Constrained Surface Computations and Applications in Geometric Modeling

Christoph M. Hoffmann*
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907
USA

June 1992

Abstract

Many desirable operations occurring in geometric and solid modeling can be expressed and understood simply in intuitive terms. Examples include offsets, equal-distance surfaces, variable-radius blending surfaces, and connection structures that establish a point correspondence between curves. The simplicity is due to their specification in terms of certain constraints that relate to simple notions such as distance or perpendicularity. However, a precise mathematical formulation using established paradigms of CAGD and geometric modeling is difficult or impossible, and so one would have to settle for an approximation. We discuss how to express such surfaces simply and precisely, using the dimensionality paradigm.

Given that a constrained surface has been expressed with the dimensionality paradigm, we examine how to interrogate it. We consider operations such as surface/surface intersection, global surface approximation, and the determination of local surface properties such as curvature.

The medial-axis transform (MAT) was developed in computer vision as a shape abstraction. Other names for the MAT include skeleton and symmetric-axis transform. In engineering applications, current research studies how to use information provided by the MAT for automatic mesh generation and feature recognition. The MAT is closely related to concepts from descriptive geometry and from theoretical mechanics. In consequence, algorithms for computing it can take advantage of these connections directly or conceptually. We discuss such computational approaches.

*Supported in part by ONR Contract N00014-90-J-1599, NSF Grant CCR 86-19817, and NSF Grant ECD 88-03017.

Contents

1 Introduction

Geometric shape computations by computer require many extensive calculations that are based on significant mathematical theories. In geometric modeling, shapes are composed and altered in many sophisticated ways, and so it is advisable to carefully consider using a uniform, mathematical representation of the shape elements, because nonuniform representations would lead to unnecessary complexity in the programs that implement geometric operations. On the other hand, the algorithms processing a specific representation may, in certain situations, not fully utilize all geometric properties of the specific shape because the properties may not be accessible in the given representation. So, an algorithm based on a uniform representation may produce more complicated or less accurate results. The unrecognized properties might be more evident in a different representation, motivating conversion between several geometric representations. In this tension of conflicting advantages of different representations, a further difficulty arises in that not all representations are fully convertible to each other, and that, furthermore, a conversion that is possible in principle could be too expensive with the current state of the art.

This paper omits a deeper analysis of the conversion problem and the many ongoing efforts that address it from a variety of perspectives [21]. Instead, two approaches to representing surfaces and solids are discussed that are different from those widely used nowadays. The representations are motivated by a search for representations that simplify specific geometric operations that are intuitively very straightforward. That is, there are geometric operations that are simple to understand, yet remain difficult to implement with the traditional representations. In part, the traditional difficulty is due to the fact that the operations may lead outside the class of shape elements that can be represented exactly, and in part the difficulties also reflect mathematical complications that arise in specific situations in which there are degeneracies.

In the first approach to be discussed, a surface is represented as a system of nonlinear equations that formally defines a manifold in n-dimensional space, and we are interested in a projection of this manifold into a subspace. We have called this approach the *dimensionality paradigm* [17] because by raising the dimension of ambient space one can simplify the task of representing a complex surface. The advantage here is that the additional variables and equations can express, in a most natural way, geometric constraints that define the surface we are interested in. The approach allows us to deal with a broad class of geometric operations in a uniform way. Moreover, so representing surfaces and curves generalizes both parametric and implicit representations.

The second representational approach discusses the *medial-axis transform* (MAT), a concept commonly associated with computer vision and image processing but also rooted in geometry and classical mechanics. Here we consider the MAT as a representation of solids. The MAT fundamentally relates to Eu-

clidean distance. It therefore interacts with the dimensionality paradigm, first because the MAT component surfaces are simple to represent exactly with the dimensionality paradigm, and second because the MAT naturally identifies the localities at which offset surfaces, also easily defined with the dimensionality paradigm, exhibit self-intersection.

2 The Dimensionality Paradigm

We consider a surface as the natural projection of a 2-manifold in n-dimensional space into a subspace of dimension 3. The manifold is specified by a system of nonlinear equations in n variables. Usually, the equations are algebraic, although this would not be strictly necessary. This view includes the following two important special cases:

1. Implicit algebraic surfaces. We have $n = 3$, and the single equation $f(x, y, z) = 0$ is algebraic.

2. Integral parametric surfaces. Here $n = 5$, and the projection is into the (x, y, z)-subspace.

Like many generalizations, this view of surfaces is oblivious to some existing structure. For example, a parametric surface is also a mapping from $\mathbf{R}^2$ to $\mathbf{R}^3$, but in this view it is the projection of a 2-manifold in 5-dimensional space into a subspace of dimension 3.

In some geometric modeling applications, a new surface is constructed from one or more given surfaces which we will call *base surfaces*. We encode the construction as a set of equations that lay down constraints that the new surface must satisfy. Ultimately, the equations define a manifold in n-space. The surface we seek is now the natural projection of this manifold into a three-dimensional subspace. Instead of eliminating variables — a task that is frequently intractable — we will work directly with the system of (nonlinear) equations.

2.1 Surface Definition

2.1.1 An Example

We consider the definition of an equal-distance surface as an example of the dimensionality paradigm, following the presentation of [22]. When we are given two implicit surfaces $f(x, y, z) = 0$ and $g(x, y, z) = 0$, we can describe the equal-distance surface as follows:

1. Let $p = (x, y, z)$ be a point on the equal-distance surface. Moreover, let $p_f = (u_1, v_1, w_1)$ be a point at minimum distance from p on f, and let $p_g = (u_2, v_2, w_2)$ be a point at minimum distance from p on the surface g. Then:

2. The point p_f satisfies the equation of f, and the point p_g satisfies the equation of g.

3. The distance (p, p_f) is equal to the distance (p, p_g).

4. The line $\overline{p, p_f}$ is normal to f at p_f.

5. The line $\overline{p, p_g}$ is normal to g at p_g.

Assertion (1) is essentially a variable declaration. There are nine variables, comprising the coordinates of three points. Assertions (2–5) define geometric relationships that these points must satisfy.

We translate Assertions (2–5) into equations, using the variable names of (1). We obtain in sequence:

$$
\begin{aligned}
f(u_1, v_1, w_1) &= 0 \quad (1)\\
g(u_2, v_2, w_2) &= 0 \quad (2)\\
(x-u_1)^2 + (y-v_1)^2 + (z-w_1)^2 - (x-u_2)^2 - (y-v_2)^2 - (z-w_2)^2 &= 0 \quad (3)\\
[x-u_1, y-v_1, z-w_1] \cdot [-f_{v_1}, f_{u_1}, 0] &= 0 \quad (4)\\
[x-u_1, y-v_1, z-w_1] \cdot [f_{w_1}, 0, -f_{u_1}] &= 0 \quad (5)\\
[x-u_1, y-v_1, z-w_1] \cdot [0, -f_{w_1}, f_{v_1}] &= 0 \quad (6)\\
[x-u_2, y-v_2, z-w_2] \cdot [-g_{v_2}, g_{u_2}, 0] &= 0 \quad (7)\\
[x-u_2, y-v_2, z-w_2] \cdot [g_{w_2}, 0, -g_{u_2}] &= 0 \quad (8)\\
[x-u_2, y-v_2, z-w_2] \cdot [0, -g_{w_2}, g_{v_2}] &= 0 \quad (9)
\end{aligned}
$$

Partial differentiation is denoted using subscripts. For example, f_{u_1} denotes $\partial f / \partial u_1$.

