Efficient Computation of A Simplified Medial Axis

Mark Foskey
foskey@cs.unc.edu
Department of Radiology

Ming C. Lin
lin@cs.unc.edu
Department of Computer Science

Dinesh Manocha
dm@cs.unc.edu
Department of Computer Science

University of North Carolina at Chapel Hill
http://www.cs.unc.edu/gamma/smx/

ABSTRACT

Applications of of the medial axis have been limited because of its instability and algebraic complexity. In this paper, we use a simplification of the medial axis, the θ-SMA, that is parameterized by a separation angle (θ) formed by the vectors connecting a point on the medial axis to the closest points on the boundary. We present a formal characterization of the degree of simplification of the θ-SMA as a function of θ, and we quantify the degree to which the simplified medial axis retains the features of the original polyhedron.

We present a fast algorithm to compute an approximation of the θ-SMA. It is based on a spatial subdivision scheme, and uses fast computation of a distance field and its gradient using graphics hardware. The complexity of the algorithm varies based on the error threshold that is used, and is a linear function of the input size. We have applied this algorithm to approximate the SMA of models with tens or hundreds of thousands of triangles. Its running time varies from a few seconds, for a model consisting of hundreds of triangles, to minutes for highly complex models.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*; I.4 [**Image Processing and Computer Vision**]: Reconstruction, Image Representation

General Terms

Algorithms, Experimentation, Performance, Theory

Keywords

Distance field, Medial axis

1. INTRODUCTION

The medial axis [5] of a solid, defined as the set of centers of maximal balls contained in the solid, has been proposed as a tool for shape analysis, surface reconstruction, motion planning, and many other applications. It is useful because it provides a local lower-dimensional characterization of the solid. In particular, for a solid in 3D the medial axis consists of a union of surfaces that provide information about the shape and topology of the solid. If the distance to the boundary is also stored for each medial axis point, the resulting structure is known as the medial axis transform (MAT) and the entire boundary representation can be reconstructed from it.

The use of the medial axis has been limited mainly by two significant drawbacks: First, it is *unstable*, in that small deformations in the boundary of the solid can lead to large changes in the medial axis. Second, it is difficult to compute because of the underlying algebraic complexity. For a polyhedron, the surfaces constituting the medial axis are quadrics, and the seam curves can have degree four. For solids with curved boundaries, the medial axis sheets and seam curves can have much higher degree. Geometric computation with primitives of such high degree is hard to make both reliable and fast.

In this paper, we investigate a subset of the medial axis, which we call the *θ-simplified medial axis*, or θ-SMA. The θ refers to the angle formed by the vectors connecting a point on the medial axis to its corresponding closest points on the object boundary. We call this angle the *separation angle*, and the θ-SMA is simply the set of medial axis points for which the separation angle exceeds θ. The relationship between the stability of the medial axis and the separation angle has been known in the literature and used in many applications including surface reconstruction and skeleton-based modeling [1, 13, 26].

Main Results: We present novel properties of the θ-SMA and a fast algorithm to compute an approximation of the θ-SMA of a complex polyhedron. The θ-SMA, as indicated above, is parameterized by a minimum separation angle θ. It has the property that $M_{\theta_i} \subset M_{\theta_j}$ whenever $\theta_i > \theta_j$. Moreover, M_θ more closely approximates the medial axis as $\theta \to 0$, and becomes more stable as $\theta \to \pi$. We describe a formal characterization of the simplification of the medial axis as a function of θ. Given the distance function at each point on M_θ, an approximation to the boundary of the original solid can be reconstructed, and we give a formula relating the tightness of this approximation to θ.

We also present a novel and fast algorithm to compute an approximation to M_θ at an adjustable resolution ϵ. The ϵ determines the maximum error between the computed ap-

SM'03 June 16-20, 2003 Seattle, Washington, USA

proximation and θ-SMA. It is based on efficient computation of a distance field and its gradient using a spatial decomposition. The complexity of the resulting algorithm is $\Theta(n/\epsilon^3)$ where n is the number of primitives in the model and ϵ is the resolution (voxel width). We describe an adaptive subdivision scheme for computing a bounded-error approximation. Moreover, we present a number of techniques to improve the quality of the approximation by smoothing operations and accelerate the performance of the overall algorithm.

The algorithm has been implemented and applied to complex polyhedra composed of tens or hundreds of thousands of triangles. Its running time ranges from a few seconds, for a model composed of hundreds of triangles, to minutes for highly complex models, with measurements performed on a 2 GHz PC with an nVidia GeForce 4 graphics card.

As compared to other approximate schemes, our approach offers the following advantages:

- **Complex Models:** It can handle very large and complex models as the running time is a linear function of the input size.
- **Efficiency:** We use fast algorithms for computing the distance field and its gradient based on interpolation-based rasterization hardware. As a result, our algorithm can handle complex models composed of tens of thousands of polygons in a few minutes.
- **Approximation:** The ϵ-approximation to the θ-SMA is everywhere within $\sqrt{3}/2\ \epsilon$ of the medial axis, and it converges to the true θ-SMA as $\epsilon \to 0$.
- **Stability:** The criterion for simplification is scale-invariant, so that small shallow bulges are ignored, but thin, extended features in the medial axis are represented.
- **Simplification:** The simplification criterion is rather intuitive, depending only on the separation angle.

The rest of the paper is organized as follows. In Section 2 we give an overview of related work. In Section 3 we define M_θ and present some of its properties. In Section 4 we present our algorithm, and in Section 5 we analyze the time complexity of our algorithms and various sources of error in the approximation. We describe our implementation in Section 6 and highlight its performance on a number of complex models. In Section 7 we compare our approach to others in the literature, and we conclude in Section 8.

2. RELATED WORK

There is an extensive literature on both the computation and the simplification of the MAT and related constructions. In this section, we give a brief overview of exact and approximate algorithms for MAT computation as well as simplification.

2.1 Medial Axis Computation

At a broad level, algorithms for medial-axis computation can be classified into four categories: thinning algorithms, distance field based algorithms, algebraic methods, and surface-sampling approaches. These categories differ in terms of the underlying representations used for the medial-axis as well as how they compute it.

2.1.1 Thinning Algorithms

Thinning algorithms use a voxel-based representation of the initial figure, and perform erosion operations to arrive at a set of voxels approximating the medial axis. Lam et al. [19] give a survey of these approaches, and Zhang et al. [32] compare various methods. These methods are significant in the areas of image processing and pattern recognition, since the input data is represented as a discrete grid.

2.1.2 Distance Field Computations

Many approaches compute an approximation of the medial axis based on distance fields. Danielsson [12] uses a scanning approach in 2D to create an image in which each pixel contains the Euclidean distance to the nearest pixel on the boundary of the figure being analyzed. Moreover, the resulting distance map can be analyzed for local directional maxima to get an approximation of the medial axis. This algorithm has also been extended to three and higher dimensions [22].

