f does not vanish on $\mathcal{S}$, we say that the surface $\mathcal{S}$ is a *regular implicit surface*. In this case, $\mathcal{S}$ is a differentiable manifold of class C^{k-1}.

If the domain U is the Euclidean space, $U = \mathbf{R}^n$, then $\mathcal{S} = f^{-1}(c)$ is a closed, hypersurface of $\mathbf{R}^n$, therefore it separates the Euclidean space into two connected components having $\mathcal{S}$ as a common boundary. When $\mathcal{S}$ is bounded, one of these connected components is also bounded, that is, it describes a *solid* in the Euclidean space. This shows that the implicit function f can also be used to describe implicit solids of the Euclidean space.

More precisely, an implicit solid object $\mathcal{O}$ (with boundary) is the inverse image $f^{-1}(I)$ of the interval $I = (-\infty, c]$ (or $[c, \infty)$, depending on the orientation convention).

$$\mathcal{S} = \{p \in U : f(p) \leq c\}$$

The implicit solid $\mathcal{O}$ is the set of points satisfying the inequality $f(p) - c \leq 0$.

2.2 Subdivision of Space

The implicit function is a *point-membership* classification function f that returns a value according to the relationship of a point $p \in U$, given as its argument, with the implicit surface $\mathcal{S}$ defined by it:

$$f(p) \begin{cases} > c & p \in \text{positive side of } \mathcal{S}. \\ = c & p \in \mathcal{S}. \\ < c & p \in \text{negative side of } \mathcal{S}. \end{cases}$$

The implicit function induces a subdivision of the ambient space into connected components that, depending on the sign of $f(p) - c$, correspond to the interior and exterior regions of the implicit solid $\mathcal{O}$. Figure 2 shows an example of a solid object with two connected components, one bounded and the other unbounded.

The characteristic function $\chi_{\mathcal{O}} : \mathbf{R}^n \to \{0, 1\}$ of a solid object $\mathcal{O}$, defined as $\chi_{\mathcal{O}}(p) =$

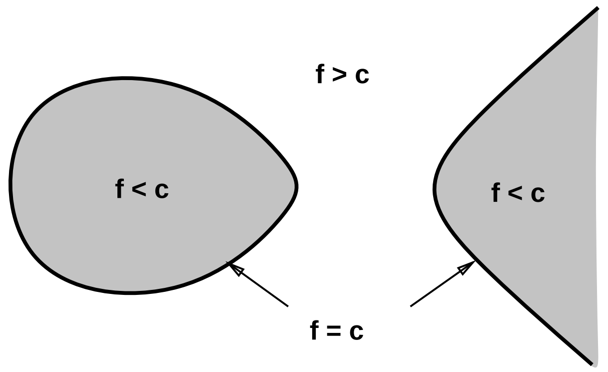


Figure 2: Implicit solid

1 if $p \in \mathcal{O}$ and $\chi_{\mathcal{O}}(p) = 0$ if $p \notin \mathcal{O}$, can be trivially derived from the implicit equation:

$$\chi_{\mathcal{O}}(p) = \begin{cases} 1 & \text{if } f(p) \leq c \\ 0 & \text{otherwise} \end{cases}$$

2.3 Tubular Neighborhoods

The concept of tubular neighborhood is of fundamental importance in the study of implicit defined shapes. It provides a geometric relationship between the implicit shape and its normal gradient vector fields. If the shape is regular, translating it along the normal directions provide an offset operation such that all of the offset shapes are also regular. This creates an infinite family of regular shapes the approximate the original shape, and provides robust techniques for performing different operations with regular implicit shapes.

A *normal segment* $[p, b]$ to a surface $\mathcal{S}$ at the point p is a line segment from p to b such that $p \in \mathcal{S}$ and $[p, b]$ is contained in $T_p^{\perp}\mathcal{S}$, the orthogonal complement of the tangent space of $\mathcal{S}$ at p. The point p is called the *foot point* of b in $\mathcal{S}$. The set of all segments normal to $\mathcal{S}$ at the point p with length less than ε, is denoted as $B^{\perp}(p, \varepsilon)$.

An *admissible normal radius* for a subset $Z \subset \mathcal{S}$ is a real number $\varepsilon > 0$ such that any two normal segments, $[p, x]$ and $[q, y]$, with $p \neq q \in Z$ and length $< \varepsilon$ do not intersect. See Figure 3.

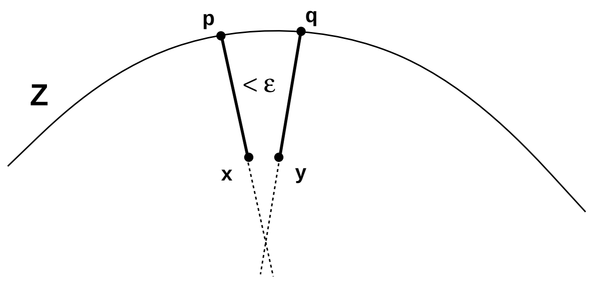


Figure 3: Admissible Normal Radius

An ε-*tubular neighborhood* V_ε of a surface $\mathcal{S}$ is defined as the union of all normal segments to $\mathcal{S}$, $[p,x], p \in \mathcal{S}$, with radius $\varepsilon(p)$ such that $\varepsilon(p)$ is an admissible normal radius for $\mathcal{S}$ at p. That is

$$V_\varepsilon(\mathcal{S}) = \bigcup_{p\in\mathcal{S}} B^\perp(p;\varepsilon(p)).$$

See Figure 4.

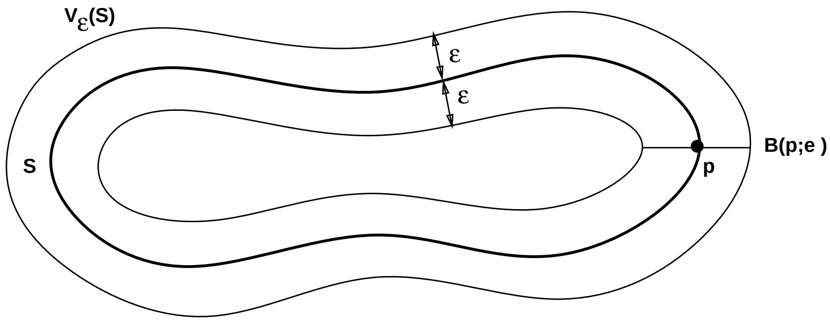


Figure 4: Tubular Neighborhood

It is an easy task to prove that any regular implicit hypersurface $\mathcal{S} = f^{-1}(c)$ has a tubular neighborhood. This is a consequence of the fact that the gradient vector field of f does not vanish on $\mathcal{S}$.

3 Multiscale Edges

An edge of a real valued function is associated with points of sharp variation of the function. These points are among the most fundamental features for analyzing structural properties of the function.

3.1 Variation and Regularity

The behavior of a function can be precisely described through the Lipschitz and Hölder conditions.

A function $f : U \subset \mathbf{R}^n \to \mathbf{R}$ is Lipschitz with constant K, $K > 0$, if, for all pairs of points p and q in U, f satisfies

$$\|f(p) - f(q)\| \leq K|p - q|,$$

where $\| \; \|$ is a norm in $\mathbf{R}^n$. This condition implies that f is differentiable and has bounded derivative. The variations of the function f can be measured by its Lipschitz constant.

A function f is Hölder continuous at a point $p \in U$, with exponent α, if there exists constants $\alpha > 0$, and $K > 0$, such that

$$\|f(p) - f(q)\| \leq K|p - q|^\alpha$$

for all points q in a neighborhood of p. The Hölder condition gives information about the differentiability of the function f at the point p.

The function f is uniformly Hölder, with exponent α, over the domain U if there exists a constant K such that the above condition holds for any $p, q \in U$.

The Hölder regularity α_p of a function f is the superior bound of all values α such that f is Hölder α at p. Note that the Lipschitz condition is a particular case of the uniform Hölder condition for which $\alpha = 1$.

When $n < \alpha_p < n + 1$, $n \in \mathbf{N}$, f is n times differentiable at p but its n-th derivative is not continuous at that point.

3.2 Edges and Scale

The study of a function at different scales gives information about its behavior on neighborhoods of variable size. Particularly, the evolution of edges across scales reveals the nature of function variations over its domain.

The definition of edges at multiple scales requires the auxiliary notion of smoothing. A *smoothing function* is any real function θ whose integral is non-zero. The smoothing function corresponding to scale s is

$$\theta_s(x) = \frac{1}{s}\theta(\frac{x}{s}).$$

In the context of signal processing, θ can be interpreted as the impulse response of a low-pass filter. One example of such a function is the Gaussian.

The edges of a function f at scale s, or *multiscale edges*, are defined as the sharp variations of $f(x)$ convolved with $\theta_s(x)$.

