1990

How to Construct the Skeleton of CSG Objects

Christoph M. Hoffmann
Purdue University, cmh@cs.purdue.edu

Report Number:
90-1014

HOW TO CONSTRUCT THE SKELETON OF CSG OBJECTS*

Christoph M. Hoffmann†

Computer Sciences Department
Purdue University

Technical Report CSD-TR-1014
CAPO Report CER-90-33
September, 1990

* To appear in the proceedings of the 4th IMA Conference "The Mathematics of Surfaces", Bath, England, published by Oxford University Press.

† Supported in part by the National Science Foundation under grants CCR 86-19817 and DMC 88-07550, and by the Office of Naval Research under contract N00014-90-J-1599.

How to Construct the Skeleton of CSG Objects*

Christoph M. Hoffmann†
Computer Science Department
Purdue University

September 1990

Abstract

We describe an algorithm for constructing the skeleton (medial-axis transform) of three-dimensional solids defined in constructive solid geometry.

1 Introduction

Consider a bounded solid T in $\mathbf{R}^3$ with compact boundary, such as a CSG object. For every point p in $\mathbf{R}^3$, we define its *distance* from the boundary of T as the minimum Euclidean distance $d(p,q)$ where q is on the boundary of T. For every p, there is always at least one point q in the boundary of T such that $d(p,q)$ is minimum, and such a point q will be called a *footpoint* of p on T.

The *interior skeleton* of T consists of all points p that are interior points of T and have more than one footpoint on T, and the limits of point sequences in this set; i.e., the relative closure of the set of all interior points with multiple footpoints. Similarly, the *exterior skeleton* of T consists of all points p that are exterior to T and have more than one footpoint, along with their limit points. An example in two dimensions is sketched in Figure 1. In the literature, skeletons are also referred to as medial-axis transforms and Voronoi diagrams; e.g. [1, 23].

The interior skeleton can be used in pattern recognition, e.g., [1, 12]; and in finite-element mesh generation [16, 17]. For example, as argued in [17], knowing the skeleton allows one to answer a number of basic shape interrogations including detecting constrictions and their length scales, extracting holes, and decomposing complex shapes into topologically simple subdomains. Such queries

*To appear in the proceedings of the 4th IMA Conference "The Mathematics of Surfaces," Bath, England, published by Oxford University Press.

†Supported in part by the National Science Foundation under grants CCR 86-19817 and DMC 88-07550, and by the Office of Naval Research under contract N00014-90-J-1599.

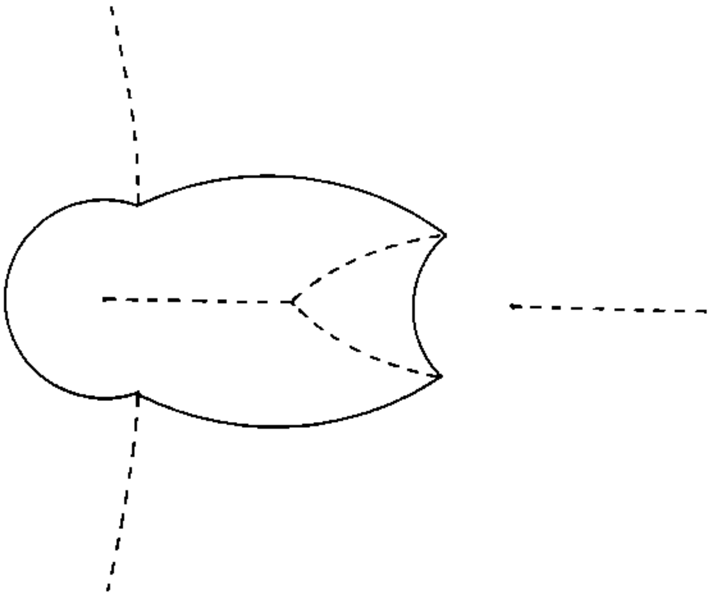

Figure 1: Interior and exterior skeletons of a simple domain

facilitate generating finite-element meshes of high quality. The exterior skeleton can be used in motion planning, e.g., [25]; and for mesh generation for computational fluid dynamics.

It is well-known that the interior skeleton can represent a solid as follows. With each skeleton point, associate its distance from the boundary of T. Consider the set of spheres centered on the points of the skeleton and of radius equal to the center's distance from the boundary. Then the envelope of these spheres is the boundary of T.

Previously published work on the skeleton has considered the problem in two dimensions, [3, 11, 12, 16, 17, 18, 23, 24]. Although an extension to three dimensions has been suggested early-on [1], very little has been published to date on this problem, unless we admit for T sets of discrete points, and include the extensive work on Voronoi diagrams; see, e.g., [19]. In [15], it is proposed to decompose three-dimensional shapes into spheres. The centers of these spheres can be thought of as a discrete approximation of the interior skeleton. [14] proves that the bisector of a linearly separable set in any dimension is a manifold.