Vleugels and Overmars [30] use a spatial subdivision to represent the medial axis, relying on nearest-neighbor queries to determine whether a cell must be further subdivided. They subdivide if the cell has vertices in different Voronoi regions and is larger than a certain threshold.

Hoff et al. [17] use interpolation-based graphics hardware to render a polygonal approximation of the distance field. The distance field is created in the depth buffer. We have extended this algorithm to compute the gradient of the distance field, also using rasterization hardware. We then use the gradient field for fast computation of the medial axis.

Siddiqi et al. [25] have also presented an approximate algorithm based on distance fields. Their analysis is based on a differential equation simulating the inward progress of a front starting at the boundary of the object. They compute a vector field that, at every point $\mathbf{p}$, is equal to the vector from the nearest point on the surface, to $\mathbf{p}$. They consider a point to be on the medial axis if the mean flux of the vector field, entering a neighborhood of the point, is positive.

2.1.3 Algebraic Methods

There is a family of methods that rely fundamentally on the fact that the algebraic form is explicitly known for each surface patch (i.e., each sheet) of the medial axis of a polyhedron.

Etzion and Rappoport [16] represent the curves and surfaces symbolically, but use a spatial subdivision to resolve the connectivity of the curves. They use algebraic tests to determine whether the surfaces pass into the cells of the subdivision, and subdivide until either the proper connectivity is determined, or a minimum cell size is reached. The presence of a minimum cell size means that it is not always possible to fully resolve the local connectivity of sheets and seams.

Most algorithms that represent the medial axis symbolically use a *tracing* approach [21]. Starting from a junction point on the medial axis, a seam emanating from the junction is followed. The seam terminates at another junction and the process is applied recursively. Using such an approach, Chiang [8] gives an algorithm for computing the medial axis of a planar region bounded by piecewise C^2 curves. Also, Dutta and Hoffmann [15] and Hoffmann [18] present an approach to compute the medial axes of constructive solid geometry (CSG) models.

Sherbrooke et al. [24], Reddy and Turkiyyah [23], and Culver et al. [11] have demonstrated tracing algorithms for polyhedra, all using different methods to find the endpoints of the seam curves. Culver et al. represent the medial axis exactly by means of systems of algebraic equations manipulated using rational arithmetic. Their method computes an exact representation of the medial axis provided there are no degeneracies (such as more than four seams intersecting at a point).

All of the methods in this family have been applied to polyhedra composed of only a few hundred faces. It is not clear whether they can be applied to complex models composed of tens or hundreds of thousands of faces. Either their running time is more than $O(n^2)$, where n is the number of faces, or these algorithms are susceptible to accuracy and robustness problems.

2.1.4 Surface Sampling Approaches

Surface sampling methods represent the initial figure as a dense cloud of sample points presumed to be on or near the boundary. The medial axis of the figure is approximated by a subset of the Voronoi diagram of the point cloud. Different algorithms based on this approach use different methods for selecting the desired subset of the Voronoi diagram. Many such variations have been proposed. Boissonnat [6] classified certain triangles of the Delaunay tetrahedralization of the point cloud as interior to the model; the Voronoi vertices dual to those tetrahedra approximate the medial axis.

Using a similar approach, Amenta et al. [1] construct an approximate, simplified medial axis which they use as a stage in a surface reconstruction from the original point cloud, a common application for this approach. Dey and Zhao [13, 14] also create a simplified surface model of a medial axis. Turkiyyah et al. [29] focus on improved accuracy rather than simplification. They follow the initial approximation with a numerical optimization step to move the sample points so that the Voronoi vertices are closer to the true medial axis. All of these authors give good surveys of this literature ([1, 13, 29]).

These algorithms have been applied to models composed of tens of thousands of points. One of the main issues when applying these algorithms to polyhedral models is in generating appropriate point samples on the boundary to ensure a tight approximation of the medial axis. In general, the worst-case running time of these algorithms can be $O(n^2)$, where n is the number of point samples. Recently, Attali and Boissonnat [2] have shown that the running time is only linear when the points are distributed on a fixed number of well-sampled facets. However, the point sampling of the surface has to satisfy certain criteria.

2.2 Medial Axis Simplification

A fundamental problem with the medial axis as a tool in shape analysis and surface reconstruction is that it is *unstable*, in the sense that small perturbations in the surface model lead to large changes in the structure of the medial axis. For a polyhedral model, every pair of adjacent faces produces a medial axis sheet extending to the edge connecting the two faces, producing a cluttered and uninformative medial axis. A number of methods for simplifying the medial axis have been proposed.

One of the criteria to identify parts of a medial axis that are stable is what we call the separation angle $S(\mathbf{x})$. It is the maximum angle formed by the vectors connecting the medial axis point $\mathbf{x}$ to its closest points on the boundary, and portions of the medial axis with a larger separation angle tend to be more stable. This has been noted by several researchers based on analyzing functions on the boundary surface [4], investigating the effect of noise [3] or samples [7] on the medial axis or in other skeleton-based applications [26]. Amenta et al. [1] use a similar criterion to determine whether a point on the medial axis is stable.

Dey and Zhao [13, 14] use a pair of criteria to retain faces from the Voronoi diagram of a set of points. For one criterion, they consider the angle between an approximate inward-pointing surface normal and a Delaunay edge (dual to a Voronoi face). If that angle is small, the Voronoi face is retained. The other criterion retains Voronoi faces if they are much farther from the surface sample points than the sample points are from each other.

Styner et al. [27] iteratively merge and prune sheets according to a pair of cost functions designed to minimize the change to the reconstructed model. They achieve a substantial reduction in medial axis complexity.

Choi and Seidel [10] study the stability of the medial axis and derive a bound on one measure of the instability of the medial axis for solids satisfying certain hypotheses.

3. θ-SIMPLIFIED MEDIAL AXIS

In this section we formally define the θ-SMA and give some of its properties. While the relationship between the separation angle and stability is well known, we are not aware of this particular subset of the medial axis being studied as an object in its own right. We show the degree to which it is more stable than the medial axis. Moreover, we define the θ-SMAT, which includes the distance information just as the MAT does, and show that the original model can be reconstructed from the θ-SMAT to an accuracy that depends in a simple way on θ. The significance of this relationship is that it is a way of quantifying the importance of the portion of the medial axis retained in the θ-SMA. If the original model can be reconstructed with reasonable accuracy, then one can argue that the most significant portions of the medial axis are being preserved.

Notation and Terminology: In this paper, vectors and points will be in boldface. Sets and functions will generally be denoted by capital letters. Unless otherwise specified, X will denote a solid with a polyhedral boundary.