For the special case of a Gaussian, the characterization above has a simple physical interpretation. Assuming that the function f measures spatial density, its representation at a given scale s is the result of a diffusion process whose duration is proportional to s. In the 1-dimensional case, this gives a one-parameter family of functions $F(x;s)$ that is a solution of the *diffusion equation*

$$\frac{d^2F}{dx^2} = \sigma\frac{dF}{ds}$$

with boundary condition $F(x;0) = f(x)$, where σ is a diffusion constant.

The kernel of this partial differential equation is the Gaussian. So, instead of solving the diffusion equation, f may be convolved with a Gaussian filter that replaces the density with its average value over a window of diameter s. When s is small, the smoothing of f by θ_s is negligible and its edges provide the locations of sharper function variations.

Edges of a function can be detected from the information provided by its first or second derivatives. In one dimension, the location of multiscale edges are, by convention, at inflection points of the function f smoothed by θ_s. These points correspond to the extrema of the first derivative, or equivalently, to the zero-crossings of the second derivative of the smoothed function. This is illustrated in Figure 5.

The statements of the previous paragraph are easily generalized to dimension n. The edges of a function $f : \mathbf{R}^n \to \mathbf{R}$ at scale s are defined as the set of points where the magnitude of the gradient of f convolved with θ_s is maximum in the direction indicated by the gradient vector. Thus, edge points are the inflection points of the hypersurface $g = \text{graph}(f * \theta_s)$.

3.3 Edges in Implicit Defined Shapes

In the case of implicit shape models, edges are generally located at the boundaries determined by some level surface. This is a consequence of the fact that in order to define a surface $\mathcal{S} = f^{-1}(c)$, the value of f has to go from $c - \varepsilon$ to $c + \varepsilon$ in the direction normal to $\mathcal{S}$.

Although the implicit function f may present sharp variations in other parts of its domain without crossing the level c, for implicit shape models the only interest is in the variations producing that isocontour.

Normally, it is reasonable to assume that all variations of an implicit function f are associated with some family of surfaces that is defined by f.

4 Edge Detection and Reconstruction

In this section, we review a method for edge detection and reconstruction with wavelets that was developed by Mallat and Zhong in [6]. This method will be used in our process to generate a smooth implicit function from multiscale edges.

The wavelet transform is a mathematical tool for the analysis of function variations at multiple scales [5]. In particular, an appropriately chosen dyadic wavelet transform allows the detection of multiscale edges very efficiently [6]. From this type of information it is possible to reconstruct a close approximation of the function [4].

4.1 Edge Detection with Wavelets

As we have seen, multiscale edges of a function f are the inflection points of a smoothed version of f, $f_s = f * \theta_s$, obtained by convolving it with an appropriate low-pass filter θ_s.

The wavelet transform of a one-dimensional function is defined as a convolution with a family of functions $\psi_{s,x}$,

$$(T_\psi f)(s,x) = < f, \psi_{s,x} > = f * \psi_{s,x} \quad ,$$

where s is a scale parameter and x is a translation parameter. Therefore, it constitutes a good mechanism to compute multiscale edges.

We can define a wavelet ψ which is the derivative of a scaling function ϕ (in the present context, ϕ is a smoothing function). If ψ is chosen to be the second derivative of ϕ, then the edges of f at multiple scales are given by the zero-crossings of $(T_\psi f)(s,x)$. Similarly, if ψ is chosen to be the first derivative of ϕ, then the edges of f correspond to the extrema of $(T_\psi f)(s,x)$. This is illustrated in Figure 5.

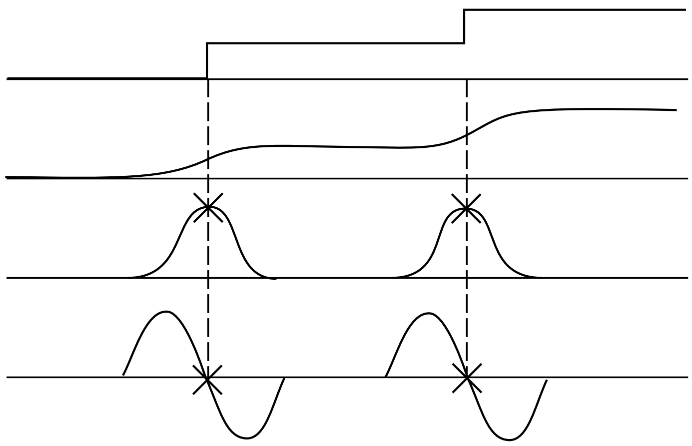


Figure 5: Multiscale Edge Detection (from Mallat)

As observed in [6], when ϕ is a Gaussian, the first procedure is exactly a Marr-Hildreth edge detection [7] and the second is the same as a Canny edge detection [8]. In principle, these two procedures are equivalent but the extremum detection has some advantages because it allows to distinguish between different types of edges [9].

It is implied in the above definition of $\psi_{s,x}$ that the parameters s and x vary continuously, therefore $(T_\psi f)(s,x)$ is a continuous wavelet transform. For most purposes, a continuum of scales is not required. The best option is to employ a dyadic wavelet transform in which the scale parameter varies along the sequence $(2^j)_{j\in\mathbf{Z}}$. It provides a natural decomposition of the frequency domain and allows fast numerical implementations [4].

4.2 Analysis of Multiscale Edges

The dyadic wavelet transform allows a complete characterization of the singularities of a function. If the wavelet is the first derivative of a smoothing function, the multiscale information gives a precise description of the function edges as well.

The evolution across scales of the wavelet transform depends upon the Hölder regularity of the function. From the maxima of the wavelet transform it is possible to compute the Hölder exponent at singular points. The decay of $(T_\psi f)$ also allows one to measure smooth variations of the function. This is stated precisely in the theorem below.

Theorem 1 *A function $f : \mathbf{R} \to \mathbf{R}$ is Hölder continuous with exponent α over an interval I if there exists a $K > 0$ such that the wavelet transform of f satisfies*

$$\|(T_\psi f)(2^j, x)\| \leq K(2^j)^\alpha,$$

for all points $x \in I$.

A proof can be found in [9].

The Hölder regularity is computed by finding the constant K and the exponent α such that $K(2^j)^\alpha$ best approximates the decay of $|(T_\psi f)(2^j, x)|$ over all scales 2^j.

This gives a method to discriminate different types of singularities of the function f. When f is not singular, its smoothness at a point x_0 is modeled as the convolution of a function h, that is singular at x_0, with a Gaussian of variance σ^2.

If the wavelet ψ is the derivative of a function ϕ that is close to a Gaussian in the sense that $\phi_{2^j} * g_\sigma \approx \phi_{s_0}$, with $s_0 = \sqrt{2^{2j} + \sigma^2}$, then the wavelet transform of f can be written as

$$(T_\psi f)(2^j, x) = 2^j \frac{d}{dx}(f * \phi_{s_0})(x) = \frac{2^j}{s_0}(T_\psi h)(s_0, x),$$

where $(T_\psi h)(s_0, x)$ is the wavelet transform of h at scale s_0. This shows that the wavelet transform at scale 2^j of a singularity smoothed by a Gaussian of variance σ^2, is equivalent to the wavelet transform of a non-smoothed singularity at scale $\sqrt{2^{2j} + \sigma^2}$.

It is not a coincidence that such an approach is essentially the same as the one used in the definition of multiscale edges (see Section 3).

Although the above theorem only characterizes Hölder exponents over intervals, it is sufficient to study isolated singularities and therefore it applies to edge detection [9]. The theorem also extends trivially to n-dimensions.

4.3 Reconstruction from the Wavelet Maxima

A method to reconstruct a close approximation of a function from the maxima of its wavelet transform was developed in [4]. It relies on the fact that a rich representation of a function f is given by the local extrema of the wavelet transform $(T_\psi f)(2^j, x)$ of f with respect to the wavelet $\psi = \frac{d\phi(x)}{dx}$ and by the values of $(T_\psi f)(2^j, x)$ at the corresponding locations, (i.e. the multiscale edges of f).

The algorithm employs a spline wavelet function that is the first derivative of a cubic spline, similar in shape to a Gaussian. A wavelet with only one vanishing moment is chosen because the main interest is in a characterization of sharp transitions of the function itself.

The reconstruction is accomplished through an iterative procedure based on alternating projections. The derivation of the method follows from a characterization of the set of functions g which has the same wavelet transform maxima as the function f.

It is intuitive that, for any scale 2^j, there exists an infinite set of functions g_j with local extrema equal to $(T_\psi f)(2^j, x)$. However, such a sequence of functions $(g_j)_{j \in \mathbf{Z}}$, is the dyadic wavelet transform of a function if and only if it satisfies the reproducing kernel associated with ψ [9].