In this paper we explain how to construct the interior skeleton of three-dimensional solids. We restrict the solids to CSG objects [20]. Doing so simplifies some of the analytical computations necessary to determine critical points and curves of the skeleton. However, except for the determination of critical points our algorithm applies unchanged to solids with curved boundaries as long as the surfaces involved are twice differentiable.

In [7] we have investigated the geometry of the surfaces that arise in the construction of the skeleton of CSG objects. Here, we study how to find initial points on the curves and surfaces comprising the skeleton and give a general algorithm for constructing the vertices, edges, and faces of the skeleton. The algorithm constructs all parts of the skeleton by increasing distance from the boundary, in a four-dimensional (x, y, z, r)-space. The resulting skeleton, in 4D, is then a complete representation of the original solid, as explained before.

2 Skeleton Construction

In [16, 17], Patrikalakis and Gursoy construct the interior skeleton of planar domains bounded by circle and line segments as follows:

1. Considering pairs and adjacent triples of boundary elements, determine a set of initial vertices in the skeleton that lie closest to the domain boundary.

2. Offset the boundary into the domain interior so that the closest initial vertices in the skeleton are on the offset boundary.

3. Repeat Steps 1 and 2 until the boundary has zero area.

Note that a boundary element is either an edge or a vertex.

The algorithm of Patrikalakis and Gursoy is a grass-fire algorithm. The principal technical reason for restricting boundary elements to circle and line segments is that the reduction in step 2 then yields a smaller domain that is again bounded by circle and line segments. Thus, no new types of boundary elements are introduced. A second reason for choosing these shape primitives is that all edges of the skeleton must be conic and line segments, so that the computation determining skeleton vertices is equivalent to intersecting two conics, a simple algorithmic problem. More complex curve segments as boundary elements generate intersection problems that would be much harder to solve.

Neither property generalizes to CSG objects. The offset of a CSG object is not necessarily a CSG object; for, while the offset of a natural quadric is again a natural quadric, the tubular offset surfaces of certain edges are not natural quadrics. Note, however, that each tube could be approximated by a CSG object using the techniques of [21, 22]. Thus, an approximate skeleton might be constructed with an algorithm that retains many of the characteristics of the planar version literally. On the other hand, it is not really necessary to construct the offset solid explicitly, for all proximity computations can be expressed in terms of the original boundary elements. Even so, determining closest bisecting points between boundary elements requires considerably more machinery than solving two bivariate quadratic equations. Note that in the three-dimensional problem, the boundary elements are the faces, edges, and vertices of the CSG object.

We use the following algorithm for computing the skeleton of a CSG object:

1. By considering all pairs of boundary elements, determine for each pair the points that are equidistant from both elements and have minimum distance.

2. Sort the points by their distance from the boundary.

3. Processing the points by increasing distance, construct the skeleton by tracing the arising edges and faces.

Steps 1 and 3 are explained in some detail in the following sections. Note that the algorithm is structurally very similar to the two-dimensional version. In contrast to the two-dimensional method, however, not all critical points of the skeleton are found. In consequence, step 3 of our algorithm includes a local adjacency analysis that can determine all adjacent edges and faces for a given skeleton point.

Note that certain equidistant (bisecting) points, found in step 1, form curves or surfaces. For example closest bisecting points between two parallel cylindrical faces are on a line, and nearest equidistant points between concentric spherical faces lie on a sphere.

In principle, our algorithm can be used for solids with more complex boundaries. However, complex curved surfaces introduce algebraic complexities into step 1. In consequence, the generalization would necessitate introducing techniques for solving general systems of algebraic equations.

A different approach can be used to construct an *approximate* skeletons as follows. First, sample the solid's surface obtaining a set of surface points. Using a robust point-set Voronoi algorithm, construct the Voronoi diagram of this point set. Then, eliminate certain Voronoi edges and surfaces thereby obtaining an approximate skeleton. For details and a two-dimensional example see [4].

3 Determining Nearest Bisecting Points

Given two boundary elements, we show how to determine nearest bisecting points between them. The boundary elements are the vertices, edges, and faces of the solid. Certain boundary elements have nearest bisecting points with respect to themselves. For instance, the center of the sphere, the central circle of a torus and its center of symmetry are such points. Likewise, the center of an edge that is a circular arc must be considered.

Edges and faces are subsets of space curves and of surfaces, respectively. We call the corresponding curve or surface the *carrier* of the boundary element. The problem of determining nearest bisecting points between two boundary elements is reduced to the problem of determining nearest bisecting points between the corresponding carriers. Briefly, if F is a face, we first determine nearest bisecting points with respect to its carrier f. Those nearest bisecting points with footpoints outside F are discarded. Next, nearest bisecting points are determined with respect to the bounding edges of F. Again nearest bisecting points with footpoints not on the edge are discarded. Finally, nearest bisecting points with respect to the vertices of F are determined. Edges are handled analogously. In consequence, determining nearest bisecting points between two (nonadjacent) faces may require determining nearest bisecting points between all edge pairs bounding the faces. Since we should determine nearest bisecting points between all pairs of boundary elements, essentially no additional work is required apart from testing whether their footpoints lie on edges or faces.