Given a set of geometric primitives $S = \{P_i\}$, the *Voronoi region* of a primitive P_i is the set of points that are at least as close to P_i as to any other primitive. The collection of Voronoi regions is the *generalized Voronoi diagram*, or GVD. The medial axis of a polyhedron is a subset of the GVD of its faces, edges, and vertices.

We will say that an edge or vertex of X is *reflex* if its incident faces are not coplanar and it intersects the boundary of a ball whose interior lies in the interior of X.

Let X be a polyhedral solid with medial axis M. Recall that M can be characterized as the closure of the set of points in the interior of X having at least two nearest neighbors on the boundary of X. (Sometimes the requirement that the points be in the interior of X is relaxed.) Consider a point $\mathbf{x} \in M$, and let $NS(\mathbf{x})$ denote the set of its nearest neighboring points on the boundary of X. There is a sphere centered at $\mathbf{x}$ that does not cross the boundary

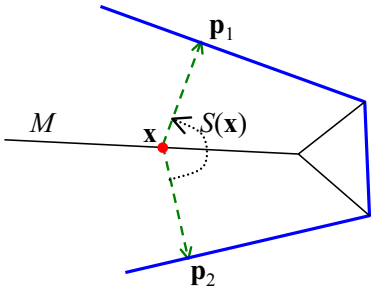


Figure 1: The separation angle $S(\mathbf{x})$ for a point on the medial axis. The thick border is the boundary of X.

of X, but that touches it at just the points of $NS(\mathbf{x})$.

For each pair of points $\mathbf{p}_1, \mathbf{p}_2 \in NS(\mathbf{x})$, we can consider the angle $\angle \mathbf{p}_1\mathbf{x}\mathbf{p}_2$. (We will treat all angles as values in $[0, \pi]$.) If $\mathbf{x}$ has more than two nearest neighbors, then we consider the largest angle subtended by a pair of nearest neighbors. We call this angle the *separation angle* $S(\mathbf{x})$ for the medial axis point $\mathbf{x}$:

$$S(\mathbf{x}) = \max_{\mathbf{p}_1, \mathbf{p}_2 \in NS_{\mathbf{x}}} (\angle \mathbf{p}_1\mathbf{x}\mathbf{p}_2)$$

(see Figure 1).

The intuitive motivation for this definition is as follows: If the separation angle is exactly π, then $\mathbf{x}$ is directly between its nearest neighbors, while if the angle is small, then both neighboring points are on the same side of $\mathbf{x}$, and there is space on the other side of $\mathbf{x}$ that is, in a natural sense, deeper in X.

Given an angle θ, define the *θ-simplified medial axis* M_θ of X to be the set of points of M with separation angle greater than θ. When we wish to emphasize the relationship of M_θ to a particular solid X, we will write $M_\theta(X)$. The following facts follow from the definitions of M and M_θ:

- The θ-SMAs are nested, with larger angles implying smaller subsets. That is, if $0 < \theta_1 < \theta_2 < \pi$, then $M_{\theta_2} \subset M_{\theta_1}$.
- $\overline{\bigcup_{\theta \in (0,\pi)} M_\theta} = M$, where $\overline{X}$ denotes the closure of X.

In this sense we can say that $M_\theta \to M$ as $\theta \to 0$. Note that, even though we specified X as a polyhedral solid, all of the above applies to any solid.

3.1 Quantifying the Significance of θ

In this section we derive a formula that quantifies the degree to which the θ-SMA retains the significant portions of the medial axis. It is well known that X can be reconstructed from M along with the radius values for each point on M. If we use M_θ instead of M in the reconstruction, then we get a subset of X, which we can call X_θ. The accuracy with which X_θ approximates X is a measure of the degree to which M_θ captures the important geometric features of X. Next, we formalize these notions, explaining what we mean by the accuracy of the approximation, and show how the accuracy is related to the separation angle θ, used as the angle cutoff in simplifying the original medial axis.

Formally, the *medial axis transform* is the *set* of all maximal balls contained in X. The centers of the balls constitute the medial axis, and retaining the balls is equivalent to retaining the radius information associated to each medial axis point. We define the *θ-simplified medial axis transform* (θ-SMAT) to be the subset of the MAT consisting of those balls centered on points of the θ-SMA. X can be reconstructed as the union of all the maximal balls in the MAT of X, and $X_\theta \subset X$ is the union of the balls in the θ-SMAT of X. By construction, M_θ is the medial axis of X_θ, but note that X_θ may not correspond to a polyhedron.

We can measure how closely X_θ approximates X in two ways. First, we can compare the volumes of the two spaces, computing the ratio $\mathrm{Vol}(X)/\mathrm{Vol}(X_\theta)$, where $\mathrm{Vol}(X)$ denotes the volume of X. Second, we can look at the distance between points on the boundary of X_θ and the nearest neighboring points on X. For each point $\mathbf{p}$ on the boundary of X_θ, there is a well-defined *local radius* $R(\mathbf{p})$ given by the radius of the smallest maximal ball touching $\mathbf{p}$ (see Figure 1). We can measure the local error as the distance from $\mathbf{p}$ to its nearest neighbor $\mathbf{p}'$ on the boundary of X, as compared to the local radius of $\mathbf{p}$. That is, the local error $E(\mathbf{p})$ is defined by

$$E(\mathbf{p}) = \frac{\|\mathbf{p} - \mathbf{p}'\|}{R(\mathbf{p})},$$

where $\mathbf{p}'$ is the point on the boundary of X that is nearest to $\mathbf{p}$.

The following theorem shows how well X_θ approximates X, as a function of θ.

THEOREM 1. *Let*

$$g(\theta) = \frac{1}{\sqrt{1 - \frac{4}{3}\sin^2\frac{\theta}{2}}}.$$

Then $\mathrm{Vol}(X)/\mathrm{Vol}(X_\theta) \le g(\theta)^3$ *and, for each point* $\mathbf{p}$ *on the boundary of* X_θ, $E(\mathbf{p}) \le g(\theta) - 1$.

PROOF. We claim that, if all the balls of the θ-SMAT are enlarged by a factor of $g(\theta)$, then their union will contain X. The largest local feature that can be excluded from X_θ is a corner such that the normals to the respective faces differ by an angle no greater than θ. If all the balls are enlarged by an appropriate ratio to include such corners, then their union will include all of X. We will argue that $g(\theta)$ as defined above is the required ratio by which all the balls of the θ-SMAT must be enlarged to include all of X.

First consider the 2D analogue (Figure 2). If there are two adjacent edges whose normals differ by an angle equal to the threshold angle θ, then no disk tangent only to those two edges will be added to the medial axis transform. Hence a disk such as the one shown will be the medial axis disk that is closest to the vertex $\mathbf{p}$, and the external radius r_{ext} is the radius to which that disk must be enlarged to contain all of that corner. Thus, in two dimensions, $g(\theta) = r_{\mathrm{ext}}/r = \sec(\theta/2)$.