The reconstruction problem can be broken into two conditions. Let $(x_{j,i})_{i\in\mathbf{Z}}$ be the abscissa where $|(T_\psi f)(2^j,x)|$ is locally maximum. The wavelet transform of g, at each scale 2^j, has to satisfy the following:

1. The local maxima of $|(T_\psi g)(2^j,x)|$ are located at $(x_{j,i})$.
2. For each local maximum at $x_{j,i}$, $(T_\psi g)(2^j,x_{j,i}) = (T_\psi f)(2^j,x_{j,i})$.

Define the set Γ of all sequences $(g_j)_{j\in\mathbf{Z}}$ such that, for all scales 2^j, g_j has the same extrema as $(T_\psi f)(2^j,x)$. Define also the space V of all valid dyadic wavelet transforms $(T_\psi f)(2^j,x)$. Then, the solution of the problem above is expressed as

$$\Lambda = \Gamma \cap V = \left\{(T_\psi f)(2^j,x)\right\}, \quad j\in\mathbf{Z}.$$

In order to find the solution, it is necessary to compute the intersection of Γ with V. One can prove that, since Γ is an affine space and V a Hilbert space, alternate projections on Γ and V converge to the orthogonal projection on Λ. This is illustrated by Figure 6.

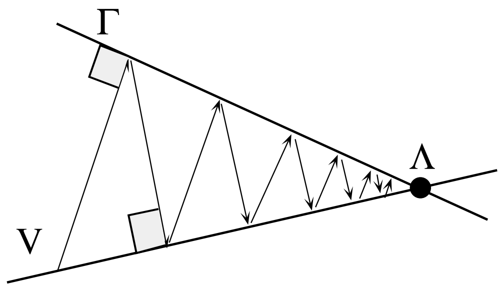


Figure 6: Alternating Projections

The orthogonal projection on the space V is implemented by performing a inverse wavelet transform T_ψ^{-1} followed by a direct wavelet transform T_ψ. This is a consequence of the fact that any dyadic wavelet transform is invariant under the operator

$$P_V = T_\psi \circ T_\psi^{-1}.$$

The operator P_Γ is implemented by adding piecewise exponential curves to each function of the sequence (g_j) such that for any index $j \in \mathbf{Z}$ and all maxima positions $x_{j,i}$,

$$g_j(x_{j,i}) = (T_\psi f)(2^j, x_{j,i}).$$

The reconstruction algorithm consists of applying the operator $P = P_V \circ P_\Gamma$ until convergence, because any element of Λ is a fixed point of P.

[4] conjectured that the local extrema constitute a complete representation, i.e., the intersection of Γ with V has only one element. It has been proved that this conjecture is false [10], but all functions $g \in \Lambda$ are very close to f and, in practice, the reconstruction algorithm always computes a good approximation of the original function.

5 Generating a Smooth Implicit Function

An important problem in modeling implicit defined shapes is the generation of an implicit function having the boundary of a given solid object as one of its level sets.

In this section, we develop a method to compute such an implicit function from the edge information provided by the characteristic function χ of the solid. The method is based on the wavelet edge detection and reconstruction algorithm described in Section 4. It exploits multiscale edge information to provide control over the properties of the resulting implicit function. This is done by manipulating the values of the wavelet maxima across scales.

5.1 Method

The main idea behind this method is the observation that a function $f : \mathbf{R}^n \to \mathbf{R}$ can be determined from its edges at multiple scales.

The dyadic wavelet transform maxima of f gives the location of its edges at scales 2^j, as well as a measure of how smooth f is in a neighborhood of these edges. Thus, if the maxima values are altered, then the smoothness of f changes. Under certain conditions, it is possible to make f smoother while maintaining the positions of its edges invariant. This means that an implicit function f defining a surface $\mathcal{S} = f^{-1}(c)$ may be manipulated without modifying the isosurface at level c. On the other hand, this procedure may be also used to modify the shape of $\mathcal{S}$ if that is desired.

The method consists of the following steps:

1. Start with a function $\chi_{\mathcal{O}} : U \rightarrow \{0, 1\}$, that is the characteristic function of a solid object $\mathcal{O}$ whose boundary is the surface $\mathcal{S}$.
2. Compute the dyadic wavelet transform $(T_\psi \chi_{\mathcal{O}})(2^j, x)$ of $\chi_{\mathcal{O}}$ at scales 2^j, $j = 1, \ldots, J$ as defined in Subsection 4.1.
3. Identify the local maxima of $|(T_\psi \chi_{\mathcal{O}})(2^1, x)|$ and record their locations.
4. For each point belonging to an edge of $\chi_{\mathcal{O}}$ at scale 1, analyze the evolution of the wavelet transform across scales and derive new wavelet maxima values for scales $2^j, j = 1, \ldots, J$.
5. Generate a smooth function f from the maxima values using the iterative reconstruction algorithm of Subsection 4.3.

5.2 Smoothing the Edges

The crucial part of the above method is the determination of how to derive the wavelet maxima to obtain the desired effect on a neighborhood of the edges of f. This procedure replaces the function discontinuities of $\chi_{\mathcal{O}}$ corresponding to the boundary of the solid with smoother transitions which still define the same bounding surface.

Let us first investigate the problem in one dimension and consider the wavelet transform maxima of an isolated step edge $g(x)$ (range of $g \in \{0, 1\}$). Since the function g is discontinuous at the edge, it has Hölder regularity $\alpha = 0$. Consequently, the value of the wavelet transform at that point is constant at all scales.

$$\|(T_\psi g)(2^j, x)\| = C$$

Figure 7 shows a plot of a step edge and its wavelet transform maxima at scales 2^j,

$j = 1, \ldots, 6$.

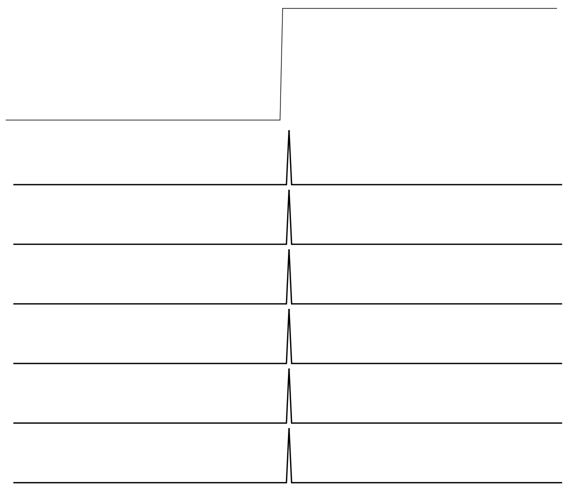


Figure 7: Wavelet Transform Maxima of a Step Edge

The goal of the smoothing procedure is to transform the singularity at the edge of g in a regular point which is an inflection point of a smooth function h (i.e. h is differentiable and its first derivative has an extremum at that point). This is equivalent to creating an edge with Hölder exponent $\alpha = 1$. The wavelet transform maxima of such an edge evolves as

$$\|(T_\psi h)(2^j, x)\| = K2^j.$$

Figure 8 shows a plot of a smooth edge and its wavelet transform maxima at scales 2^j, $j = 1, \ldots, 6$.

In order to produce a smooth edge, we select a scale 2^J as the reference and assume that $C = K2^J$. The wavelet transform maxima at all scales 2^j, $j < J$, should evolve as $C2^{j-J}$. We take C as the maxima value at scale 2 and set the maxima values at the remaining scales 2^j, $j = 2, \ldots, J$ according to the rule above. When an edge is reconstructed from these maxima values, the smoothness of the edge is proportional to the scale 2^J and the value of h at the edge is $1/2$.

When more than one edge is considered, it is necessary to detect at which scale the smoothing of one edge may cause an interference in other edges, and thus, result in an undesirable change in the function (note that the smoothing of a function may cause

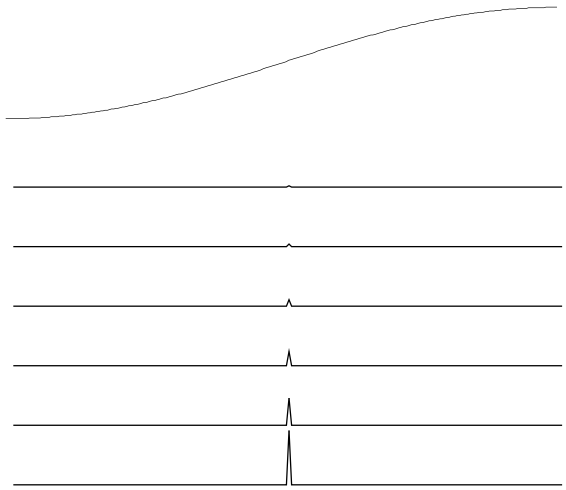


Figure 8: Wavelet Transform Maxima of a Smooth Edge

a change in its edge locations or even make close edges merge together). The region of influence of a point p at scale s is equal to the support of the smoothing function ϕ_s centered at p. Two edges do not interfere at a given scale if their regions of influence have a null intersection at that scale.