In order to find nearest bisecting points between the carriers f and g, we find pairs of footpoints p and q, that is, points p on f and q on g such that $d(p,q)$ is minimum. Such a pair will be called a *closest approach pair*. Clearly then the bisector of the pair is a nearest bisecting point.

The geometric basis for identifying a closest approach pair (p,q) is the well-known observation that the connecting line $\overline{pq}$ must be perpendicular to f at p and perpendicular to g at q. In fact, this condition also describes point pairs at which the Euclidean distance has local extrema, both minima or maxima. So, we determine *all* extremal point pairs and compute their distances discarding all that do not have minimum distance.

In the following, we describe how to compute extremal approach pairs with the surfaces and curves that are carriers of faces and edges in CSG solids. We will omit a number of special cases. For example, the closest approach pairs between two intersecting surfaces are the points on the surface intersection; the closest approach pairs between two parallel cylinders lie on corresponding generators; and so on.

3.1 A Generic Procedure for Finding Closest Approach Pairs

In order to determine closest approach pairs (p,q) between two carriers f and g, we can formulate and solve a system of algebraic equations. The system has the following structure:

$$\begin{aligned} f(p) &= 0 \\ g(q) &= 0 \\ \mathsf{perp}(p,q,f)& \\ \mathsf{perp}(q,p,g)& \end{aligned} \tag{1}$$

The first two lines assert that p is on f and that q is on g. If f is a surface given implicitly, then the first line is a single equation. If f is a curve that is the intersection of f_1 and f_2, then the line represents the two equations $f_1(p) = 0$ and $f_2(p) = 0$. Finally, if f is a point, then the line is not an equation, but the coordinates of p are the given coordinates of the point f. Similar considerations apply to the second equation.

The third line asserts that the connecting line $\overline{pq}$ is normal to f at p. With f an implicit surface, this can be expressed by three equations, as explained in [8, 9, 10]. If f is a curve, the third line represents the condition that $\overline{pq}$ is perpendicular to the tangent to f at p, [9], and if f is a point the third line expresses no condition.

For CSG objects, f and g are algebraic and therefore (1) is a system of algebraic equations whose variety should be zero-dimensional. While this approach will not generate many cases, it has the disadvantage that the system (1) is algebraically unnecessarily complex. This makes the system harder to solve and generates unnecessary candidate pairs. Therefore, we consider the arising cases in greater detail in an effort to simplify matters. In fact, many cases will require solving only a single quadratic or quartic equation, and can therefore be handled very easily.

3.2 Closest Approach Pairs Between CSG Surfaces

We consider closest approach pairs for two algebraic surfaces f and g, each of which is either a plane, a natural quadric, or a torus. We organize the various cases by the type of g, considering only the interesting surfaces f.

It is helpful to consider a point to be a sphere of radius zero, to consider a line to be a cylinder of radius zero, and to consider a circle to be a torus of minor radius zero. For example, determining the closest approach pair of two cylinders is equivalent to determining the closest approach pair between their axes. This reduces the number of cases to be considered.

3.2.1 Closest Approach Pairs with a Plane

Let g be a plane, given in implicit form by

$$a_1x + b_1y + c_1z + d = 0$$

and assume that $a_1^2 + b_1^2 + c_1^2 = 1$. Most CSG surfaces f pose uninteresting problems, except the torus. So, let f be a torus, choosing the coordinate system as shown in Figure 2, such that the axis of rotation coincides with the z-axis and the major radius is 1. Moreover, without loss of generality we assume that $b_1 = 0$.

Let $q = (a_0, b_0, c_0)$ be a point on the plane g, and let C be the interior skeleton of the torus, a circle of radius 1 in the xy-plane. The point q has a corresponding point of extremal approach on the torus if we can draw a line L from q, perpendicular to g, through the circle C to meet the z-axis, for then L will intersect the torus normally. Since $b_1 = 0$, the line L cannot be perpendicular to the plane g unless it lies in the plane $y = 0$, so we know that $b_0 = 0$. In order to pass through the circle C, furthermore, L must pass through the point $(1,0,0)$ or $(-1,0,0)$. Let $z = \lambda_0$ be the intercept of L with the z-axis. Then the point $(\lambda_0 a_0/(\lambda_0 - c_0), 0, 0)$ is the intersection with the xy-plane. So,