In three dimensions, the corresponding situation consists of three faces coming together at $\mathbf{p}$ such that each pair of normals differs by θ. Let $\mathbf{x}$ be the center of the maximal ball closest to the extremal vertex $\mathbf{p}$, and let $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ be the points nearest to $\mathbf{x}$ on each of the three faces meeting at $\mathbf{p}$ (see Figure 3).

Consider the planes passing through $\mathbf{x}$, $\mathbf{p}$, and $\mathbf{p}_i$ for $i = 1$, 2, and 3. Since each pair of normals differs by the same angle, these planes must have dihedral angles of $2\pi/3$ to one

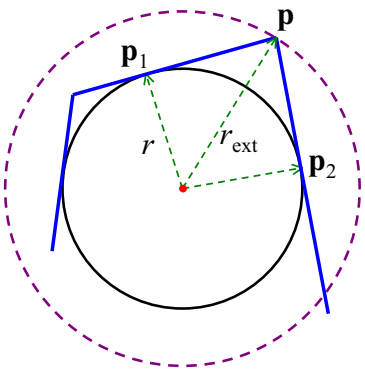


Figure 2: Computing the error bound. The angle subtended by $\mathbf{p}_1$ and $\mathbf{p}_2$ is equal to θ, so no circle tangent only to the edges containing $\mathbf{p}_1$ and $\mathbf{p}_2$ is represented in M_θ. If the solid circle is enlarged into the dashed circle, then the vertex $\mathbf{p}$ will be included. The radius r is equal to the local radius $R(\mathbf{p}_1) = R(\mathbf{p}_2)$.

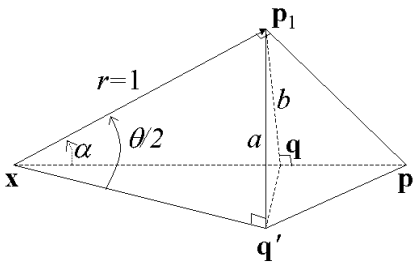


Figure 3: Computing $r_{\text{ext}} = \|\mathbf{p} - \mathbf{x}\|$. The vector $(\mathbf{p}_1 - \mathbf{x})$ is a normal vector from $\mathbf{x}$ to a face on the boundary of X. There are two other such faces; the endpoints of the normal vectors to those faces, $\mathbf{p}_2$ and $\mathbf{p}_3$, are not shown. The point $\mathbf{q}$ bisects $\overline{\mathbf{p}_1\mathbf{p}_2}$, which is one side of an equilateral triangle.

another, and the points $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ form an equilateral triangle in a plane orthogonal to $\overline{\mathbf{x}\mathbf{p}}$. Let $\mathbf{q}$ be the point where this plane crosses $\overline{\mathbf{x}\mathbf{p}}$, and let $\mathbf{q}'$ be the midpoint of $\overline{\mathbf{p}_1\mathbf{p}_2}$. Let $r_{\text{ext}} = \|\mathbf{p} - \mathbf{x}\|$, and $r = \|\mathbf{p}_1 - \mathbf{x}\|$. Then our goal is to compute the ratio r_{ext}/r. For convenience, assume without loss of generality that $r = 1$, so that we only need to compute r_{ext}.

If we denote $\angle\mathbf{p}_1\mathbf{x}\mathbf{p}$ by α, then $r_{\text{ext}} = 1/\cos\alpha$. We will compute $\sin\alpha$. Let $a = \|\mathbf{p}_1 - \mathbf{q}'\|$, and $b = \|\mathbf{p}_1 - \mathbf{q}\|$. Then $b = \sin\alpha$, and $a = \sin(\theta/2)$. (This is because $\angle\mathbf{p}_1\mathbf{x}\mathbf{p}_2 = \theta$.) Also, $a = b\sqrt{3}/2$ because $\overline{\mathbf{p}_1\mathbf{q}'}$ is perpendicular to $\overline{\mathbf{q}\mathbf{q}'}$, and $m\angle\mathbf{p}_1\mathbf{q}\mathbf{q}' = \pi/3$. Bear in mind that $\mathbf{q}$ is the center of the equilateral triangle $\triangle\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$. Thus,

$$\sin\alpha = b = \frac{2}{\sqrt{3}}a = \frac{2}{\sqrt{3}}\sin\theta/2.$$

Therefore,

$$r_{\text{ext}} = \frac{1}{\sqrt{1-\sin^2\alpha}} = \frac{1}{\sqrt{1-\frac{4}{3}\sin^2\frac{\theta}{2}}}$$

□

3.2 Stability and Connectivity

One of the benefits of the θ-SMA is that it is more stable than the medial axis. The medial axis of a finely tessellated polyhedron will have a sheet for every adjacent pair of faces, and many other pairs as well. The θ-SMA will only retain sheets for pairs of faces whose normals differ by an angle greater than θ, and thus, whose respective dihedral angles are less than $\pi - \theta$. Thus, introducing new vertices to generate a finer tessellation of the model will not create new sheets of the θ-SMA unless the new faces that are introduced to the polyhedron create sufficiently small angles with each other or with other faces in the model.

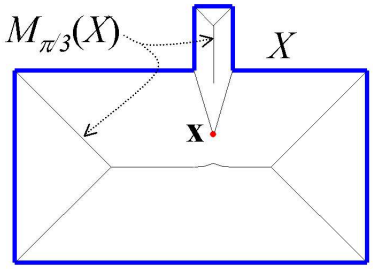


Figure 4: Disconnectedness. The point $\mathbf{x}$ is on the medial axis but has a small separation angle.

However, by design the θ-SMA detects small, elongated features, as in Figure 4. If such features are expected to arise as noise, then the θ-SMA will be affected by the noise. The relationship of θ to the stability of the simplified medial axis is illustrated by Figure 5.

The θ-SMA does not in general preserve the homotopy type of the model. It can be disconnected and have holes, even if X is simply connected. In Figure 4, $M_\theta(X)$ is shown for $\theta \approx \pi/3$. The point $\mathbf{x}$ is on the medial axis of the space X, but not on $M_{\pi/3}$ because its separation angle is too low. If $\mathbf{x}$ were to move towards the rectangular feature at the top, the separation angle would increase until it exceeded the threshold angle, at which point $\mathbf{x}$ would be on $M_{\pi/3}$.

This lack of a connectivity guarantee can be problematic for some applications. However, for others, what is desired is a characterization of the geometric properties of an object whose connectivity may already be understood. Also, it is possible to use a larger value of θ to select significant components of the object, and then compute again with a smaller value of θ to achieve improved connectivity. The more connected version can then be pruned, retaining just enough information to connect the components corresponding to the larger value of θ.