In order to be able to preserve the original edge locations in the smoothed function, the edges should be diffused to the extent that they do not interfere with each other.

This type of interference can be detected from the wavelet transform. Since we started with an original function containing only step edges, the value of the wavelet transform maxima should remain constant across scales. This will be true for all scales 2^j, unless there is an interference from the smoothing of another edge at some scale. When processing an edge, this fact is used to determine the scale of reference 2^J for smoothing the edge without interferences of other edges.

More precisely, for each edge point p, of the characteristic function $\chi_{\mathcal{O}}$ at scale 1, we analyze the evolution of the dyadic wavelet transform of $\chi_{\mathcal{O}}$ from scale 2 to scale $2^{J_{\max}}$ in order to determine at which scale the value of $(T_\psi \chi_{\mathcal{O}})(2^j, x)$ fails to remain constant. This scale 2^{J_s} is taken as the reference scale for smoothing at the point p. For scales 2^j, $j = 1, \ldots, J_s$, we create a wavelet maxima point with value $C 2^{j-J_s}$, where C is the original value of the wavelet transform of $\chi_{\mathcal{O}}$ corresponding to point p at scale

2^{J_s} (which, as we have seen, is constant for a step edge in this range of scales). For scales 2^j, $j = J_s, \ldots, J_{\max}$, we simply create a maxima point whose value is equal to $(T_\psi \chi_{\mathcal{O}})(2^j, x)$ at $x = p$.

Figure 9 shows the smoothing of a function containing several step edges based on the method described above. In the figure, the top row displays the initial and final functions superimposed, for better comparison. The other rows display the maxima values computed using the smoothing algorithm. Since the edges are spaced by different amounts, they are diffused to different degrees by the reconstruction procedure.

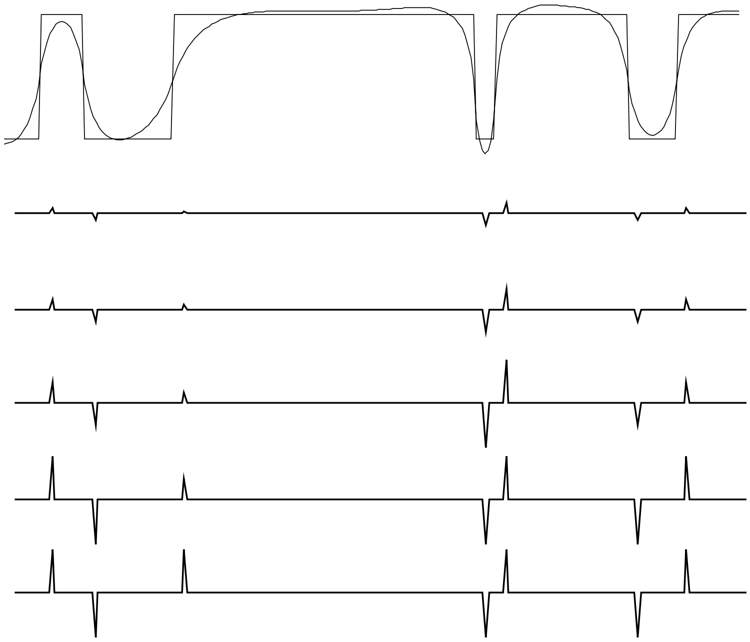


Figure 9: Smoothing of Multiple Edges

To extend the smoothing method to n-dimensions we need to consider the magnitude as well as the direction of the gradient vector field of the smoothed function.

As we have seen, in n-dimensions, the edges of a function f are defined as the points where the modulus of the gradient vector of $f * \phi$ is maximum along the direction of $\nabla(f * \phi)$. The representation of the wavelet transform in spherical coordinates gives exactly the modulus and direction angles of the gradient of f smoothed by ϕ (recall that the wavelet functions ψ_i are the partial derivatives of the smoothing function ϕ).

For an n-dimensional step edge, both the modulus and the direction of the wavelet transform remain constant across scales. For a smooth edge, the modulus of the wavelet transform evolves as in the 1-dimensional case, while its direction remains constant.

Figure 10 shows the spherical representation of the wavelet transform of the image of a 2-dimensional smooth function.

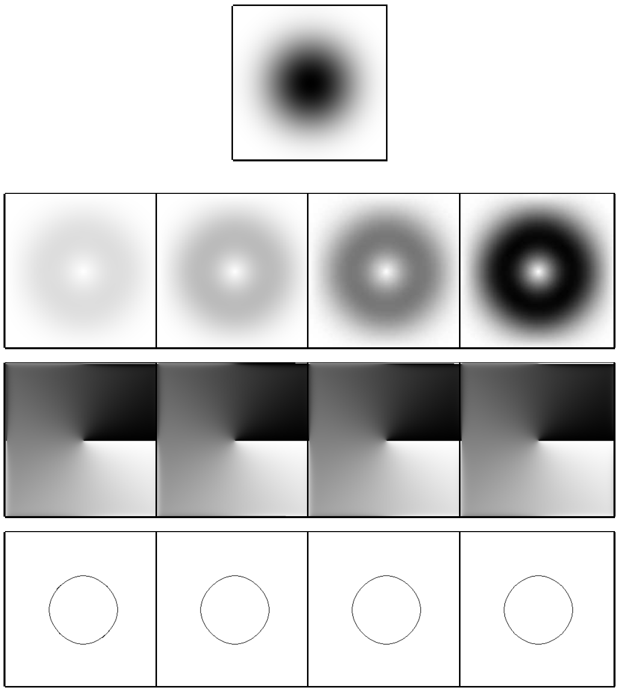


Figure 10: Spherical Representation of the Wavelet Transform of a Smooth Edge

Figure 11 is a plot of the amplitude of the modulus of wavelet transform corresponding to the central horizontal line of the image in Figure 10.

The method in n-dimensions is essentially the same as in 1-dimension, except that the modulus and the direction of the wavelet transform must be considered. The determination of the reference scale 2^{J_s} is based on the evolution of both the modulus and the direction of $(T_{\psi_i}\chi_{\mathcal{O}})(2^j; x, y)$. To create a wavelet maxima point, the value of the modulus is attenuated as in the 1-dimensional case, while the direction is maintained. This is intuitive, since the goal is to smooth the edge while preserving its location as well as its normal vector field.

We now show examples that illustrates the use of the method in two and three

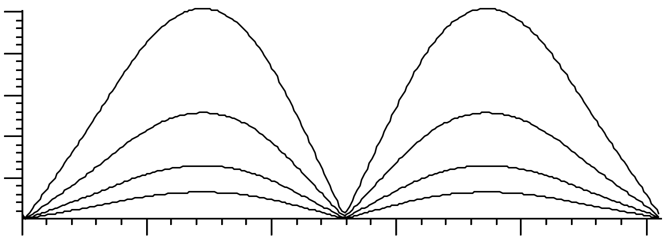


Figure 11: Evolution of the Modulus of the 2D Wavelet Transform

dimensions.

Example 1 Diamond Shape

The shape is a diamond whose characteristic function is given by the binary image in Figure 12(a). The smoothed function is shown in Figure 12(b). Note that the original image was blurred by different amounts along the edges. Near the corners there is almost no blur while at the midpoints of the edges there is extensive blurring. Figure 12(c) shows the isocontour of the smoothed image corresponding to gray level 0.5.

Example 2 Sphere

The shape is a ball defined by a density array of $64 \times 64 \times 64$ samples. A smooth implicit function is generated from this data set using the method. The surface corresponding to isovalue 0.5 is inferred from the smooth function and a polygonal approximation is produced. Figure 13(a) shows an image of the input density array. Figure 13(b) shows the smooth implicit function. Figure 14 shows the 64 slices of the density array for the smooth implicit function. Figures 15(a) and (b) show a close view of input and output density arrays. Figure 16 shows the polygonal approximation of the isosurface extracted from the function in Figure 13(b) using a polygonization algorithm. Figure 17 shows a shaded image of the surface. Note that the sphere's surface was completely recovered in spite of the low resolution of the input array.

5.3 Controlling Smoothness

The evolution across scales of the wavelet transform maxima can be used to limit the variations of the implicit function and control the tubular neighborhood of the implicit surface.

The smoothness of the edges defining a surface $\mathcal{S}$ is related to a tubular neighborhood of $\mathcal{S}$ in which the implicit function gives useful information about $\mathcal{S}$. Particularly, the gradient of f at a point q in this neighborhood gives a path to the foot point of q in $\mathcal{S}$.