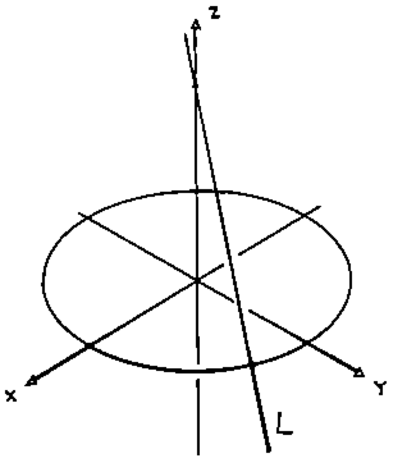

Figure 2: Torus skeleton and one torus normal

L must satisfy the following equations

$$\begin{aligned} a_0a_1 + c_0c_1 + d &= 0 \\ a_1c_0 - a_0c_1 - a_1\lambda_0 &= 0 \\ \lambda_0 a_0 &= \pm(\lambda_0 - c_0) \end{aligned} \tag{2}$$

The first equation states that q is on the plane g, the second that the line intercepting the z-axis at $(0, 0, \lambda_0)$ is normal to the plane, and the last one that this line passes through $(1, 0, 0)$ or $(-1, 0, 0)$. The unknowns are a_0, c_0, and λ_0.

For each sign choice, the system (2) consists of two linear and one quadratic equations, with two solutions. Each solution will generate two extremal points on the torus. A distance computation settles which ones have minimum distance from the plane.. Summarizing, we have

Theorem 3.1 Determining closest approach pairs between a plane and a torus requires solving two systems of three equations in three unknowns, each system consisting of two linear and one quadratic equation.

3.2.2 Closest Approach Pairs with a Sphere

From a geometric point of view, closest approach determination with a sphere is essentially a two-dimensional problem. Since all CSG surfaces are surfaces of rotations, the closest approach points will lie in a plane through the axis of rotation that contains the center of the sphere. In consequence, we expect very simple equations.

Without loss of generality, we assume that g is the point $q = (a, b, c)$, and is either a vertex or the center of a sphere. The closest approach to a cylinder f is determined by the orthogonal projection of q onto the line that is the cylinders axis. This requires solving a linear equation and is trivial.

Assume next that f is the cone

$$x^2 + y^2 - u^2z^2 = 0 \tag{3}$$

The normals to the cone lie on cones with the equation

$$x^2 + y^2 - (z - \lambda_0)^2/u^2 = 0 \tag{4}$$

Here, λ_0 is the z-coordinate of the intersection of the normals with the cone axis. To find a closest approach pair we determine λ_0 so that the point (a, b, c) lies on the corresponding cone of normals, i.e., we solve a quadratic equation in λ_0.

If f is a torus, we align the coordinate system as described in Section 3.2.1. We seek a line through the unit circle in the xy-plane, intercepting the z-axis at λ_0, and passing through the point $q = (a, b, c)$. Those lines form cones with the equation

$$x^2 + y^2 - (z - \lambda_0)^2/\lambda_0^2 = 0 \tag{5}$$

So, we determine for which value of λ_0 this cone contains the point q, i.e., by solving a quadratic equation in λ_0. In summary, we have

Theorem 3.2 Determining closest approach pairs between a point or a sphere and a line or a cylinder requires solving a linear equation, and determining closest approach pairs between a point or a sphere and a cone or a torus requires solving a quadratic equation.

3.2.3 Closest Approach Pairs with a Cylinder

Without loss of generality we replace the cylinder g with its axis, and assume that the axis is given as the parametric line

$$(a_0 + a_1\mu,\ b_0 + b_1\mu,\ c_0 + c_1\mu) \ \text{ where } \ a_1^2 + b_1^2 + c_1^2 = 1 \tag{6}$$

If f is another cylinder, we assume a coordinate system in which the cylinder axis of f is $(0, 0, \lambda)$. We assume that the two lines are skew, for otherwise the problem is uninteresting. A closest approach pair is found along the line of constriction; that is, along the common normal of the two lines. Perpendicularity to the two given lines is expressed by two linear equations. If λ_0 and μ_0 are the two (unknown) parameter values that specify the closest approach points on the two lines, the direction of the line of constriction is the vector

$$(a_0 + \mu_0 a_1,\ b_0 + \mu_0 b_1,\ c_0 + \mu_0 c_1 - \lambda_0) \tag{7}$$

So, we must solve the linear system

$$\begin{aligned} c_0 + \mu_0 c_1 - \lambda_0 &= 0 \\ \mu_0 - c_1\lambda_0 + (a_0 a_1 + b_0 b_1 + c_0 c_1) &= 0 \end{aligned} \tag{8}$$

each equation stating perpendicularity to one of the lines.

If f is a cone, we assume a coordinate system in which the cone's axis coincides with the z-axis. So the cone equation is (3). The surface normals form the cones (4), and we seek λ_0 such that the corresponding cone of normals has a generator that intersects the line g at a right angle. Let μ_0 be the parameter value specifying the intersection of the cone of normals with the line g. The generator through this point has the direction (7). Hence, the unknowns μ_0 and λ_0 are found by solving the system

$$\begin{aligned} (a_0 + \mu_0 a_1)^2 + (b_0 + \mu_0 b_1)^2 - (c_0 + \mu_0 c_1 - \lambda_0)^2/u^2 &= 0 \\ \mu_0 - c_1\lambda_0 + (a_0 a_1 + b_0 b_1 + c_0 c_1) &= 0 \end{aligned} \tag{9}$$

consisting of a linear and a quadratic equation. Here the first equation states that the line g intersects the cone of normals at the point specified by μ_0, and the second equation states that the corresponding generator is perpendicular to g.