Finally, if it is desired to simplify X, one can remove small detached components of M_θ, yielding a pruned version of the θ-SMA. After that one can reconstruct an approximation to X from the pruned θ-SMA.

4. ALGORITHM

In this section we present a fast algorithm to approximate $M_\theta(X)$. The algorithm has two variations, one based on a uniform voxel grid, and the other on an adaptive subdivision of space. We first give an overview of the algorithm. We then describe in more detail the criterion we use to determine whether to add a face to the representation of M_θ, after which we describe the different spatial subdivision approaches. We conclude by describing two approaches to improving the surface representation of the θ-SMA.

The algorithm is based on a vector field that we call the

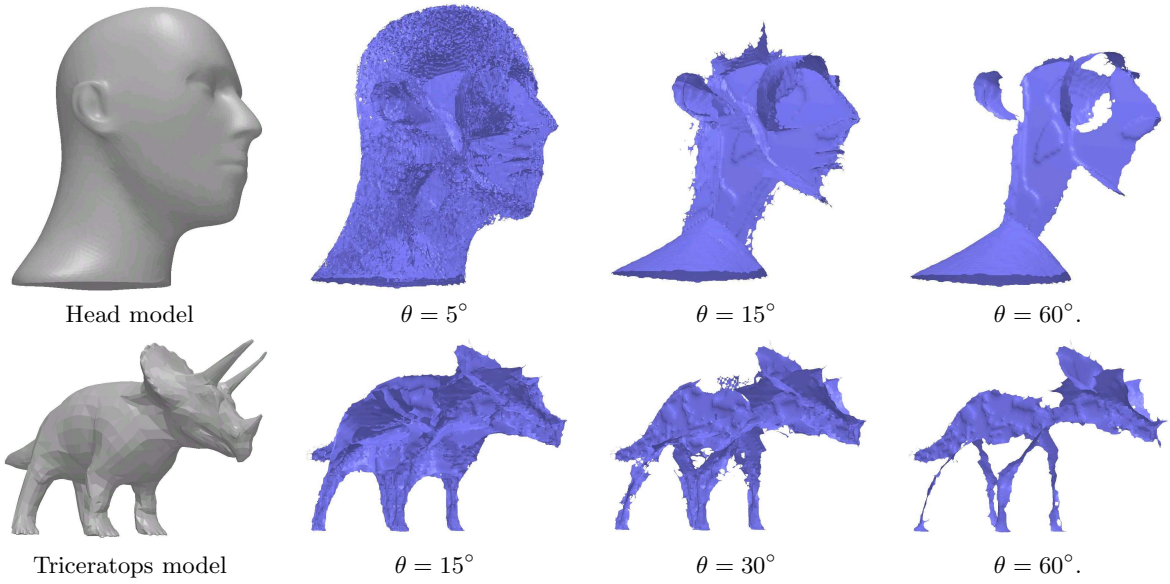


Figure 5: Different Θ-SMA for the same model. As the separation angle increases, the number of high frequency or sharp components decreases.

neighbor direction field of X, denoted N_X. If $\mathbf{x}$ is a point having a unique nearest neighbor $\mathbf{p}$ on the boundary of X, then

$$N_X(\mathbf{x}) = \frac{1}{\|\mathbf{p} - \mathbf{x}\|}(\mathbf{p} - \mathbf{x}).$$

This field consists of the negated gradients of the distance field defined by the boundary of X, and it is well-defined everywhere outside the boundary and medial axis of X.

Using N_X, we define a *separation criterion* to determine whether an arbitrary line segment in the interior of X crosses a sheet of the medial axis. The essence of the criterion is that two points $\mathbf{x}_1$ and $\mathbf{x}_2$ are taken to be on opposite sides of a medial axis sheet if $N_X(\mathbf{x}_1)$ and $N_X(\mathbf{x}_2)$ diverge. We use this criterion to test either the centers of the voxels of a uniform grid, or the cell vertices of an adaptive subdivision.

When a pair of points passes the separation criterion, we add a facet between them to our model of M_θ. Once the polygonal model is generated, it can be filtered to improve the fit of the represented sheets to those of the actual θ-SMA.

4.1 The Separation Criterion

For a given pair of points we first determine the angle between the respective direction vectors given by N_X. If the angle is not greater than the threshold θ then we reject the pair. However, if it is greater than the threshold, we need to be careful to avoid false positives. If, say, a reflex vertex is the nearest neighbor to both points in a pair, then both direction vectors will converge towards the vertex (see Figure 6). If the points are close enough to the vertex, then the angle between the vectors can be greater than the threshold, even though the segment between the points does not cross the medial axis.

To avoid this error, we need to check whether the vectors diverge. We check this condition by ensuring that the heads of the vectors are at least as far apart as the tails, where

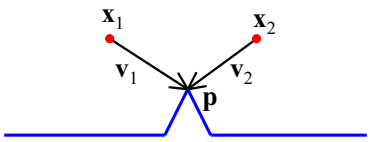


Figure 6: The direction vectors at neighboring voxels can differ by a large angle even when the voxels are not on different sides of the medial axis.

the lengths of the vectors are scaled to equal the separation between the neighboring points.

Given the separation criterion, we now present algorithms for two spatial subdivision schemes, a uniform grid and an adaptive grid.

4.2 Uniform Subdivision

The simplest spatial subdivision is a uniform grid. There are efficient ways to compute a distance field and its gradient that make use of the uniformity of the grid [17, 12]. We extend the algorithm presented in [17] for fast computation of the distance field.

Our goal is to create a uniform sampling of the direction field of the model X. We divide the volume into an axis-aligned voxel grid, referring to a set of voxels with a constant z-value as a *slice*. The algorithm we use relies on the parallel nature of interpolation-based graphics hardware to perform the computation efficiently for one slice at a time. The algorithm simultaneously computes a distance field and a direction field over a uniform 2D grid for each slice. We will describe the computation of the distance field first and then explain how we use it for direction computation. For each slice, the distance field is a scalar function $D_X : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. If we decompose X into sub-objects X_i, then D_X is determined by the lower envelope (or minima) of the set of all the distance functions D_{X_i}. We thus decompose X into its

faces, edges, and vertices and compute the lower envelope of the distance fields of each of these primitives.

The distance functions of these primitives can be represented in a simple form. We highlight these functions for points, lines and planes. For edges and triangular faces, these definitions are combined in piecewise fashion to represent the full distance field for the primitive. We describe formulas for the slice $z = 0$ and for primitives placed in particularly convenient configurations. The general forms can be derived by simple coordinate transformations.

For a point $\mathbf{p} = (0, 0, c)$ and the slice $z = 0$, the distance field is the hyperboloid

$$D_{\mathbf{p}}(x, y) = \sqrt{x^2 + y^2 + c^2}.$$

For a point with arbitrary coordinates we perform a translation on the distance field.