The maximum tubular neighborhood $V_\phi^{\max}(p)$, relative to a smooth function ϕ, of a surface $\mathcal{S} = f^{-1}(c)$ at a point p is given by the largest scale s for which a smooth edge can produce the level surface $f^{-1}(c)$.

There are three useful ways to determine multiscale edges for generating a smooth implicit function:

(A) Produce the maximum tubular neighborhood for ϕ, $V_{\max_\phi}(\mathcal{S})$.

(B) Produce a tubular neighborhood with prescribed inferior and superior limits, V_s, $s \in [s_0, s_1]$.

(C) Produce a uniform tubular neighborhood, V_s, $s = K$.

Note that in option (A), s_0 and s_1, correspond to the the inferior and superior limits of $V_{\max_\phi}(\mathcal{S})$. This is the maximum tubular neighborhood at each point $p \in \mathcal{S}$ that can be generated from the multiscale edges of $f * \phi$ such that the level surface $\mathcal{S} = f^{-1}(c)$ is preserved. In option (B), s_0 and s_1 are set arbitrarily. Option (C) is just a particular case of option (B), in which $s_0 = s_1$.

We remark that options (B) and (C) may cause a change in the shape of the surface if the limits exceed the maximum allowable by the smoothing function ϕ and the surface $\mathcal{S}$.

6 Applications

The generation of a smooth function f corresponding to a level surface $\mathcal{S} = f^{-1}(c)$ has various applications in modeling and graphics. It gives control over the properties of the implicit function f, providing at the same time valuable information about the tubular neighborhood of the level surface $f^{-1}(c)$. An implicit function that describes the boundary of arbitrary solid shapes can be created using the method discussed in Section 5. The solids may be defined either in implicit or in parametric form.

6.1 Approximate Conversion from Parametric to Implicit

An important application of the smoothing method is to the conversion of a parametric surface that is the boundary of a solid object to an implicit surface approximating it. The need for this type of conversion may arise in many practical situations, such as in tolerance analysis in CAD/CAM models.

The conversion is performed by first producing a volumetric representation of the characteristic function of the solid region enclosed by the parametric surface.

The volumetric representation is produced in two steps:

1. Rasterize the parametric surface.

2. Fill the volume enclosed by the surface.

Algorithms for rasterization of parametric objects can be found in [11]. An algorithm for filling in regions from their boundaries can be found in [12].

The resolution required for the volumetric representation can be estimated based on the smallest surface element, in the case of polygonal surfaces, or on surface curvature, in the case of higher order surfaces.

After the discrete characteristic function has been computed, the smoothing method described in Section 5 is applied to generate the implicit function.

Now, we give an example that demonstrates the use of the method in the conversion of parametric curves to a volumetric implicit form.

Example 3 B Shape

We start with a letter "B" described by its outline, in the form of a Bezier curve. The outline is rasterized into an image array of 256×256 resolution. The interior of the "B" shape is filled. The smoothing method is then applied to the solid "B" shape.

Figure 18(a) shows the control polygon of the two dimensional Bezier curve that describes the font outline. Figure 18(b) shows the image of the solid "B" shape after rasterization of the outline and filling its interior. Figure 18(c) shows the implicit function resulting from the smoothing process. Figure 18(d) shows the isocontour corresponding to level 0.5.

6.2 Manipulation of the Tubular Neighborhood

The tubular neighborhood of an implicit solid can be manipulated by restricting the values of the wavelet transform maxima of its characteristic function, as discussed in Section 5. This is an important application of the method, because the knowledge of the limits of the tubular neighborhood is the key to robust computation with implicit objects.

The following two examples demonstrate the versatility of the method and the degree of control provided over the characteristics of the smooth implicit function it generates. We use as a starting point the same "B" shape of example 3. The multiscale edges are created to limit the tubular neighborhood of the implicit function. Note that, as a consequence, the shape of the font is altered accordingly.

Example 4 Limiting the Tubular Neighborhood

In this example, we limit the tubular neighborhood of the implicit function from above and below. The limits are set by restricting the multiscale edge analysis to scales on the interval 2^j, $j = 3, \ldots, 6$.

Figure 19(a) shows the smooth implicit function resulting from the process. Figure 19(b) shows a set of evenly spaced isocurves in the tubular neighborhood of the function. Figure 19(c) shows the "B" shape generated by thresholding the function at isolevel 0.5.

Figure 20 depicts the two dimensional implicit function as a three dimensional height hypersurface.

Example 5 Uniform Tubular Neighborhood

In this example, we generate an implicit function with uniform tubular neighborhood. The smooth implicit function is reconstructed from the edges at scale 2^j, $j = 6$.

Figure 21(a) shows the smooth implicit function resulting from the process. Figure 21(b) shows a set of evenly spaced isocurves in the tubular neighborhood of the function. Figure 21(c) shows the "B" shape generated by thresholding the function at isolevel 0.5.

Figure 22 depicts the two dimensional implicit function as a three dimensional height hypersurface.

6.3 Modification of Implicit Functions

Another important application of the smoothing method is the modification of an implicit function leaving invariant one of its level surfaces. This may be desirable in situations where it is necessary to perform computations with the level surface in implicit form. In that case, if the original implicit function is very irregular, the smoothing process can generate an implicit function having the same level surface, that is better suited to computation.

The modification is performed by first producing a volumetric representation of the characteristic function of the interior region defined by the level surface. This is a straightforward operation that requires only to evaluate the implicit function at each grid point of the volume and compare its value with the value corresponding to the level surface. The value of the characteristic function at that point is set to 0 or 1 depending on outcome of the test (the function value is greater or less than the isovalue).

This example demonstrates the use of the method for modifying an implicit function given in volumetric form.

Example 6 Head Slice

The data set is an MRI (Magnetic Resonance Imaging) slice of a human head with resolution 256×256. We select an isocontour corresponding to some features of interest (brain, etc). To create the solid regions enclosed by these isocurves, we set to 1 or 0 the sample values whose density is respectively less or greater than the selected isodensity. The resulting binary array gives the characteristic function of the shapes of interest in the data. The smoothing procedure is applied to it.

Figure 23(a) shows the original MRI data set. Figure 23(b) shows the binary image that defines the features of interest. Figure 23(c) shows the smooth implicit function produced by the method. Figure 23(d) shows the boundary of the features corresponding to isodensity 0.5 (of the smoothed function).

Note that the method was able to perform a selective smoothing capturing all details of the selected isocurves.

7 Summary

In this paper, we developed a method to generate a smooth implicit function from the characteristic function of a solid shape. The method is based on multiscale edge synthesis and employs the wavelet edge detection and reconstruction algorithm from [4]. Although, this algorithm has been applied to image processing problems such as compression [6] and noise removal [9], it has not been applied before to geometric modeling problems.

Our method provides effective control over the variations of the implicit function and of their extent. It also allows control over the tubular neighborhood of the boundary of the generating shape. The method is very flexible and can be used to create an implicit function with the desired characteristics from practically any type of geometric description: parametric, implicit or volumetric. It computes the solution numerically using a fast discrete filtering algorithm. The resulting function is represented in the form of a volumetric array.

One application of this method is to the approximate conversion from parametric to implicit surfaces. This is a general algorithm for arbitrary shapes. Previous algorithms were restricted to particular types of models, such as rational parametric surfaces [13], [14].

References

[1] D. Terzopoulos. *Multiresolution Computation of Visible-Surface Representations*. PhD thesis, Massachusetts Institute of Technology, January 1984.

[2] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):129–139, March 1986.

[3] R. Szeliski. Fast surface interpolation using hierarchical basis functions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'89)*, pages 222–228, San Diego, California, June 1989. IEEE Computer Society Press.

[4] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 14:710–732, 1992.

[5] I. Daubechies. *Ten Lectures on Wavelets*. Number 61 in CBMS-NSF Series in Applied Mathematics. SIAM Publications, Philadelphia, 1992.

[6] S. Mallat and S. Zhong. Wavelet transform maxima and multiscale edges. In M. B. Ruskai et. al., editor, *Wavelets and Their Applications*, pages 67–104. Jones and Bartlett Publishers, 1992.

[7] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. Lon*, 207:187–217, 1980.

[8] J. Canny. A computational approach to edge detection. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 8:679–698, 1986.

[9] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. Preprint Courant Institute of Mathematical Sciences, New York University, 1991.

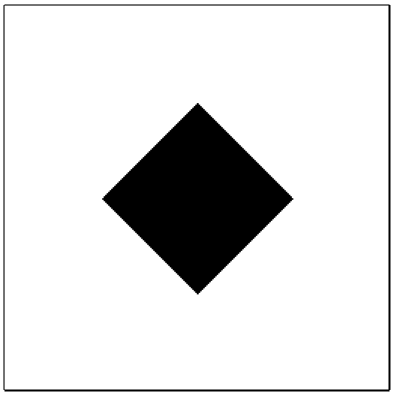
[10] Y. Meyer. *Wavelets, Algorithms and Applications*. SIAM, 1993.