Equations (1–3) express Assertions (2) and (3). Equations (4–6) together express Assertion (4), since the three vectors

$$
\begin{aligned}
&[-f_{v_1}, f_{u_1}, 0]\\
&[f_{w_1}, 0, -f_{u_1}]\\
&[0, -f_{w_1}, f_{v_1}]
\end{aligned}
$$

are tangent to f and span the tangent space as long as p_f is not a singular point on the surface. Similarly, Eqs. (7–9) together express Assertion (5). If a base surface were given parametrically, then a similar translation into equational form could be given.

The entire system of equations defines a manifold in 9-dimensional space. The projection of that manifold into the (x, y, z)-subspace contains the equal-distance surface. For other examples of surface definitions with the dimensionality paradigm see:

- [17, 18], for offset surfaces;
- [17], for constant-radius rolling-ball blends;
- [8, 17], for variable-radius rolling-ball blends;
- [12], for ruled surfaces used in parametric blending;
- [14], for trimming surfaces in skeleton computations; that is, for surfaces obtained from the normals through a fixed curve on a surface.

2.1.2 Faithful Definition Systems

The above equation system also defines certain points that are unwanted because they do not reflect the geometric intent. Such points can be present for one of two reasons:

1. Distance constraints are expressed by local extremum conditions. Thus, *global* minimum distance is not expressed.
2. Some of the equations may become dependent at certain points. For example, if p_f is a singular point, then Eqs. (4–6) vanish. Hence, the example system also defines a manifold that projects to the equal-distance surface of g and the singular point.

It is not possible to express global minimum distance without introducing inequalities, and this would significantly complicate subsequent geometric computations with surfaces so defined. In consequence, this type of extraneous solution must be excluded algorithmically, by suitably programming surface interrogation algorithms.

The possible local interdependence of the individual equations can be excluded equationally. We do this by adding more equations that encode that certain quantities are different; [23]. The idea is familiar from the refutational approaches in automated geometry theorem proving, [24, 25].

It is not always simple to define precisely what is meant by "extraneous" solution. In [23], extraneous solutions are defined as follows for equi-distance surfaces. Let $p = (x, y, z)$ be a point of the equal-distance surface, $p_f = (u_1, v_1, w_1)$ a point on f at minimum distance from p, and $p_g = (u_2, v_2, w_2)$ a point on g also at minimum distance from p. The points p_f and p_g are *footpoints* of p on f and g, respectively. Footpoints and the associated surface point(s) are said to *correspond*. Then a solution is *extraneous* if it corresponds to a footpoint that, in turn, corresponds to infinitely many solutions. Using this definition, it can be shown that all real extraneous solutions to Eqs. (1–9) must arise as follows, [23]:

1. Footpoints p_f or p_g are singular. We obtain as extraneous solutions those points that are at equal distance from the singular point and the other surface. In case both footpoints are singular but not coincident, there is an additional plane. If both footpoints are singular and coincident, then every point in $\mathbf{R}^3$ is an extraneous solution.

2. The footpoints coincide, and are regular. Moreover, the base surfaces intersect tangentially. Now the common surface normal is extraneous.

In the proof, [23] assumes that f and g are algebraic surfaces, because Bezout's Theorem is used to show that all other footpoints must correspond to finitely many points of the equal-distance surface.

We modify the system of equations to exclude extraneous points. All singular footpoints on the base surfaces are excluded by

$$\begin{aligned}(\alpha f_{u_1} - 1)(\alpha f_{v_1} - 1)(\alpha f_{w_1} - 1) &= 0\\ (\beta g_{u_2} - 1)(\beta g_{v_2} - 1)(\beta g_{w_2} - 1) &= 0\end{aligned}$$

where α and β are new variables. The equations express that not all partial derivatives of f and of g vanish simultaneously at footpoints. Next, we add the equation

$$(\gamma U_1 - 1)(\gamma U_2 - 1)(\gamma U_3 - 1)(\gamma N_1 - 1)(\gamma N_2 - 1)(\gamma N_3 - 1) = 0$$

where

$$\begin{aligned} U_1 &= u_1 - u_2 & \qquad N_1 &= f_{v_1} g_{w_2} - f_{w_1} g_{v_2}\\ U_2 &= v_1 - v_2 & \qquad N_2 &= f_{w_1} g_{u_2} - f_{u_1} g_{w_2}\\ U_3 &= w_1 - w_2 & \qquad N_3 &= f_{u_1} g_{v_2} - f_{v_1} g_{u_2}\end{aligned}$$

It expresses that the footpoints p_f and p_g are distinct, or else that the surface normals through them do not coincide. It can be shown that the resulting system has no extraneous real solutions.

In the same way, [23] excludes extraneous solutions from offsets, constant-radius blends and variable-radius blends. Obviously, the definition of extraneous solution is not suitable to surface definitions in which, conceptually speaking, points on a base curve are associated with curves on a surface. This would be the case with rolling-ball blends: If the radius remains fixed, the spine of the blending surface is the intersection of two offset surfaces. Each point on this curve of intersection is associated with a circle on the blending surface, and so "corresponds" to infinitely many points.

2.1.3 Closure

In current work, Chen and Hoffmann are exploring how the dimensionality paradigm behaves under iterated operations including offset, equal-distance,

and other surface construction operations. So far, the literature has discussed only cases in which the base surfaces are implicit or parametric. In contrast, this work extends the problem to base surfaces that are in turn constructed with the dimensionality paradigm. Roughly speaking, the operations can be iterated as long as expressions can be formulated that define the surface normal in projection. By exploiting the geometry of the individual operations, such expressions can be formulated.

2.2 Interrogating Higher-Dimensional Surfaces

A sizeable body of algorithmic infrastructure has been developed that deals with surfaces defined with the dimensionality paradigm:

1. Given two surfaces and an initial point on both, evaluate their intersection; see [4, 16, 18]. This algorithm is of the marching type. It is robust and can evaluate very high-degree surface intersections without significant precision problems.

2. Given a surface and an initial point, evaluate locally the curvatures, [12], and give a local parametric or local explicit surface approximant of arbitrary contact order, [12, 17]. The local parametric approximant generalizes the approximant used for the surface intersection algorithm.

3. Given a surface and an initial point, globally approximate the surface; [12]. The strategy of this algorithm is reminiscent of the marching-cubes technique. However, the interplay of the definitional space and the projected space makes it quite different.