For the line L in the xz plane given parametrically by $(ta, 0, tc)$ with $a^2 + c^2 = 1$, the distance is given by the elliptical cone

$$D_L(x, y) = \sqrt{x^2c^2 + y^2}.$$

For a general line, we perform a translation and a rotation.

Finally, let F be the plane defined by the equation $ax + by + cz + d = 0$. If we assume that a, b, and c are chosen so that $a^2 + b^2 + c^2 = 1$, then the distance from a point to the plane is simply found by evaluating the left-hand side of the equation at that point. Thus, for the slice $z = 0$ we have

$$D_F(x, y) = ax + by + d.$$

In 3-space, the Voronoi region of the interior of a triangular face is defined by the three planes perpendicular to the face and passing through the edges. Points in this region are closer to the interior of the triangle than any of its edges or vertices. Similarly, the Voronoi region of the interior of a segment is defined by the planes normal to the segment and passing through the endpoints. For points outside the Voronoi region of the interior of a face or segment, we define the distance to be infinite by convention. Then, when the lower envelope of the distance fields for all the faces, edges and vertices is taken, the proper nearest neighbor will be determined for each point.

As we generate the distance field for each primitive, we also generate the direction field for that particular primitive. The distance field allows the lower envelope to be defined, and the lower envelope determines, for each point in the volume, which primitive defines the direction field at that point. With the point $\mathbf{p}$, the line L, and the plane F defined as above, unnormalized direction fields are given by

$$\begin{aligned} N_{\mathbf{p}}(x, y) &= (-x, -y, c) \\ N_L(x, y) &= (-xc^2, -y, xac) \\ N_F(x, y) &= -(ax + by + d)(a, b, c) \end{aligned}$$

To extract M_θ, we construct N_X for each slice by combining the direction fields for the primitives of X. We then evaluate each pair of voxels in the x, y, and z directions, adding a face to the approximate θ-SMA for each pair that passes the separation test. The computation of the distance field and the direction field maps very well to the rasterization hardware. More details are given in Section 6.

4.3 Adaptive Subdivision

Given the non-linear nature of the medial axis, in many applications it is possible to compute a better approximation by using a non-uniform grid. We present an algorithm based on octree subdivision of the space. This approach requires two primitive operations. First, one needs to evaluate the neighbor direction field $N_X(\mathbf{x})$ at an arbitrary point $\mathbf{x}$ in the volume of interest. Second, one must be able to determine whether an axis-aligned box contains, overlaps, or is contained by the object X. Both of these tests can be performed quickly using either a spatial subdivision to index the faces of the boundary of X, or by using a bounding volume hierarchy of X. There are standard collision detection packages that also provide the capability for distance queries, an example being PQP [20]. While these algorithms have been designed for object-object distance computation, it is straightforward to modify them to handle point-object computation. For instance, given a bounding volume hierarchy of the object, one can compute the feature on X that is closest to $\mathbf{x}$ by computing the distance from $\mathbf{x}$ to the bounding volumes at different levels in the hierarchy. Given the closest feature, the algorithm also computes the direction vector from $\mathbf{x}$.

Using these two primitive operations, the algorithm is as follows:

1. Begin with a single cell containing X.
2. Until a chosen cell size is reached, iteratively subdivide the cells that either
   (a) contain at least part of X but are not contained in X, or
   (b) are contained in X and have a pair of neighboring cell vertices that meet the separation criterion.
3. For each pair of vertices meeting the separation criterion, add a face to the medial axis as in the uniform grid approach.

This algorithm is more memory efficient, as well as more time efficient (in terms of operation count), than the the uniform grid algorithm. However, the uniform subdivision scheme is simpler to implement and maps well to the rasterization hardware.

4.4 Refining the Medial-Axis Approximation

The polyhedral approximation generated by the spatial subdivision schemes represents the θ-SMA up to a specified resolution. However, the sheets of the medial axis (which correspond to a portion of a quadric surface) are not well approximated by the axis-aligned facets of the voxel grid. In this section, we present two methods to refine the medial axis approximation.

Smoothing. When we use uniform subdivision of space, the algorithm we we use to compute the distance field produces distance values with a bounded error, with a bound equal to half the diagonal width of a voxel. For this reason, we cannot use the distance mesh to achieve subpixel accuracy in placing the faces of the medial axis mesh. However, we use the smoothing algorithm proposed by Taubin [28], a fast, non-shrinking smoothing filter. Because this filter is non-shrinking, it retains the shape of the medial axis sheets, while avoiding the stair-stepping appearance of the axis-aligned faces.

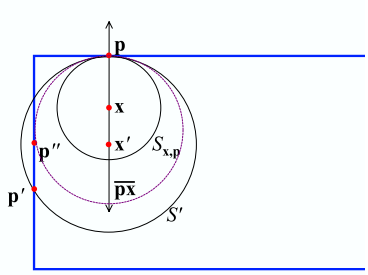


Figure 7: The first step in an iterative refinement of the approximate θ-SMA. $\mathbf{x}$ is the initial guess, and $\mathbf{p}$ is the nearest neighbor of $\mathbf{x}$ on the boundary. $S_{\mathbf{x},\mathbf{p}}$ is the center centered on $\mathbf{x}$ and passing through $\mathbf{p}$. There is a maximal circle $S_{\max}$ (not shown) that is contained in X and contains $S_{\mathbf{x},\mathbf{p}}$. We approach the medial axis by approaching $S_{\max}$. $S_{\max}$ touches the boundary in at least two places. $\mathbf{p}'$ and $\mathbf{p}''$ are successive approximations to to the second point where $S_{\max}$ touches the boundary of X.

Iterative Retraction. We have described an approach to perform distance queries (to the boundary) and direction computation at arbitrary points in the space. Based on this information, we can quickly find points that are very close to the medial axis. Let $\mathbf{x}$ be a point in the interior of X, but not on the medial axis. Let $\mathbf{p}$ be the unique point on the boundary of X nearest to $\mathbf{x}$, and let $\mathbf{p}'$ be any other point on the boundary of X. Then $\mathbf{p}'$ places an upper bound on how far $\mathbf{x}$ can be from the medial axis. Let $S_{\mathbf{x},\mathbf{p}}$ be the sphere centered at $\mathbf{x}$ and passing through $\mathbf{p}$, with radius $\|\mathbf{x}-\mathbf{p}\|$. Since $\mathbf{x}$ is not on the medial axis, $S_{\mathbf{x},\mathbf{p}}$ is not a maximal sphere. Let $S_{\max}$ be the maximal sphere containing $S_{\mathbf{x},\mathbf{p}}$. Then $S_{\max}$ exhibits the following properties:

- It will be tangent to $S_{\mathbf{x},\mathbf{p}}$, and hence centered on the line $\overline{\mathbf{px}}$ passing through $\mathbf{p}$ and $\mathbf{x}$.
- It will be no larger than the unique sphere S' centered on $\overline{\mathbf{px}}$ and passing through both $\mathbf{p}$ and $\mathbf{p}'$, because S' already touches two points on the boundary of X.