[11] Arie Kaufman. Efficient algorithms for 3d scan-conversion of parametric curves, surfaces, and volumes. *Computer Graphics (SIGGRAPH '87 Proceedings)*, 21(4):171–179, July 1987.

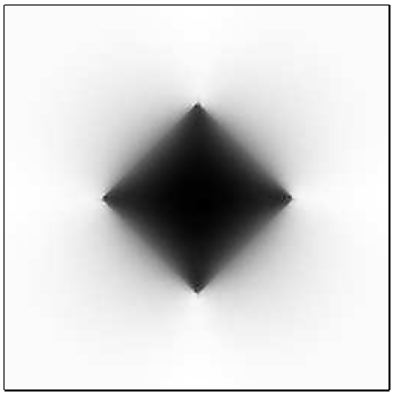
[12] T. Pavlidis. Filling algorithms for raster graphics. *Computer Graphics (SIGGRAPH '78 Proceedings)*, 12(3):161–166, August 1978.

[13] D. Manocha and J. Canny. Implicitization of rational parametric surfaces. In *The Mathematics of Surfaces*. Oxford University Press, England, 1990.

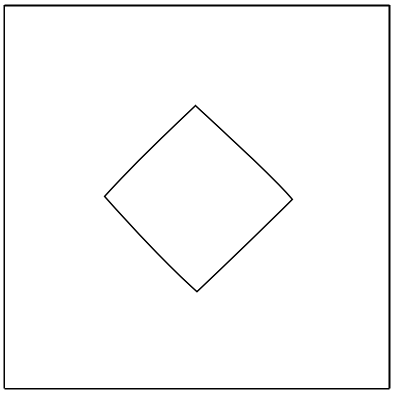
[14] M. Kalkbrenner. Implicitization of rational parametric curves and surfaces. Technical Report RISC Linz, Johannes Kepler Univ., Linz, Austria, 1990.



(a)

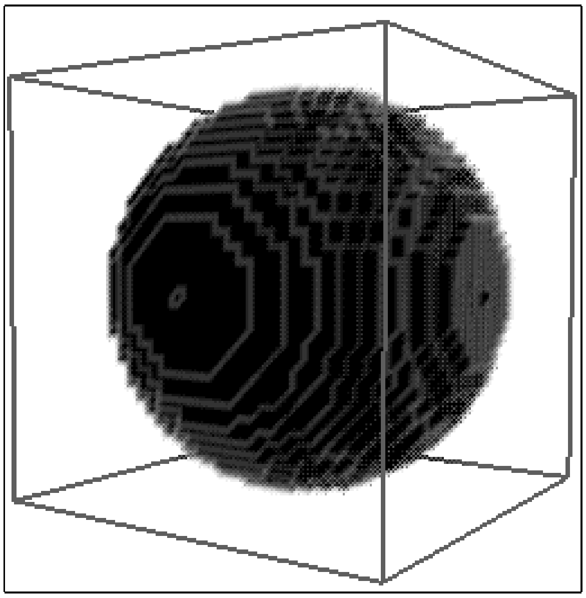


(b)

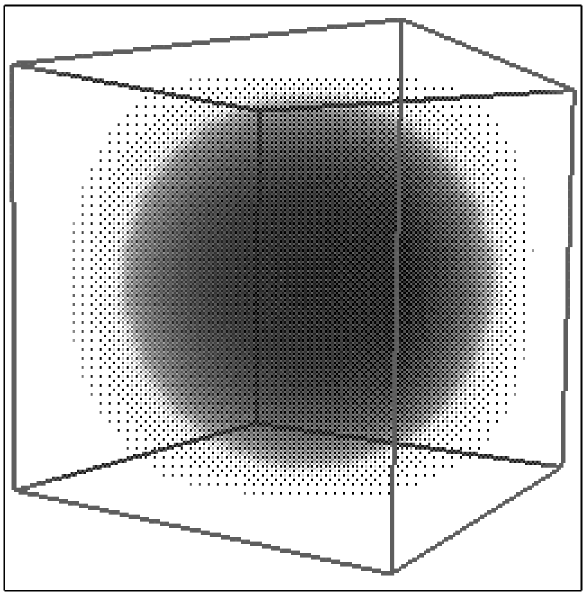


(c)

Figure 12: Smoothing in Two Dimensions



(a)



(b)

Figure 13: Smoothing of a Ball Shape: (a) input array; (b) output array

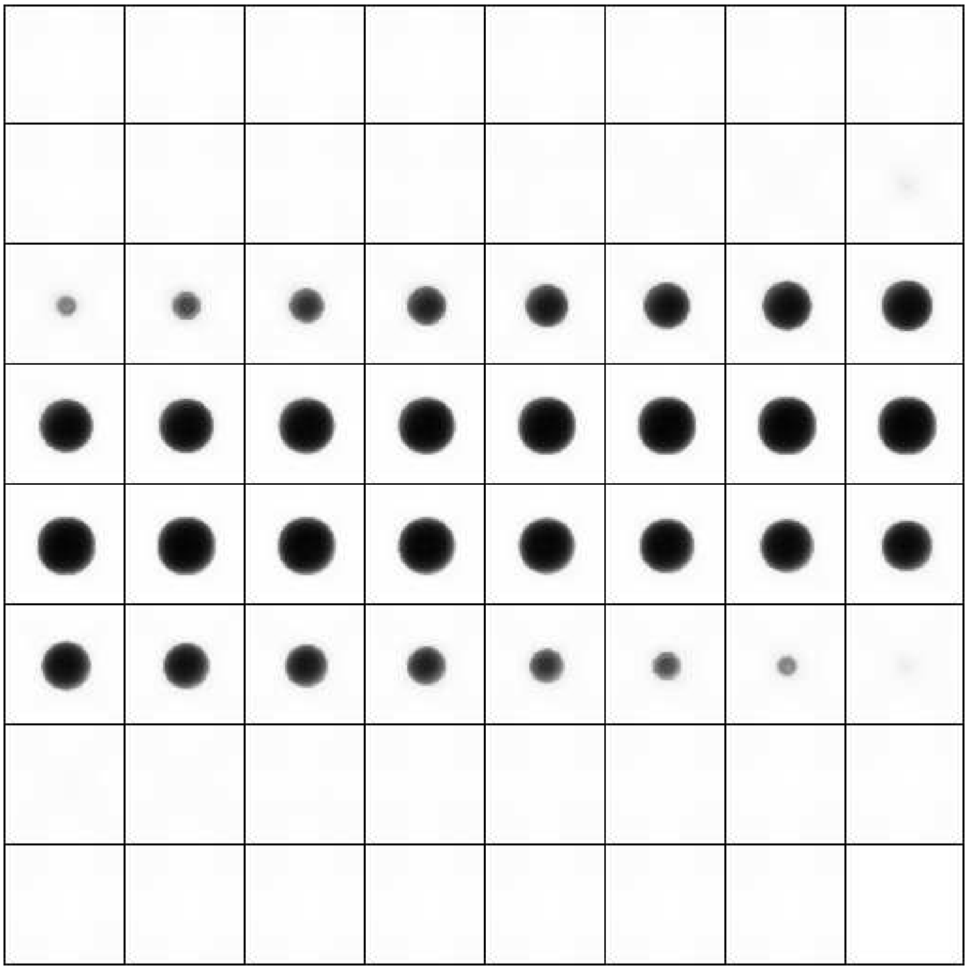
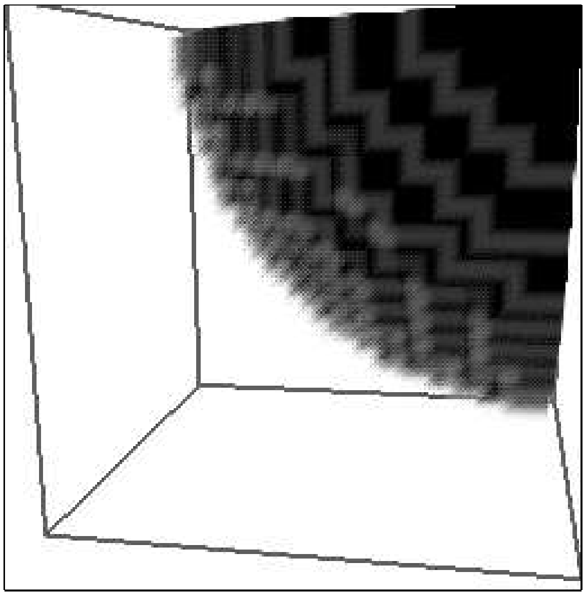
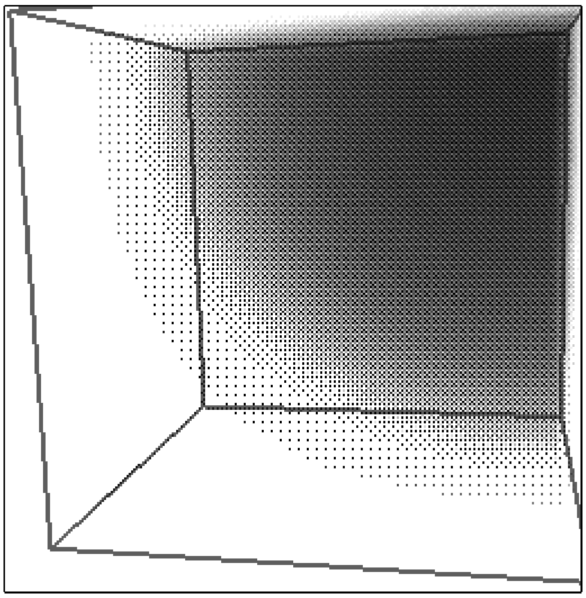