These algorithms do not require the system to consist of algebraic equations, but assume that the equations are continuously differentiable.

There are many techniques for finding initial points. When the geometry of the surface is unknown, then general techniques such as [1, 2, 6, 7, 28] can be used. However, when the geometric intent of the system is known specialized methods should do much better, because the overall geometry of the target surface and the base surfaces gives many opportunities to derive initial points.

2.2.1 Local Parametric Approximation

Consider the manifold $\mathcal{S}$ defined by

$$\begin{aligned} f_1(x_1, x_2, ..., x_n) &= 0 \\ f_2(x_1, x_2, ..., x_n) &= 0 \\ &\vdots \\ f_m(x_1, x_2, ..., x_n) &= 0 \end{aligned} \tag{10}$$

where $p = (u_1, u_2, ..., u_n)$ is assumed to be a solution of the system. Assume that every hypersurface f_i is regular and twice continuously differentiable at p, and that the matrix of first-order partials

$$\left(\frac{\partial f_i}{\partial x_j} \right)_{i,j}$$

has rank $n-2$, also at p. Then $\mathcal{S}$ defined by Eq. (10) has dimension 2 in the neighborhood of p. Its natural projection into the (x_1, x_2, x_3)-subspace will be denoted by $\pi(\mathcal{S})$. We compute a local approximant to the manifold at p, using the approach of [12, 17]. More precisely, we compute n coordinate functions

$$\begin{aligned} x_1 &= h_1(s,t) \\ x_2 &= h_2(s,t) \\ &\vdots \\ x_n &= h_n(s,t) \end{aligned} \tag{11}$$

such that

$$p = (h_1(0,0), ..., h_n(0,0))$$

and

$$f_i(h_1(s,t),\ h_2(s,t),\ ...,\ h_n(s,t)) \equiv 0$$

for $i = 1, ..., m$. Note that the projection $\pi(\mathcal{S})$ of the manifold $\mathcal{S}$ into the (x_1, x_2, x_3)-subspace is approximated by

$$\begin{aligned} x_1 &= h_1(s,t) \\ x_2 &= h_2(s,t) \\ x_3 &= h_3(s,t) \end{aligned}$$

in the vicinity of the projection $\pi(p)$ of p.

The Taylor expansion of f_i at $p = (u_1, ..., u_n)$ is

$$\begin{aligned} f_i(x_1, ..., x_n) &= f_i(u_1 + \delta_1, u_2 + \delta_2, ..., u_n + \delta_n) \\ &= f_i(u_1, ..., u_n) \\ &\quad + f_i^{(1)}\delta_1 + \cdots + f_i^{(n)}\delta_n \\ &\quad + [f_i^{(1,1)}\delta_1^2 + \cdots + f_i^{(n,n)}\delta_n^2]/2 \\ &\quad + f_i^{(1,2)}\delta_1\delta_2 + \cdots + f_i^{(1,n)}\delta_1\delta_n \\ &\quad + f_i^{(2,3)}\delta_2\delta_3 + \cdots + f_i^{(n-1,n)}\delta_{n-1}\delta_n \\ &\quad + \text{higher order terms} \end{aligned} \tag{12}$$

where $f_i^{(j,k)}$ denotes the partial derivative

$$\frac{\partial^2 f_i}{\partial x_j \partial x_k}$$

We choose s and t such that $p = (h_1(0,0),\ h_2(0,0),\ ...,\ h_n(0,0))$. Then the Taylor expansion of the h_i is

$$\begin{aligned} h_k(s,t) &= h_k(0,0) \\ &\quad + h_k^{(s)} s + h_k^{(t)} t \\ &\quad + [h_k^{(s,s)} s^2 + 2h_k^{(s,t)} st + h_k^{(t,t)} t^2]/2 \\ &\quad + \text{higher order terms} \end{aligned}$$

where $h_k^{(s)}$ denotes the partial derivative of h_k by s, and so on. By assumption, there is a neighborhood of p in which

$$f_i(h_1(s,t),\ h_2(s,t),\ ...,\ h_n(s,t)) \equiv 0$$

We set

$$\begin{aligned} \delta_1 &= h_1^{(s)} s + h_1^{(t)} t + [h_1^{(s,s)} s^2 + 2h_1^{(s,t)} st + h_1^{(t,t)} t^2]/2 + \cdots \\ \delta_2 &= h_2^{(s)} s + h_2^{(t)} t + [h_2^{(s,s)} s^2 + 2h_2^{(s,t)} st + h_2^{(t,t)} t^2]/2 + \cdots \\ &\vdots \\ \delta_n &= h_n^{(s)} s + h_n^{(t)} t + [h_n^{(s,s)} s^2 + 2h_n^{(s,t)} st + h_n^{(t,t)} t^2]/2 + \cdots \end{aligned}$$

whence

$$\begin{aligned} \delta_k^2 &= (h_k^{(s)})^2 s^2 + 2h_k^{(s)} h_k^{(t)} st + (h_k^{(t)})^2 t^2 + \cdots \\ \delta_k \delta_j &= h_k^{(s)} h_j^{(s)} s^2 + (h_k^{(s)} h_j^{(t)} + h_k^{(t)} h_j^{(s)}) st + h_k^{(t)} h_j^{(t)} t^2 + \cdots \end{aligned}$$

By substitution

$$f_i^{(1)} h_1^{(s)} + f_i^{(2)} h_2^{(s)} + \cdots + f_i^{(n)} h_n^{(s)} = 0 \tag{13}$$

$$f_i^{(1)} h_1^{(t)} + f_i^{(2)} h_2^{(t)} + \cdots + f_i^{(n)} h_n^{(t)} = 0 \tag{14}$$

$$f_i^{(1)} h_1^{(s,s)} + f_i^{(2)} h_2^{(s,s)} + \cdots + f_i^{(n)} h_n^{(s,s)} = -c_i \tag{15}$$

$$f_i^{(1)} h_1^{(s,t)} + f_i^{(2)} h_2^{(s,t)} + \cdots + f_i^{(n)} h_n^{(s,t)} = -d_i \tag{16}$$

$$f_i^{(1)} h_1^{(t,t)} + f_i^{(2)} h_2^{(t,t)} + \cdots + f_i^{(n)} h_n^{(t,t)} = -e_i \tag{17}$$

where $i = 1, ..., m$. The right hand sides are, respectively,

$$\begin{aligned} c_i &= f_i^{(1,1)}(h_1^{(s)})^2 + \cdots + f_i^{(n,n)}(h_n^{(s)})^2 \\ &\quad +2[f_i^{(1,2)}h_1^{(s)}h_2^{(s)} + \cdots + f_i^{(1,n)}h_1^{(s)}h_n^{(s)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(s)}h_n^{(s)}] \end{aligned}$$

$$\begin{aligned} d_i &= f_i^{(1,1)}h_1^{(s)}h_1^{(t)} + \cdots + f_i^{(n,n)}h_n^{(s)}h_n^{(t)} \\ &\quad +f_i^{(1,2)}h_1^{(s)}h_2^{(t)} + \cdots + f_i^{(1,n)}h_1^{(s)}h_n^{(t)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(s)}h_n^{(t)} \\ &\quad +f_i^{(1,2)}h_1^{(t)}h_2^{(s)} + \cdots + f_i^{(1,n)}h_1^{(t)}h_n^{(s)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(t)}h_n^{(s)} \end{aligned}$$

$$\begin{aligned} e_i &= f_i^{(1,1)}(h_1^{(t)})^2 + \cdots + f_i^{(n,n)}(h_n^{(t)})^2 \\ &\quad +2[f_i^{(1,2)}h_1^{(t)}h_2^{(t)} + \cdots + f_i^{(1,n)}h_1^{(t)}h_n^{(t)} \\ &\qquad \cdots + f_i^{(n-1,n)}h_{n-1}^{(t)}h_n^{(t)}] \end{aligned}$$