If f is a torus, we assume a coordinate system as described before, in Section 3.2.1 and illustrated in Figure 2, so that the exterior skeleton is the z-axis and the interior skeleton the unit circle in the xy-plane. We seek a line L through the unit circle, intercepting the z-axis at $z = \lambda_0$, such that L intersects the line g in a right angle, at a point with parameter value μ_0. The two unknowns λ_0 and μ_0 are found by solving

$$\begin{aligned} (a_0 + \mu_0 a_1)^2 + (b_0 + \mu_0 b_1)^2 - (c_0 + \mu_0 c_1 - \lambda_0)^2/\lambda_0^2 &= 0 \\ \mu_0 - c_1\lambda_0 + (a_0 a_1 + b_0 b_1 + c_0 c_1) &= 0 \end{aligned} \tag{10}$$

The first equation fixes the intersection with g and is quartic. The second equation states perpendicularity and is linear. We summarize the findings with

Theorem 3.3 Let g be a line or a cylinder. Determining the closest approach pairs with a cylinder or line requires solving a linear system, determining closest approach pairs with a cone requires solving a quadratic equation, and with a torus it requires solving a quartic equation.

3.2.4 Closest Approach Pairs with a Cone

Assume that f and g are both cones. We assume that one of the cones, say f, has the equation (3), and that the axis of g is given by the line (6). Moreover, the axes of the two cones are assumed to be skew. We identify a normal cone of f and a normal cone of g such that each cone contains the apex of the other. The normal cones to g can be expressed as a quadratic form $\hat{g}(x, y, z, \mu_0)$, where μ_0 specifies the apex position on the axis of g. Thus, we must solve a system with the variables λ_0 and μ_0 of the form

$$\begin{aligned} (a_0 + \mu_0 a_1)^2 + (b_0 + \mu_0 b_1)^2 - (c_0 + \mu_0 c_1 - \lambda_0)^2/u^2 &= 0 \\ \hat{g}(0, 0, \lambda_0, \mu_0) &= 0 \end{aligned} \tag{11}$$

The system consists of two quadratic equations.

If f is a torus, we assume a coordinate choice as before with the axis of rotation the z-axis and the interior skeleton of the torus the unit circle in the xy-plane. The axis of g is again assumed to be given by (6), and the normal cones by $\hat{g}(x, y, z, \mu_0)$. Here, we solve the system

$$\begin{aligned} (a_0 + \mu_0 a_1)^2 + (b_0 + \mu_0 b_1)^2 - (c_0 + \mu_0 c_1 - \lambda_0)^2/\lambda_0^2 &= 0 \\ \hat{g}(0, 0, \lambda_0, \mu_0) &= 0 \end{aligned} \tag{12}$$

Thus we must solve a quadratic and a quartic equation. Summarizing the situation, we have

Theorem 3.4 Let g be a cone. Then the closest approach pairs with a skew cone are found by solving a system with two quadratic equations, and the closest approach pairs with a torus are found by solving a system with a quadratic and a quartic equation.

3.2.5 Closest Approach Pairs with a Torus

Let f be a torus, where the coordinate system is aligned as before, and assume that g is another torus whose axis is specified by (6). We assume that the two axes are skew. We must find a line intercepting the axis of f at $z = \lambda_0$ and the axis of g at a point specified by μ_0 that also intersects the interior skeletons of g and of f. Such lines are determined by two cones of normals, one consisting of normals to f, the other of normals to g, where each cone contains the apex of the other cone. We recall that the normal cone to a torus has an implicit equation of degree 4 when the apex position is a variable, so that we must solve a system of two quartic equations.

Theorem 3.5 Let f and g be two tori. Then the closest approach pairs are determined by solving a system of two quartic equations.

3.3 Closest Approach Pairs Involving Curves

Edges of CSG objects have carriers that can be represented as the intersection of two surfaces g and h. The surfaces have degree one, two, or four. If p is a point on the carrier, it satisfies the implicit equations of g and of h. Moreover, the normal plane to the curve, at p, is spanned by the gradients to g and h, evaluated at p and denoted N_g and N_h, respectively. The tangent vector to the curve at p is given by the cross product of the normals, i.e., by $N_g \times N_h$.

In the following, we assume that g and h intersect transversally at p. That is, p must not be a singular curve point of $g \cap h$. We formulate the system with the implicit equations of g and h. If the intersection curve is parameterizable, for example when g is a plane and h is a quadric, the parametric form of the curve can be used instead, and doing so simplifies the systems by lowering the number of variables to be determined. The details are routine and we will not discuss this variant further.