Figure 14: Slices of the 3D Array



(a)



(b)

Figure 15: Detail of the Input and Output Arrays

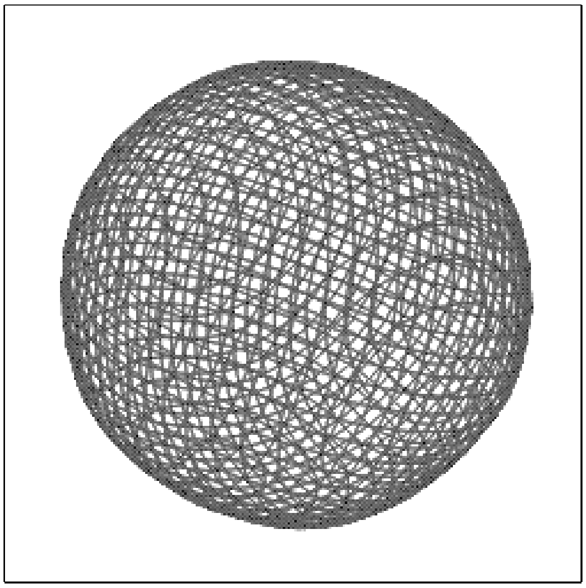


Figure 16: Wireframe of the Polygonal Isosurface Approximation

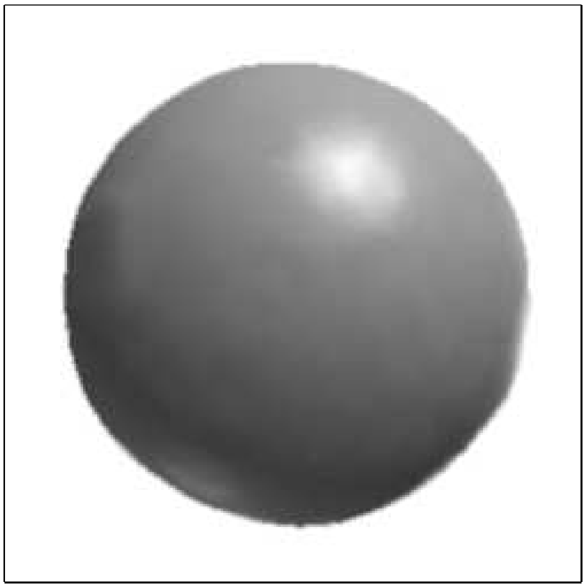
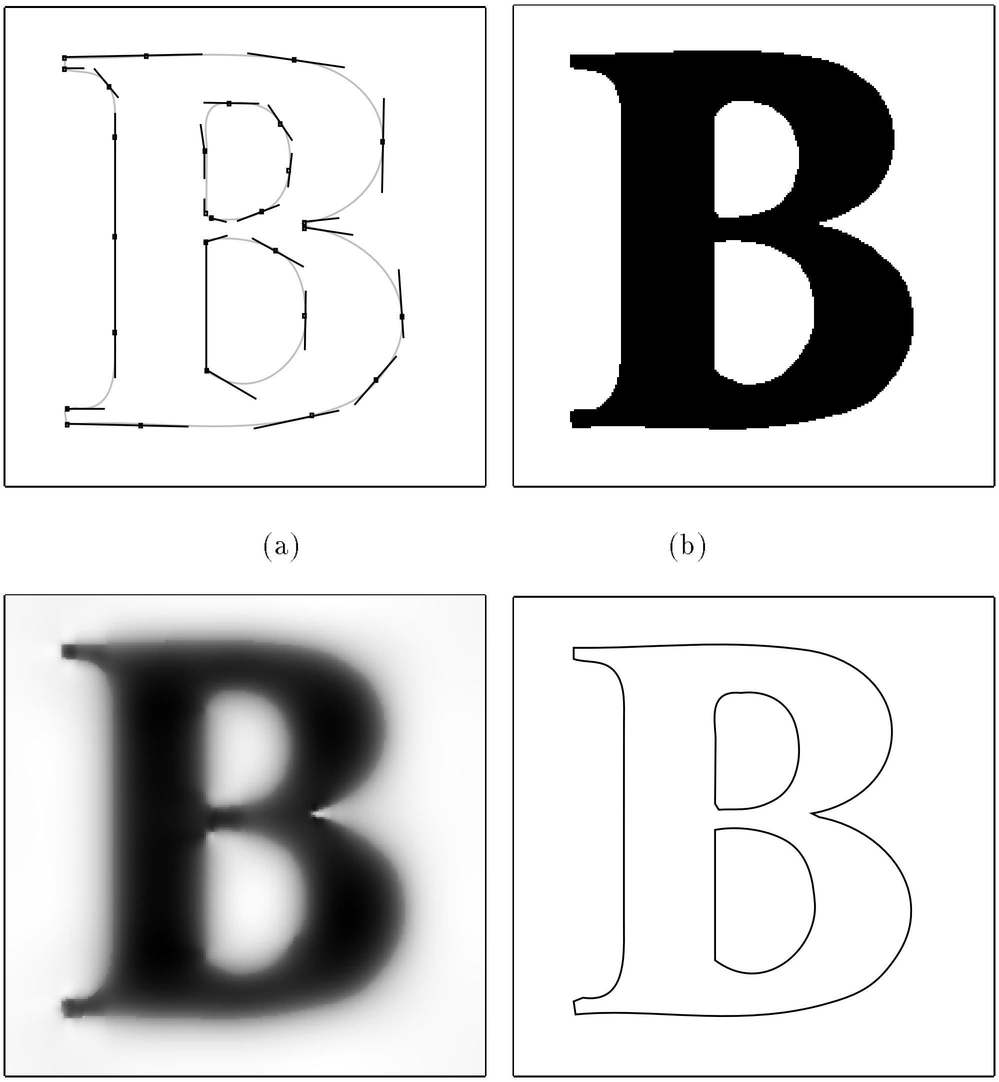
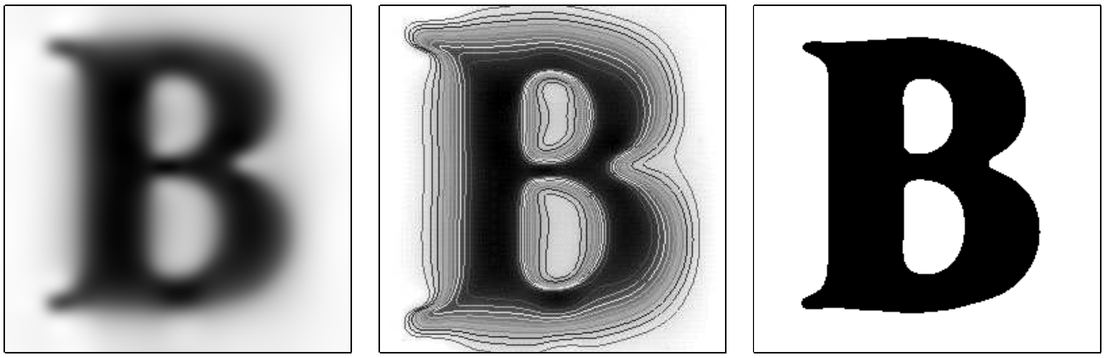


Figure 17: Shaded Image of the Polygonal Isosurface Approximation



(a) (b) (c) (d)

Figure 18: Conversion from Parametric to Implicit



(a) (b) (c)