The partial derivatives of the coordinate functions h_i are determined from these systems of linear equations, and they define an approximate local parameterization of the manifold given by Eq. (10).

The linear systems of Eqs. (13–17) have rank $n-2$. Their solutions, therefore, must have the form

$$\alpha_1 \nabla f_1 + \cdots + \alpha_{n-2} \nabla f_{n-2} + \beta \mathbf{t}_1 + \gamma \mathbf{t}_2$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ are two linearly independent tangent directions to the surface at the point p. These tangent directions are determined by the method chosen to solve the linear systems. See also [11, 17].

2.2.2 Local Curvature

As shown in [12], it is possible to determine the curvature of the surface $\pi(\mathcal{S})$ at p. The main result is

Theorem

Let $\mathbf{n}_i$ be the normal vector to the hypersurface f_i at the point p, for $1 \leq i \leq m$. Let α_i be such that the last $n-3$ components of $\mathbf{n}_0 = \sum_{i=1}^m \alpha_i \mathbf{n}_i = (a, b, c, 0, ..., 0)$ are zero, and such that $a^2 + b^2 + c^2 = 1$. Let $\mathbf{v} = (v_1, v_2, ..., v_n)$ be a tangent vector to $\mathcal{S}$ at p where $v_1^2 + v_2^2 + v_3^2 = 1$. Let H_i be the Hessian of f_i. Then the normal curvature of $\pi(\mathcal{S})$ at p in the (projected) direction $\pi(\mathbf{v})$ is given by

$$\kappa = -\mathbf{v}^T (\sum_{i=1}^m \alpha_i H_i) \mathbf{v}$$

Note that the principal curvatures and their directions can be recovered from the normal curvatures in three different directions; e.g., [32]. Thus, the principal curvatures, mean curvature, and Gauss curvature of $\pi(\mathcal{S})$ can all be determined with help of the theorem.

2.2.3 Surface Intersection

Using the same approach as described before for surface approximants, a local approximant for the surface intersection at a given point is derived. This approximant is used to derive an estimated point on the intersection that is improved using Newton iteration. At the new point, a local approximant is derived, and the above steps are repeated. Using the higher-order terms of the approximant, safe step lengths can be estimated. For details, see [4, 16].

2.2.4 Global Approximation

In contrast to pure marching and exploration schemas, a global approximation scheme requires algorithmic elements that determine whether a bounded surface area has been completely approximated. In the case of surfaces, this entails in particular determining, for closed components, whether a particular locale has already been approximated. To do so, we organize the spatial exploration using an auxiliary partition of space.

We could approximate the surface in n-space and project the approximation. Doing so is not as good as approximating the projection directly, because n-space has more independent directions, and thus the exploration using methods such as [2] slows down. Since we are ultimately interested in the projection $\pi(\mathcal{S})$, it is better to approximate the projection only. However, since the surface has been defined in a higher-dimensional space, we must lift estimated surface points somehow into the definition space.

Chuang's algorithm [11] approximates the projection only, while solving the lifting problem. A regular grid in 3-space is used to detect whether a paricular volume of space has already been explored.

1. At p, a local approximant to $\mathcal{S}$ is constructed.
2. The projected approximant, $(h_1(s,t), h_2(s,t), h_3(s,t))$, is intersected with the faces of the cube, as a function of s and t.
3. From the intersection curves with the faces, the coordinates (s_1, t_1) are determined of a point on the approximant that lies in an adjacent cube.
4. The estimated point $h_1(s_1,t_1), ..., h_n(s_1,t_1)$ is refined using Newton iteration.

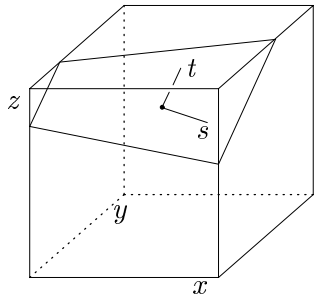

Figure 1: Linear Approximant in a Space Element and Local Frame

There is a tradeoff between the degree of the approximant, the mesh size of the grid, and the difficulty of determining face intersections and adjacent points in Steps 2 and 3. With increasing degree of the approximant a coarser mesh can be tolerated, so that fewer approximant calculations are needed. However, determining the intersection with the faces of the current cube becomes more difficult, as does the estimation of adjacent points to be explored.

We illustrate the method for linear approximants. Assume that we are at a point $p = (u_1, u_2, ..., u_n)$ on $\mathcal{S}$ that projects to $\pi(p)$ in a cube, and we have constructed the linear approximant

$$\begin{aligned} L: \quad x_1 &= u_1 + v_1 s + w_1 t \\ x_2 &= u_2 + v_2 s + w_2 t \\ &\vdots \\ x_n &= u_n + v_n s + w_n t \end{aligned}$$

L intersects the faces of the cube containing $\pi(p)$, perhaps as shown in Figure 1. Each face intersection is now determined. If the face plane is $x_1 = a$, then the line in (s,t)-space corresponding to the intersection with L is simply

$$v_1 s + w_1 t = a - u_1$$

Intersections with the planes $y = b$ and $z = c$ are analogous. So, the intersections define a polygon in (s,t)-space that corresponds to the area of L contained in the cube, as shown in Figure 2. Possibly with help of additional lines corresponding to the intersection of L with the faces of adjacent cubes, we can now find good estimates (s_1, t_1) for new points in neighboring cells, as illustrated in Figure 3.

Experiments indicate that a variation of this algorithm is more convenient: Determine the intersection of L with the *edges* of the cube in which L was constructed. The intersections are refined with Newton iteration, and the deformed polygon so obtained in each cube is triangulated, yielding a faceted approximation that is continuuous across the facet edges.