3.3.1 The Curve/Surface Case

We explain how to find closest approach points between the curve $g \cap h$ and the surface f. If p is a point on f and q is on $g \cap h$, then they are a closest approach pair only if the connection line $\overline{pq}$ lies in the normal plane to $g \cap h$ at p.

If f is a plane, let $z = 0$ be its implicit equation. We seek a curve point p at which there is a linear combination of the surface normals that is perpendicular to f. Let $p = (x_0, y_0, z_0)$ be a point on the curve at which such a linear combination exists. Then p is found by solving

$$\begin{aligned} g(x_0, y_0, z_0) &= 0 \\ h(x_0, y_0, z_0) &= 0 \\ u_1 g_x + u_2 h_x &= 0 \\ u_1 g_y + u_2 h_y &= 0 \end{aligned} \tag{13}$$

The system has one degree of freedom accounted for by the fact that only the ratio of u_1 and u_2 is important. We could therefore proceed by solving the system (13) first with $u_1 = 1$, and then solving it again with $u_1 = 0$ and $u_2 = 1$. Note that the equations do not exceed the maximum degree of g and h.

If f is a sphere, let $q = (a, b, c)$ be its center. We need to find a curve point $p = (x_0, y_0, z_0)$ in whose normal plane we can find a line that contains q:

$$\begin{aligned} g(x_0, y_0, z_0) &= 0 \\ h(x_0, y_0, z_0) &= 0 \\ u_1 N_g + u_2 N_h &= (x_0 - a, y_0 - b, z_0 - c) \end{aligned} \tag{14}$$

The third equation states that a certain linear combination of the two normals is equal to the vector from the curve point to center of the sphere. It is equivalent to three scalar equations.

If f is a cylinder, assume its axis is given by the line (6). We seek a curve point $p = (x_0, y_0, z_0)$ whose normal plane contains a line intersecting the cylinder axis at right angle. We assume that the z-axis is the cylinder axis. Thus we have to solve the system

$$\begin{aligned} g(x_0, y_0, z_0) &= 0 \\ h(x_0, y_0, z_0) &= 0 \\ u_1 N_g + u_2 N_h &= (x_0, y_0, z_0 - \lambda_0) \\ (x_0, y_0, z_0 - \lambda_0) \cdot (a_1, b_1, c_1) &= 0 \end{aligned} \tag{15}$$

Here, $(0, 0, \lambda_0)$ is the intersection of the normal line with the cylinder axis. The third equation again corresponds to three scalar equations.

If f is a cone, we seek a normal line that lies on a cone of normals to g. Assuming that the cone g has the equation (3). We solve the following system

$$\begin{aligned} g(x_0, y_0, z_0) &= 0 \\ h(x_0, y_0, z_0) &= 0 \\ u_1 N_g + u_2 N_h &= (x_0, y_0, z_0 - \lambda_0) \\ x_0^2 + y_0^2 - (z_0 - \lambda_0)^2/u^2 &= 0 \end{aligned} \tag{16}$$

For the torus, finally, we assume a coordinate system as in Section 3.2.1. We seek a line normal to the curve that intersects the z-axis at $(0, 0, \lambda_0)$ and also intersects the unit circle in the xy-plane. This line has the direction $(x_0, y_0, z_0 - \lambda_0) = u_1 N_g + u_2 N_h$. We solve the following system:

$$\begin{aligned} g(x_0, y_0, z_0) &= 0 \\ h(x_0, y_0, z_0) &= 0 \\ u_1 N_g + u_2 N_h &= (x_0, y_0, z_0 - \lambda_0) \\ x_0^2 + y_0^2 - (z_0 - \lambda_0)^2/\lambda_0^2 &= 0 \end{aligned} \tag{17}$$

Note that the last equation expresses perpendicularity of the line to the torus.

In case the space curve is the intersection of two quadrics, we obtain systems whose nonlinear equations have degree 2, and, in the case of the torus, an additional quartic equation.

3.3.2 The Curve/Curve Case

Let $g_1 \cap h_1$ and $g_2 \cap h_2$ be the two curves on which we seek closest approach pairs (p, q). With $p = (x_1, y_1, z_1)$ and $q = (x_2, y_2, z_2)$, we find the points subject to the condition that each lies on the normal plane of the other, or, equivalently, that the connection line is in both normal planes.

$$\begin{aligned} g_1(x_1, y_1, z_1) &= 0 \\ h_1(x_1, y_1, z_1) &= 0 \\ g_2(x_2, y_2, z_2) &= 0 \\ h_2(x_2, y_2, z_2) &= 0 \\ u_1 N_{g_1} + u_2 N_{h_1} &= (x_1 - x_2, y_1 - y_2, z_1 - z_2) \\ v_1 N_{g_2} + v_2 N_{h_2} &= (x_1 - x_2, y_1 - y_2, z_1 - z_2) \end{aligned} \tag{18}$$

Note that the degrees of the equations do not exceed the degree of the surfaces involved.