The center $\mathbf{x}'$ of S' is the most distant possible point from $\mathbf{x}$ where $\overline{\mathbf{px}}$ could cross the medial axis. See Figure 7.

In this way, points on the medial axis can be computed using an iterative algorithm. The algorithm proceeds in the following manner: Once S' is found, we can define $\mathbf{x}'$ to be its center, and choose a new point $\mathbf{p}''$ on the boundary of X as the boundary point nearest $\mathbf{x}'$. Using $\mathbf{p}''$ in place of $\mathbf{p}'$, we construct S'', and repeat the process. As we perform more iterations, a sequence of circles is constructed that approaches $S_{\max}$. This algorithm fits quite well with the adaptive subdivision approach as now we can compute vertices that are very close to the medial axis. Our method is similar to a method used by Wilmarth, Amato, and Stiller [31] to retract randomly generated sample points to the medial axis.

5. ANALYSIS

In this section, we analyze the performance of our algorithm. This includes the accuracy of our approximation as well as the running time.

5.1 Accuracy

In this section we show that the discrete approximation computed by our algorithm converges to the actual θ-SMA as the resolution becomes arbitrarily fine. As before, let X be a polyhedral subspace of $\mathbb{R}^3$, and let M be the medial axis of X. Let $\dot{X}$ denote the interior of X. For a given $\epsilon > 0$, let θ-simplified medial axis, $M_{\epsilon,\theta}$, be the approximation produced by our algorithm at the resolution ϵ.

The idea of our argument is that our algorithm estimates the set of points over which the neighborhood direction field N_X is discontinuous. The following theorem says that, inside X, the direction field can only be discontinuous at the medial axis. Since the direction field is not defined on the medial axis, it follows that, in $\dot{X}$, the medial axis is precisely the set of discontinuities of the direction field. We prove it based on the following theorem.

THEOREM 2. *The neighbor direction field N_X is continuous on the space $\dot{X} \setminus M$.*

PROOF. Let $\mathbf{x} \in \dot{X}$ be a point not on M. Either $\mathbf{x}$ is in a Voronoi cell of one of the faces, edges, or vertices of the boundary of X, or it is on the boundary between the Voronoi cells of a reflex edge and a face, or between the cells of a reflex vertex and a reflex edge. For each of these cases the distance field can be computed explicitly and is shown to be continuous in a neighborhood of $\mathbf{x}$. □

The result does hold for all curvilinear shapes of practical interest as well, but there are pathological cases where it fails. Choi, Choi, and Moon [9] give examples of such pathological cases in two dimensions, along with easily-satisfied criteria to ensure that a region does not exhibit such behavior.

THEOREM 3. *For a given $\epsilon > 0$, the ϵ,θ-SMA is within a Hausdorff distance of $\sqrt{3}\epsilon/2$ from a subset of the medial axis of X.*

PROOF. Let F be any face of $M_{\epsilon,\theta}$. F is a square face separating two voxels that satisfied the separation criterion. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the centers of those two voxels. Note that F is nowhere more than $\sqrt{3}\epsilon/2$ from the most distant point on the segment $\overline{\mathbf{x}_1\mathbf{x}_2}$, because each cubical voxel has side ϵ, and the distance from the center of a cube to the farthest point on its face is $\sqrt{3}/2$ times the side of the cube. We will show that the medial axis of X passes between $\mathbf{x}_1$ and $\mathbf{x}_2$. It follows that no point on F is farther than $\sqrt{3}\epsilon/2$ from some point on the medial axis.

Let $\mathbf{p}_i$ be the point on the boundary of X that is nearest $\mathbf{x}_i$, for $i = 1, 2$. The separation criterion ensures that the $\mathbf{p}_i$ are farther apart than the $\mathbf{x}_i$, which implies that the $\mathbf{p}_i$ are on different features (faces, edges, or vertices) of the polyhedral boundary of X. This inference follows by considering each type of feature in turn. Certainly the $\mathbf{p}_i$ cannot be on the same vertex. If the $\mathbf{p}_i$ are on the same edge, then the lines L_i containing $\mathbf{p}_i$ and $\mathbf{x}_i$ for each i are perpendicular to the edge. Thus the distance from $\mathbf{p}_1$ to $\mathbf{p}_2$ is the nearest distance from L_1 to L_2, and so $\mathbf{x}_1$ can be no closer to $\mathbf{x}_2$. The same reasoning applies to show that the $\mathbf{p}_i$ cannot be on the same face.

For each t with $0 \le t \le 1$, define $\mathbf{x}(t)$ to be $(1-t)\mathbf{x}_1 + t\mathbf{x}_2$, so that $\mathbf{x}(t)$ traverses the segment $\overline{\mathbf{x}_1\mathbf{x}_2}$. For each t, let $\mathbf{p}(t)$ be the nearest neighbor to $\mathbf{x}(t)$, if the neighbor is

unique. There cannot be a path traversed by $\mathbf{p}(t)$ from $\mathbf{p}_1$ to $\mathbf{p}_2$ that only crosses reflex vertices and edges, since the direction vectors converge towards such edges, and the vectors at the endpoints of the segment diverge. Hence, there must be an intervening convex edge or vertex, resulting in a discontinuity in the direction field. Therefore, $\overline{\mathbf{x}_1\mathbf{x}_2}$ crosses the medial axis, and hence F is entirely within the specified bound. □

We have shown $M_{\epsilon,\theta}$ is within a bounded distance of the medial axis of X. It remains to show that it actually converges to M_θ.

THEOREM 4. *The ϵ,θ-SMA converges to the θ-SMA in Hausdorff distance as $\epsilon \to 0$.*

PROOF. We need only consider the sheets, since the distance from the seams to the sheets is zero. Consider a point $\mathbf{x}$ on a sheet of M_θ. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the unit direction vectors to its two nearest neighbors. Recall that $S(\mathbf{x})$ denotes the separation angle for $\mathbf{x}$, that is, the angle between $\mathbf{v}_1$ and $\mathbf{v}_2$. Since $\mathbf{x}$ is on M_θ, $S(\mathbf{x}) > \theta$. Let $\eta = (S(\mathbf{x}) - \theta)/2$. By the continuity of N_X, there is a neighborhood B containing $\mathbf{x}$ such that the angle between $N_X(\mathbf{x}')$ and $\mathbf{v}_i$ is less than η for each $\mathbf{x}'$ in B and on the same side of the medial axis as v_i.