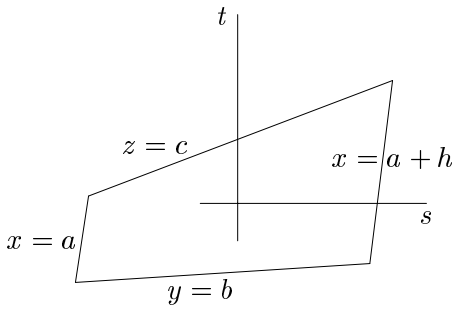

Figure 2: Corresponding Face Intersection Lines in (s, t)-Space

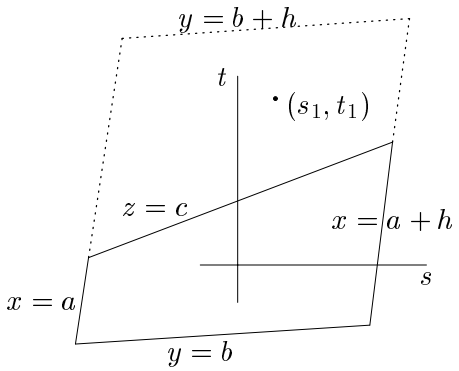

Figure 3: Finding a Point (s_1, t_1) in Adjacent Cubes

2.3 Summary

The main motivation of the dimensionality paradigm has been the observation that many geometric operations are simple to express by formulating a system of nonlinear equations, but that a subsequent elimination computation to derive an equivalent implicit equation cannot be carried out in practice, in most cases; [17]. Advances in symbolic computation will certainly shift the boundary of when the derivation of an implicit form is practical.

There is another aspect of the higher-dimensional formulation that has not been mentioned yet. Empirical data suggest that numerical geometric computations are more robust in the higher-dimensional form of a surface than in the implicit form. Since the degree of the implicit form can be extremely high even when all equations of the nonlinear system have degree no higher than 2, this observation could be simply due to working with lower degree polynomials.

3 The Medial-Axis Transform (MAT)

The *medial axis* (MA) of a three-dimensional solid is the closure of the locus of the centers of all maximal inscribed spheres. A sphere is *maximal* if there is no other inscribed sphere that contains it completely. The *medial-axis transform* (MAT) of a three-dimensional solid consists of the medial axis plus, for each medial-axis point, the radius of the sphere centered on it.

Blum has proposed the MAT as a shape representation for use in computer vision; [5, 39]. Applications of the medial axis in pattern recognition are discussed in [5, 27] and other articles. In the context of geometric modeling applications, it has been argued by several authors that the MA in 2D is useful for automatically generating finite-element meshes [3, 31, 40], because it gives information about where the 2D shape is constricted or extended, and so provides the basis for quantifying certain shape parameters needed for meshing.

The medial-axis transform is an informationally-complete solid representation in the sense of Requicha [36]. We discuss this representation and some of its applications. For simplicity, we discuss first the MAT of two-dimensional shapes. Thereafter, we discuss how to generalize to three-dimensional solids. This section follows closely the presentation of [22].

3.1 Properties of the MAT

There is an interesting connection between the MAT and certain concepts from classical geometry and from the theory of differential equations, [20]. We will discuss these connections as well.

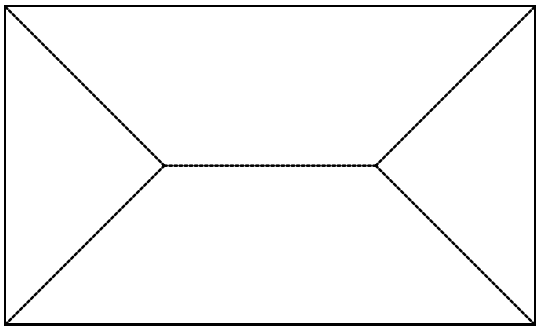

Figure 4: Interior MAT of a Rectangle

3.1.1 The MAT Representation

Let T be a bounded solid in $\mathbf{R}^2$ or $\mathbf{R}^3$ that has a smooth, compact boundary ∂T. The *distance* of the point p is defined as the minimum Euclidean distance $d(p, q)$ where q is in ∂T. For every p, there is always at least one point q on the boundary of T such that $d(p, q)$ is minimum, and such a point q will be called a *foot point* of p on T. The *interior MA* of T consists of all points p that are interior points of T and have more than one foot point on ∂T, as well as the limits of point sequences in this set. For a two-dimensional example see Figure 4. It is not difficult to see that this definition is equivalent to the earlier definition of the MA as consisting of the centers of maximal inscribed circles or spheres. Similarly, the *exterior MA* of T is the closure of all points p that are exterior to T and have more than one foot point.

We associate with each MA point its distance from ∂T. For two-dimensional solids, therefore, the MA point (u, v) at distance r from ∂T becomes the point (u, v, r) in 3-space. The MAT can therefore be thought of as a three-dimensional object, as illustrated in Figure 5. We recover the boundary ∂T from the MAT conceptually as follows: Associate with the point (u, v, r) of the MAT the circle

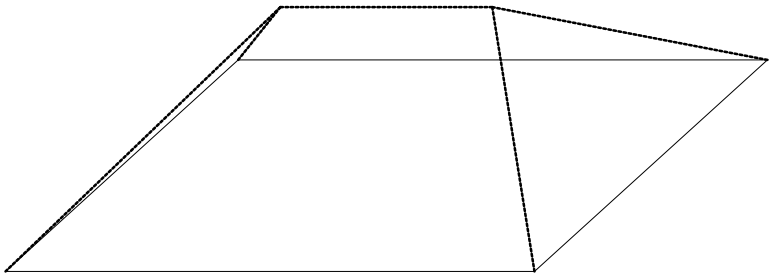

Figure 5: The MAT as 3D Object

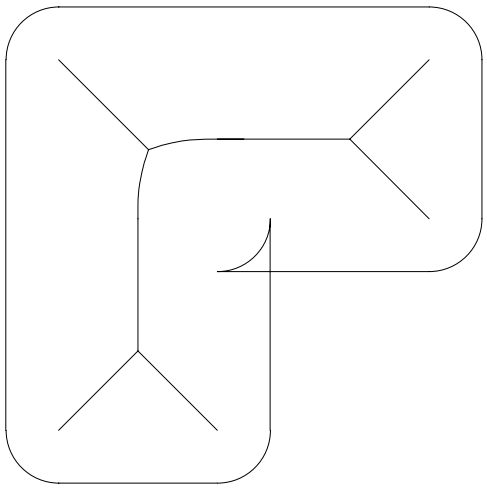

Figure 6: Self-Intersections of Positive MAT Translates

$(x-u)^2+(y-v)^2-r^2=0$. Then ∂T is the envelope of all such circles.

Likewise, the MA point (u,v,w) at distance r, of the three-dimensional solid T is represented by the point (u,v,w,r) in 4-space, thus considering the MAT a four-dimensional structure. Then ∂T is the envelope of the spheres $(x-u)^2+(y-v)^2+(z-w)^2-r^2=0$, where (u,v,w,r) is in the MAT.