4 Local Skeleton Analysis

Given a bisecting point, we develop a criterion to test whether the point is on the skeleton of the solid T, and if so, whether it is a vertex, on an edge, or on a face of the skeleton. Let p be a point of closest approach to the two elements E_1 and E_2 of the boundary of T. We determine the minimum distance of p to every other boundary element E_3. If p is closer to some E_3 than to E_1 and E_2, then p is not on the skeleton of T, otherwise it is.

As a result of the admissibility test, for each point of closest approach we find s distinct boundary elements of equal minimum distance. This is recorded as the structure

$$(p, r, (E_1, p_1), ..., (E_s, p_s))$$

where p is the point, r the minimum distance of p from the boundary of T, and, for $1 \leq k \leq s$, p_k is a footpoint of p and E_k is the boundary element on which p_k lies.

Now let $(p, r, (E_1, p_1), ..., (E_s, p_s))$ be a skeleton point so determined. Then p is on a face, edge, or vertex of the skeleton according to whether the Jacobian of the r-offset surfaces of $E_1, ..., E_s$, considered as intersecting hypersurfaces in R^n, has rank $n-2$, $n-1$, or n, assuming transversal intersection. Here, n is determined by the number of variables needed to formulate the equation, and is in the simplest case 4 [7]. So, the dimension of the tangent space at p of the r-offset intersection determines the topology of the skeleton in the neighborhood of p.

Suppose we are at a point on an edge or vertex of the skeleton, and wish to find the adjacent faces and edges of the skeleton not yet known. We proceed as follows to find the carriers of adjacent faces and edges:

1. Let $E_1, ..., E_s$ be the boundary elements whose r-offsets intersect at p and contain the footpoints of p. Let $n-k$ be the rank of the Jacobian, where $k=0$ or $k=1$.

2. Select all subsets of the E_k such that the Jacobian of the r-offsets, at p, has rank at most $n-k-1$ but is not smaller than $n-2$. The intersection of these r-offsets, for each subset, forms an adjacent Voronoi surface or curve and is one of the carriers we seek.

For each carrier so found, we then determine the edge or face that lies on it. This is done as follows.

If p is on an edge, we need to determine the direction in which the adjacent face lies. This is done by finding a point on either side of the edge and determining its distance from the boundary of T. If the distance is less than r, then the point is either not on the skeleton or else is on one of the faces we already know. Otherwise, the point is on a new face. If p is a vertex, that is, if the rank of the Jacobian is n, then the procedure for finding adjacent edges is analogous.

In this case, we do not find adjacent faces directly. Rather, we first locate the adjacent edges, and then locate adjacent faces from points on the edges.

5 Building a Skeleton Section

We assume we are given all points of closest approach between boundary element pairs, sorted by their minimum distance from the boundary. Let the corresponding distance sequence be $r_0, r_1, ..., r_m$. By a *section* of the skeleton we mean those skeleton points whose distance from the boundary is between r_{k-1} and r_k, where $1 \leq k \leq m$. There is one additional skeleton segment with points at distance greater than r_m. In step 3 of the algorithm, the skeleton is constructed section by section.

The first section is constructed beginning with all points at distance r_0. If the boundary of T contains locally convex edges, then these edges are the initial set and $r_0 = 0$. The skeleton faces and edges are now evaluated with the techniques of [6], in a four-dimensional (x, y, z, r)-space, where r is the distance of a skeleton point to the boundary of T. Using a first-order approximant, in 4-space, allows us to cut an approximant at the next critical distance r_1 if needed. Similarly, when constructing subsequent sections, we limit approximants to extend up to but not beyond the next critical distance.

By using the marching-cubes approach of [6], we can detect whether faces or edges are about to intersect. The intersection of skeleton faces can be done either by intersecting the defining equation systems, or by intersecting a face with its trimming surface, also specified as a system of equations [7].

When building a skeleton section, we thus construct vertices, edges and faces ordered by their distance from the boundary. At any time, several edges and faces have been partially evaluated, and these partial evaluations are the current *frontiers*. When reaching the next critical distance r_k, new frontiers are added to the list of current frontiers, namely those corresponding to the closest approach points at distance r_k. Eventually, each frontier closes. This happens when several frontiers come together without generating a new frontier. That is, the approximants at two or more frontiers join, and at their intersection the local approximants are not increasing in distance from the boundary, nor are there subsets of the nearest element set that have such approximants.

6 Summary

We have sketched a general algorithm for constructing the (interior) skeleton of three-dimensional solids. The algorithm first determines points of closest approach, sorts them by distance, and then constructs the skeleton by increasing distance. The restriction to CSG impacts fundamentally only the determination of closest approach points. In section 3 we show how this restriction can be exploited by deriving very simple algebraic systems that can be solved without expensive machinery. Given a good equation solver, however, the algorithm is easily extended to solids with more general boundaries.