Figure 19: Tubular Neighborhood with Limits

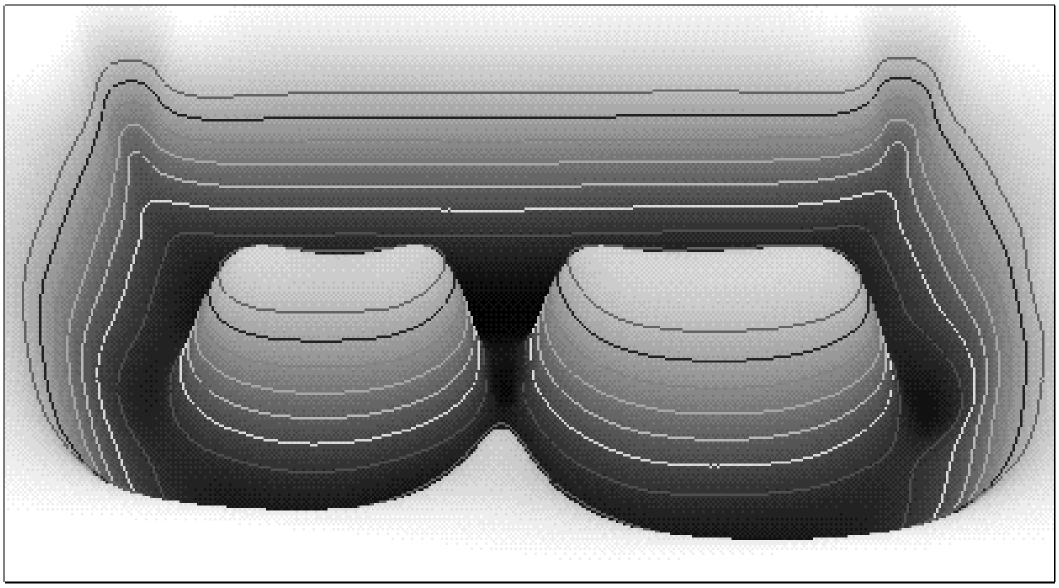
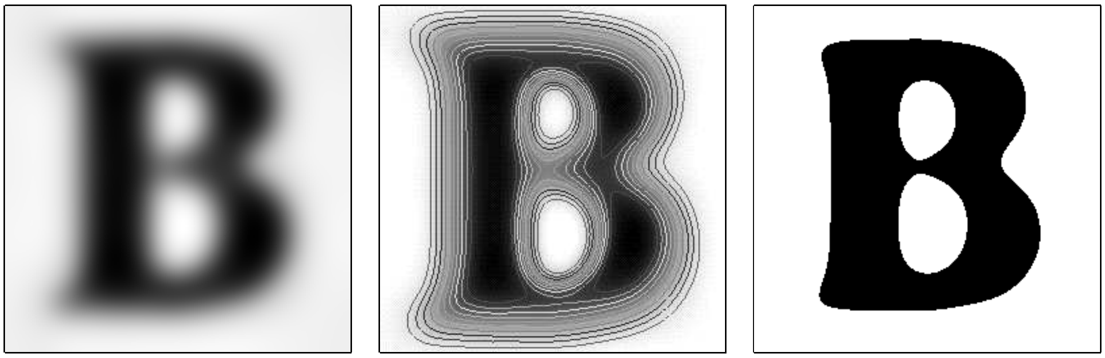


Figure 20: Implicit Function as a Height Surface



(a) (b) (c)

Figure 21: Uniform Tubular Neighborhood

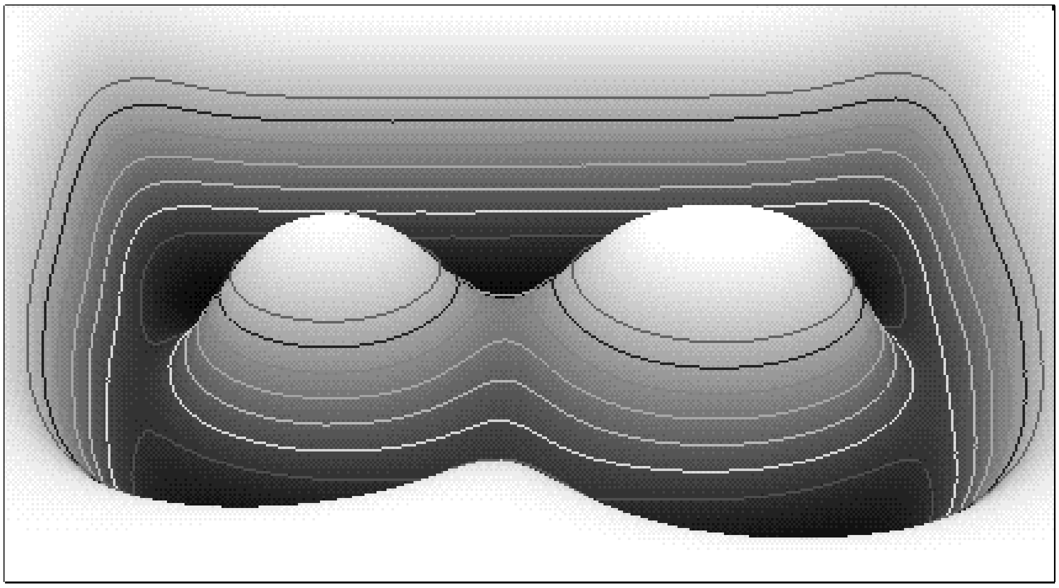
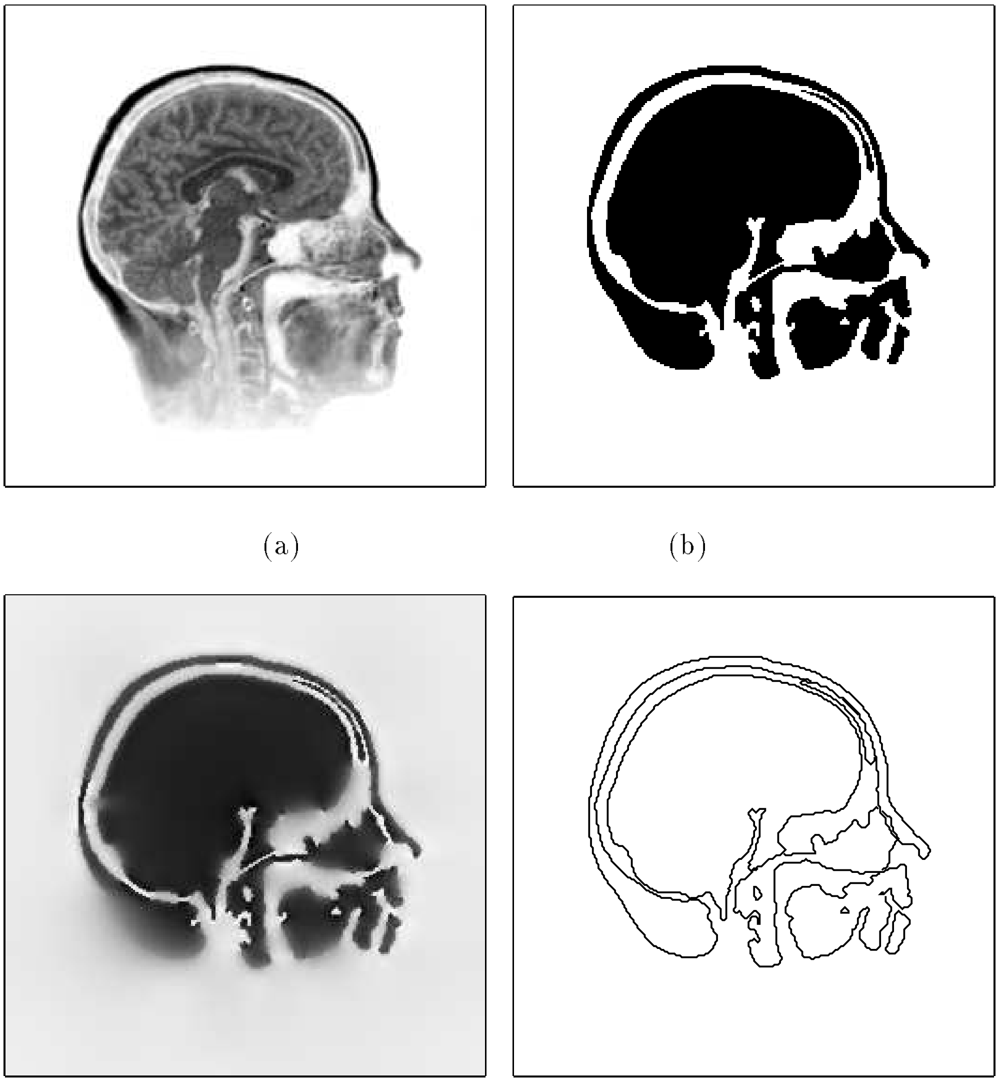


Figure 22: Implicit Function as a Height Surface



(a) (b)

(c) (d)

Figure 23: Modification of an Implicit Function