Since the r-coordinate gives the distance from the boundary, we expect that a translation of the MAT, in the r-direction, represents an interior or exterior offset of T. More precisely, with d a signed translation distance, the *d-translate* of the MAT S consists of the points

$$\{(u,v,w,r+d)\,|\,(u,v,w,r)\in S,\ r+d\geq 0\}$$

Then a negative d-translate of the MAT represents the interior d-offset of T, whereas, in the case of positive d-translates, self-intersections are possible when T is not convex. See also Figure 6.

Since the boundaries of positive translates may contain self-intersections, the interior MAT does not contain sufficient information to generate offsets in a simple way: Exterior offsets require knowing the exterior MAT as well. Briefly, a positive translate of the interior MAT is combined with a negative translate of the exterior MAT, by the same distance. Then the exterior MAT points that are clipped by the rule $r+d\geq 0$ identify self-intersections. Note that the translated interior MAT is not the MAT of the true exterior offset.

3.1.2 Cyclographic Maps And Images

The concept of cyclographic maps is due to Müller [30], and has uses in descriptive geometry. Historically, the concept evolved from circle geometries investi-

gated by Laguerre. Cyclographic maps conceptualize the MAT differently, as we now explain.

We consider *cycles*, that is, oriented circles in the (x, y)-plane. The plane is embedded in (x, y, z)-space. A counter-clockwise orientation is *positive*, as seen from points with positive z coordinates, and a *negative* orientation is clockwise. If a cycle has the center (u, v) and radius r, we associate with it the point (u, v, r) in 3-space, where r is signed according to the cycle orientation. This association defines a bijection

$$\Phi : \mathbf{R}^3 \leftrightarrow \mathcal{C}$$

between points in 3-space and cycles in the (x, y)-plane. Points in the plane correspond to cycles of zero radius.

Let C be an oriented curve in the plane and consider cycles tangent to C with consistent orientation. Let p be any point on C. All cycles that are tangent to C at p are centered on the normal to C at p. Hence, the points in 3-space corresponding to the cycles lie on a line that is inclined to the (x, y)-plane by 45° and projects onto the normal of C at p. The set of all oriented circles tangent to the curve C therefore corresponds to a ruled surface in 3-space all of whose generators intersect C, project onto the normals of C, and have slope 1 against the (x, y)-plane. This surface is the *cyclographic map* of C. In particular, if C is a line, then the cyclographic map is an inclined plane, and if C is a circle or a point, then the cyclographic map is a right circular cone with a right angle at the vertex.

Now consider the inverse of the cyclographic map. Intuitively, the family of cycles corresponding to a curve in space will have an envelope, although the envelope could be imaginary. We call this envelope the *cyclographic image* of the space curve. The space curve will arise as the locus of tangent discontinuities of the cyclographic map, of some curve C, where we assume that the cyclographic map has been restricted to a graph of a function mapping the points in the (x, y)-plane to the distance from C.

Assume that the space curve is a line ℓ. Each point on ℓ maps to a cycle. If ℓ is parallel to the (x, y)-plane, then the cycles have equal radius and so their common envelope is a pair of parallel lines in the (x, y)-plane. If ℓ is inclined, with a slope less than 1, then the cycles are enveloped by a pair of intersecting lines. The intersection is the point at which ℓ intersects the (x, y)-plane. If the line ℓ has slope 1, then the envelope is a single line perpendicular to the projection of ℓ onto the (x, y)-plane. For greater inclination angles, the cycles have no real envelope. See also Figure 7. Note that the envelope of the cycles has an orientation.

Now consider a space curve D that is not a line. For each point p of D, the corresponding cycle contributes in general two points to the cyclographic image which are real and distinct if the tangent at p has slope less than 1, coincident

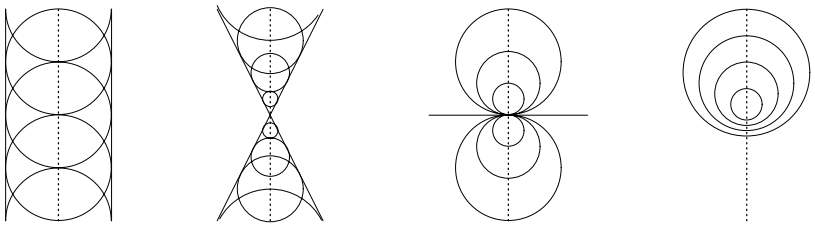

Figure 7: Cyclographic Image of a Line in Space

if the slope is exactly one, and conjugate complex otherwise. Moreover, the orthogonal projection of the space curve D onto the (x, y)-plane bisects the envelope curve C. It is not hard to see that D is the MAT of C. Furthermore, the tangent to D at p, and the tangents to the cycle corresponding to p, at the envelope points, are concurrent, [30]. See also Figure 8. This fact can be used to construct the boundary from the MAT. For other interesting geometric relationships between the space curve D and its cyclographic image see [30].

We trim the cyclographic map of the boundary ∂T of a 2D solid T as follows: For every point $q = (u, v)$ in the (x, y)-plane, the normal of the (x, y)-plane through q intersects the cyclographic map in a number of points. If q is interior of T, then we choose among the intersection points the one that lies above the (x, y)-plane and has the smallest z coordinate. If q is exterior to T, then we choose the intersection point below the (x, y)-plane whose z-coordinate has the smallest absolute value. If q is on ∂T then we choose q. With this convention, we have defined a single-valued function $\mathcal{S}$ whose domain is the (x, y)-plane and whose range is a subset of the cyclographic map of ∂T. $\mathcal{S}$ is zero only on the boundary of T, and defines the minimum Euclidean distance of every point of the (x, y)-plane to the boundary of ∂T. Clearly its singular curves are the MAT of ∂T. See Figure 9 for an example.

Since the $\mathcal{S}$ function is the Euclidean distance function, all interior and exterior offsets are its level sets and are obtained conceptually by intersecting

Figure 8: Cyclographic Image of a Space Curve

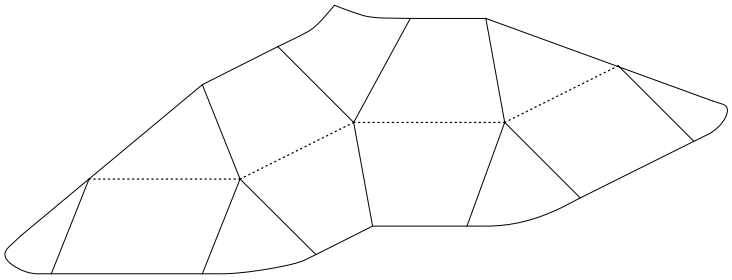

Figure 9: The Distance Function $\mathcal{S}$ Defined by the Cyclographic Map

$\mathcal{S}$ with the planes $z = d$, where d is the signed offset distance. See also Figure 9. This observation is clearly consistent with our earlier note that offsets can be obtained from translates of the MAT. The interior MAT characterizes $\mathcal{S}$ in the interior of T, but does not reflect the exterior structure of $\mathcal{S}$ completely and therefore its positive translate may generate offsets with self-intersection.