The face and edge construction of the skeleton is done numerically, based on the generic techniques developed in [5, 6, 9]. The advantage is a uniform code that applies unchanged to all types of curved boundary elements, not only those arising in CSG. By constructing approximants in a four-dimensional space, moreover, we can determine robustly when the skeleton construction is locally completed, by monitoring the current distance from the boundary.

Acknowledgements

Helmut Pottmann and Kokichi Sugihara suggested a number of valuable improvements to the manuscript. Thanks also to Ching-Shoei Chiang, Pamela Vermeer, and Jianhua Zhou for their suggestions.

References

[1] H. Blum (1967), "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*, Weiant Whaten-Dunn, eds., MIT Press, Cambridge, MA, 362–380.

[2] H. Blum, R. Nagel (1978), "Shape description using weighted symmetric axis features," *Pattern Recognition 10*, 167–180.

[3] F.L. Bookstein (1979), "The line skeleton," *Comp. Graphics Image Proc. 11*, 123–137.

[4] B. Boots, A. Okabe, K. Sugihara (1991), *Spatial tesselations — concepts and applications of Voronoi diagrams*, John Wiley and Sons, Chichester, England, 1991.

[5] V. Chandru, D. Dutta, C. Hoffmann (1990), "Variable-radius blending with cyclides," in *Geometric Modeling for Product Engineering*, 39–58, K. Preiss, J. Turner, M. Wozny, eds., North Holland.

[6] J.-H. Chuang (1990), "Surface Approximations in Geometric Modeling," PhD Dissertation, Comp. Sci., Purdue University.

[7] D. Dutta and C. M. Hoffmann (1990), "A geometric investigation of the skeleton of CSG objects," *Proc. ASME Conf. Design Automation*, Chicago, IL.

[8] C. Hoffmann (1989), *Geometric and Solid Modeling, an Introduction*, Morgan Kaufmann, San Mateo, CA.

[9] C. Hoffmann (1990), "A dimensionality paradigm for surface interrogation," *CAGD*, to appear.

[10] C. Hoffmann (1990), "Algebraic and numerical techniques for offsets and blends," in *Computations of Curves and Surfaces*, M. Gasca, W. Dahmen, S. Micchelli, eds., Kluwer Academic Publ., 499–528.

[11] D.T. Lee (1982), "Medial axis transformation of a planar shape," *IEEE Trans. Pattern Anal. Mach. Intell., PAMI-4*, 363–369.

[12] U. Montanari (1969), "Continuous skeletons from digitized images," *JACM 16*, 534–549.

[13] L.R. Nackman (1982), "Curvature relations in three-dimensional symmetric axes," *Comp. Graphics Image Proc. 20*, 43–57.

[14] L.R. Nackman, V. Srinivasan (1989), "Bisectors of linearly separable sets," *Res. rept. RC14140*, IBM Yorktown Heights, NY.

[15] J. O'Rourke, N. Badler (1979), "Decomposition of three-dimensional objects into spheres," *IEEE Trans. Pattern Anal. Mach. Intell., PAMI-1*, 295–305.

[16] N.M. Patrikalakis (1989), "Shape interrogation," *Proc. 16th Annl. MIT Sea Grant College Program Lecture and Seminar*, Hemisphere Pbl., Cambridge, MA, 83–104.

[17] N.M. Patrikalakis, H. Gursoy (1989), "Shape interrogation by medial axis transform," *Design Lab. Memo. 90-2*, Sea Grant College Program, MIT.

[18] F.P. Preparata (1977), "The medial axis of a simple polygon," *Proc. 6th Symp. Math. Foundations of Comp. Sci.*, 443–450.

[19] F. Preparata and M. Shamos (1985), *Computational geometry*, Springer Verlag, New York.

[20] A. Requicha and H. Voelcker (1977), "Constructive solid geometry," *Tech. Memo 25*, Prod. Automation Proj., University of Rochester.

[21] J. Rossignac, A. Requicha (1984), "Constant-radius blending in solid modeling," *Comp. Mech. Engr. 3*, 65–73.

[22] J. Rossignac, A. Requicha (1984), "Offsetting operations in solid modeling," *CAGD 3*, 129–148.

[23] V. Srinivasan, L.R. Nackman (1987), "Voronoi diagram of multiply connected polygonal domains," *IBM J. Res. Dev. 31*, 373–381.

[24] V. Srinivasan, L.R. Nackman, S. Meshkat (1989), "Automatic mesh generation by symmetric partitioning of polygonal domains: I. algorithm," *Res. Rept.*, IBM Yorktown Heights, NY.

[25] S. Stifter (1990), "The Roider method: A method for static and dynamic collision detection." in *Issues in Robotics and Nonlinear Geometry*, C. Hoffmann, ed., JAI Press.