According to [38], a solid can be blended by a succession of interior and exterior offsets, by the rounding radius. The exterior offset of the interior offset obtains all rounds, and the interior offset of the exterior offset obtains all fillets. But the translation of the (x, y)-plane first by d and then by $-d$ recovers the original boundary ∂T. It follows that the trimmed cyclographic map of an offset is not equal to the translate of the trimmed cyclographic map of ∂T.

We can define cyclographic maps of three-dimensional solids analogously, establishing a correspondence between oriented spheres in 3-space, and the points of a four-dimensional space. The map of a 3D solid boundary ∂T is now a curved ruled space, and the ruling consists of lines, through every point of ∂T, that are inclined in 4-space against the embedded (x, y, z)-space by $45°$. Furthermore, the lines project onto the normals in (x, y, z)-space of ∂T and envelope its focal surfaces; [42]. The various constructions and properties sketched before for the two-dimensional case generalize virtually unchanged to the three-dimensional case [29]. In particular, the MAT of three-dimensional domains is the singular set of the trimmed cyclographic map.

3.1.3 The Hamilton-Jacobi Equation

The function $\mathcal{S}$ is a subset of the cyclographic map. In turn, the cyclographic map is ruled and each of its generators is inclined $45°$ against the (x, y)-plane, or against the (x, y, z)-space in the 3-dimensional case. With t as distance variable, the geometry of the ruling is differentially expressed by the equation

$$S_t^2 = S_x^2 + S_y^2$$

for two-dimensional solids, and by

$$S_t^2 = S_x^2 + S_y^2 + S_z^2$$

for three-dimensional solids. Locally t is an explicit function of the other variables, whence

$$S_x^2 + S_y^2 = 1$$

and

$$S_x^2 + S_y^2 + S_z^2 = 1$$

respectively. The function $\mathcal{S}$, therefore, satisfies the Hamilton-Jacobi equation subject to the boundary condition

$$S = 0 \quad \text{on} \quad \partial T$$

With proper sign conventions, the Hamilton-Jacobi equation defines the trimmed cyclographic map $\mathcal{S}$. The connection between the Hamilton-Jacobi equation and offsets has been noted in [15]. Since the MAT is the locus of points at which the trimmed cyclographic map is tangent discontinuous, the MAT is the *shock wave* in the solution of the Hamilton-Jacobi equation; [41].

3.2 Approaches to Computing the MAT

3.2.1 Discrete and Approximate Algorithms

Early approaches to compute the MA in two dimensions have evaluated an approximate distance from the boundary, by one or two passes over a discretization of T. Examples include [27, 37]. Later on Danielson devised an efficient algorithm for Euclidean distance [13], and this approach can be used effectively for discrete MAT computations; [20].

Bowyer et al. [43] describes an approximate MA algorithm for three-dimensional solids constructed from algebraic half spaces of arbitrary degree. The algorithm begins by computing in a regular grid which volume elements are inside the solid, outside, or intersect the boundary. Next, for all nonboundary elements, a distance computation determines the approximate minimum distance from the boundary elements. Those that are closest to more than one boundary element constitute the approximate MA. A number of heuristics narrow the search for closest boundary elements. The algorithm has been incorporated into the Bath solids modeler.

Price et al. [35] gives an algorithm for determining the branching points of the MA of planar domains bounded by curved edges. A Delaunay triangulation is used to approximate the MA as follows: Select a large number of points from the domain boundary and construct their Delaunay triangulation, [34]. If the points are sufficiently dense, the Delaunay triangulation will be compatible with the domain boundary. Moreover, since in a Delaunay triangulation the

circumscribed circle of every triangle contains no other triangle vertices in the interior, the MA is approximated by the centers of the circumscribing circles.

The work considers in particular how to select a small number of boundary points such that the resulting Delaunay triangulation respects the domain boundary and such that the centers of the circumscribed circles contain all branch points of the MA. Exact branch points are found later using Newton iteration. Note that the approach requires only that points on the boundary can be generated easily and as densely as necessary. The method is used as basis of a finite-element mesh generation algorithm [3].

3.2.2 Geometric Approaches

Computational geometers have investigated the MA of polygonal domains because then the curves comprising the MA are lines and parabolic arcs. Preparata's algorithm [33] evaluates first the branch points of the MA. A branch point is the intersection of the bisectors of two edge pairs. By carefully sequencing the pairs under consideration, Preparata found an $O(n^2)$ algorithm for convex polygons. Conceptually interior offsets are constructed that contain a branching point. The branch point signals that certain edges of the boundary no longer contribute to the offset.

D. T. Lee [26] constructs the MA using a divide-and-conquer approach and achieves $O(n \log(n))$. Patrikalakis and Gürsoy extend Preparata's approach to 2D domains bounded by line segments and circular arcs, [31]. In this case the MA consists of segments of straight lines and conics.

Using a similar conceptual approach, [19] sketches an algorithm for constructing the MAT of CSG solids in 3-space. The algorithm computes nearest approach points between pairs of elements of the boundary. The closest approach points found are sorted by their distance from the boundary. In this phase a proximity computation also rejects those closest approach points that are nearer to a third boundary element.

Beginning at the closest approach points, the MAT is evaluated, by increasing distance from the boundary. At each moment throughout the computation, the MAT is known up to a current distance. Closest approach points at a greater distance are considered when their respective distance has been reached. Each face, edge, and vertex of the MAT is formulated as a suitable set of nonlinear equations, using the dimensionality paradigm, and is evaluated with help of the approximation algorithm described before in Section 2.2.4. In particular, the grid used to orient the exploration of faces and edges also signals that different MAT faces are about to intersect. A proximity computation determines how to trim faces in the vicinity of an edge or a vertex.

The algorithm does not attempt to exploit special geometric configurations of the boundary elements. In [14], many such configurations have been identified.

3.3 Summary

We have described the MAT, a concept from computer vision, and how it relates to cyclographic maps, a concept from descriptive geometry, and to wave propagation governed by the Hamilton-Jacobi equation. We sketched several approaches to computing the MAT, both discrete and based on special properties of the boundary geometry. The approximate and discrete approaches are most general, because the exact geometry of the MAT edges and faces is complicated, even for boundaries whose shape elements are only quadratic. In our opinion, the complicated geometry of the MAT and the unavailability of traditional, exact representations has hindered the development of geometric algorithms for computing the MAT. Only the dimensionality paradigm and the algorithmic infrastructure described in the previous section is capable of representing MAT shape elements exactly and give a way to compute the MAT of solids with arbitrary boundary geometries. Such an approach, however, has to address the problem of finding on the MAT initial points. To find them, a promising approach is to combine the discrete algorithm of [13], with the tracing approach of [19]. This approach is currently developed by Chiang [9].

Finite-element mesh generation is currently the best-established application of the MAT; e.g., [3, 31, 40]. The relationship between the MAT and Euclidean distance, moreover, suggests that other applications could include offsetting [10] and geometric tolerancing. As robust and efficient MAT algorithms become available other important applications should develop.

Acknowledgements

The work described in this survey has been carried out by myself and by others, and has been published in original articles elsewhere, as cited.

References

[1] E. Allgower, K. Georg, and R. Miranda. Computing real solutions of polynomial systems. Technical report, Colorado State University, Mathematics Department, 1990.

[2] E. Allgower and S. Gnutzmann. An algorithm for piecewise linear approximation of implicitly defined two-dimensional surfaces. *SIAM Journal of Numerical Analysis*, 24:452–469, 1987.

[3] C. Armstrong, T. Tam, D. Robinson, R. McKeag, and M. Price. Automatic generation of finite element meshes. In *SERC ACME Directorate Research Conference*, England, 1990.

[4] C. Bajaj, C. M. Hoffmann, J. Hopcroft, and R. Lynch. Tracing surface intersections. *CAGD*, 5:285–307, 1988.

[5] H. Blum. A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, MA, 1967.

[6] B. Buchberger. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. In N. K. Bose, editor, *Multidimensional Systems Theory*, pages 184–232. D. Reidel Publishing Co., 1985.

[7] B. Buchberger, G. Collins, and B. Kutzler. Algebraic methods for geometric reasoning. *Annual Reviews in Computer Science*, 3:85–120, 1988.

[8] V. Chandru, D. Dutta, and C. Hoffmann. Variable radius blending with cyclides. In K. Preiss, J. Turner, M. Wozny, editor, *Geometric Modeling for Product Engineering*, pages 39–57. North Holland, 1990.

[9] C.-S. Chiang. *The Euclidean Distance Transform.* PhD thesis, Purdue University, Computer Science, 1992. in preparation.

[10] C.-S. Chiang, C. M. Hoffmann, and R. E. Lynch. How to compute offsets without self-intersection. In *Proc SPIE Conf Curves and Surfaces in Computer Vision and Graphics*, Volume 1610, pages 76–87. Intl Society for Optical Engineering, 1991.

[11] J.-H. Chuang. *Surface Approximations in Geometric Modeling.* PhD thesis, Purdue University, Computer Science, 1990.

[12] J.-H. Chuang and C. Hoffmann. Curvature computations on surfaces in n-space. Technical Report CER-90-34, Purdue University, Computer Science, 1990.

[13] P.-E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[14] D. Dutta and C. Hoffmann. A geometric investigation of the skeleton of CSG objects. In *Proc. ASME Conf. Design Automation*, Chicago, 1990.

[15] A. Goodman. A partial differential equation and parallel plane curves. *American Mathematical Monthly*, 71:257–264, 1964.

[16] C. M. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, Cal., 1989.

[17] C. M. Hoffmann. Algebraic and numerical techniques for offsets and blends. In S. Micchelli M. Gasca, W. Dahmen, editor, *Computations of Curves and Surfaces*, pages 499–528. Kluwer Academic, 1990.

[18] C. M. Hoffmann. A dimensionality paradigm for surface interrogation. *CAGD*, 7:517–532, 1990.

[19] C. M. Hoffmann. How to construct the skeleton of CSG objects. In A. Bowyer and J. Davenport, editors, *The Mathematics of Surfaces IV*. Oxford University Press, 1990.

[20] C. M. Hoffmann. Computer vision, descriptive geometry, and classical mechanics. In B. Falcidieno and I. Herman, editors, *Proc. Eurographics Workshop on Computer Graphics and Mathematics*, Eurographics Series. Springer Verlag, 1991.

[21] C. M. Hoffmann. Implicit curves and surfaces in CAGD. *IEEE Computer Graphics and Applications*, 12, 1992. in press.

[22] C. M. Hoffmann and G. Vaněček. Fundamental techniques for geometric and solid modeling. In C. T. Leondes, editor, *Advances in Control and Dynamics, 48*, pages 101–165. Academic Press, 1991.

[23] C. M. Hoffmann and P. J. Vermeer. Eliminating extraneous solutions in curve and surface operations. *IJCGA*, 1:47–66, 1991.

[24] D. Kapur. Geometry theorem proving using Hilbert's Nullstellensatz. In *SYMSAC 86*, pages 202–208, Waterloo, Ont., 1986.

[25] D. Kapur. A refutational approach to geometry theorem proving. In D. Kapur and J. Mundy, editors, *Geometric Reasoning*, pages 61–93. M.I.T. Press, 1989.

[26] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. and Mach. Intelligence*, PAMI-4:363–369, 1982.

[27] U. Montanari. Continuous skeletons from digitized images. *JACM*, 16:534–549, 1969.

[28] A. Morgan. *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*. Prentice-Hall, Englewood Cliffs, N.J., 1987.

[29] E. Müller. Zusammenhang zwischen relativer Flächentheorie und einer Verallgemeinerung der Zyklographie. *Jahresber. Dtsche. Math. Ver.*, pages 155–160, 1923.

[30] E. Müller and J. Krames. *Die Zyklographie*. Franz Deuticke, Leipzig und Wien, 1929.

[31] N. Patrikalakis and H. Gürsoy. Shape interrogation by medial axis transform. Technical Report Memo 90-2, MIT, Ocean Engr. Design Lab, 1990.

[32] J. Pegna and F.-E. Wolter. A simple practical criterion to guarantee second order smoothnes blend surfaces. manuscript, 1989.

[33] F. Preparata. The medial axis of a simple polygon. In *Proc. 6th Symp. Mathematical Foundations of Comp. Sci.*, pages 443–450, 1977.

[34] F. Preparata and M. Shamos. *Computational Geometry.* Springer Verlag, New York, 1985.

[35] M. Price, T. Tam, C. Armstrong, and R. McKeag. Computing the branch points of the Voronoi diagram of a object using a point Delaunay triangulation algorithm. Draft manuscript, 1991.

[36] A. Requicha. Mathematical models of rigid solids. Technical Report Memo 28, University of Rochester, Production Automation Project, 1977.

[37] A. Rosenfeld and J. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13:471–494, 1966.

[38] J. Rossignac and A. Requicha. Offsetting operations in solid modeling. *CAGD*, 3:129–148, 1984.

[39] V. Srinivasan and L. Nackman. Voronoi diagram of multiply connected polygonal domains. *IBM Journal of Research and Development*, 31:373–381, 1987.

[40] V. Srinivasan, L. Nackman, J. Tang, and S. Meshkat. Automatic mesh generation using the symmetric axis transformation of polygonal domains. Technical Report RC 16132, IBM Yorktown Heights, 1990.

[41] G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1986.

[42] K. Strubecker. *Differentialgeometrie III.* Sammlung Göschen Bd. 1180/1180a, Walter de Gruyter, Berlin, Germany, 1969.

[43] A. Wallis, D. Lavender, A. Bowyer, and J. Davenport. Computing Voronoi diagrams of geometric models. In *IMPA Workshop on Geometric Modeling*, Rio de Janeiro, 1991.