Now for sufficiently small ϵ, there will be adjacent voxels in the lattice of $M_{\epsilon,\theta}$ such that both voxels are contained in B, and the voxel centers are on opposite sides of the medial axis. Each such pair of voxels determines a face of $M_{\epsilon,\theta}$ that is contained in B. Since this applies for any sufficiently small neighborhood of $\mathbf{x}$, this shows that the minimum distance from $\mathbf{x}$ to $M_{\epsilon,\theta}$ can be made arbitrarily small. Combined with Theorem 3, this completes the proof. □

5.2 Time Complexity

In this section we discuss the time complexity of our algorithm. For the uniform grid approach, the analysis depends on the computational model that is used for the graphics hardware. If we assume that the hardware takes a a constant amount of time to render the distance field for each primitive, then the algorithm we use to compute the direction field requires time $\Theta(p/\epsilon)$ where ϵ is the resolution and p is the number of primitives (faces, edges, and vertices) in the model. Extracting the θ-SMA requires a single pass through the volume, requiring time proportional to $1/\epsilon^3$, so that the running time for the entire algorithm is $\Theta(p/\epsilon + 1/\epsilon^3)$. If we assume that rendering a slice of a distance field takes time proportional to the number of voxels in the slice, then the total time is $\Theta(p/\epsilon^3)$ or, equivalently, $\Theta(pv)$, where v is the number of voxels.

For the approach using an adaptive subdivision, the running time is highly output sensitive. We note that each distance query can require time logarithmic in the size of the model, using current techniques, but the constant factor is quite small.

6. IMPLEMENTATION AND RESULTS

In this section we describe the implementation of our algorithm and highlight its performance on a number of complex benchmarks.

Model	Tris	Resolution	T(N_X)	T(SMA)
Bent Torus	2,000	127x128x42	5.42	0.321
Buddha 1	15,536	55x128x55	35.7	5.5
Buddha 2	67,240	222x512x222	1634	48.8
Buddha 3	1,087,474	55x128x55	1588	1.17
Skel. Hand	654,666	79x106x127	602	0.07
Elbow Pipe	5,306	96x59x77	6.95	1.10
Elbow Pipe	5,306	128x79x103	10.8	3.87
Elbow Pipe	5,306	192x119x155	20.4	5.59
Elbow Pipe	5,306	256x159x207	33.1	8.24
Elbow Pipe	5,306	512x318x414	127	69.3
Bunny	69,451	64x63x50	77.4	0.12
Bunny	69,451	128x126x100	238	0.982
Bunny	69,451	256x253x200	794	2.51
Head	21,764	31x41x50	13.3	0.09
Head	21,764	79x106x127	57.8	0.22
Primer Anvil	4,340	128x73x112	8.99	0.61
Shell Charge	4,460	126x128x126	33.0	10.9

Table 1: Timings for some models at various resolutions. The Buddha model is shown at three different levels of detail. **Model**: Name of the model. **Tris**: Number of triangles in the model. **Resolution**: Number of voxels along each dimension. T($\mathbf{N_X}$): Time to compute the neighbor direction field. T(**SMA**): Time to extract the θ-SMA. All timings are in seconds on a 2Ghz Pentium 4 with an nVidia geForce 4 graphics card.

6.1 Implementation

We implemented the system in C++ using Microsoft Visual Studio, with OpenGL as our graphics API. Our implementation for computing the distance field is based on the techniques described in Hoff et al. [17]. In that approach, a volume is processed one slice at a time. For each slice, and each geometric primitive in the model, a surface, called a *distance mesh*, is rendered such that the depth buffer contains the shortest 3-space distance from each point in the slice to the given geometric primitive (which may not be in the given slice). If a pixel from a given primitive's distance mesh passes the depth test, then the pixel is in that primitive's Voronoi region.

We extend this method to acquire direction information as well, by encoding the directions to the nearest primitive in the red, green, and blue channels of the color buffer. Both the directions and distances are linearly interpolated across each triangle of the distance mesh, which is a source of error that grows with the size of the triangles. The distance mesh is designed to adjust the size of the triangles to keep the error within acceptable bounds.

We encode the gradient vector at each vertex of the distance mesh. It is important to keep in mind that each triangle is part of a distance mesh associated both to a geometric primitive and a slice of the volume. The slice corresponds to a z-value in the volume, but the z-coordinates of the rendered triangles correspond to distances from the slice to the primitive. The *colors*, likewise, correspond to directions from points *on the slice* to the primitive. Henceforth, when we refer to a given triangle of a distance mesh, we imply the projection of the triangle onto the specified slice.

Each direction vector is of unit length, with the x, y, and z components represented by the red, green, and blue color components respectively. As the components are interpolated across the triangle, the magnitude differs from unit

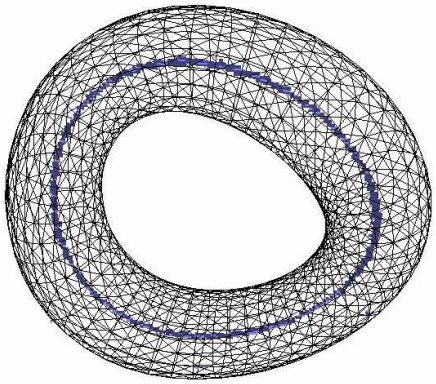
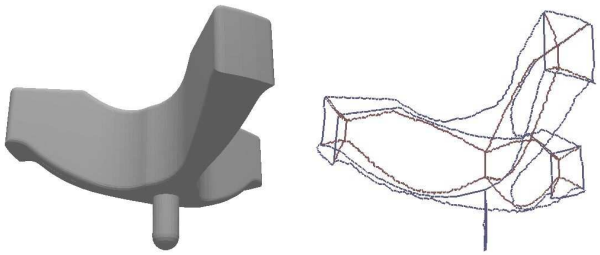


Figure 8: A torus and its θ-SMA. 2000 triangles. The grid resolution is 127x128x42.



a. **b.**

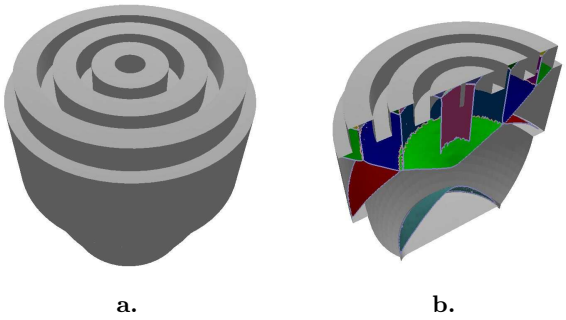
Figure 9: The "primer anvil" for a shotgun shell. 4,340 triangles, SMA computed at 128x73x112 resolution. (a) The model. (b) The θ-SMA. The seams and boundary curves of the θ-SMA are shown.

length, so that the vectors must be normalized after being read back and before computing a dot product to test the separation angle. The direction differs from the true direction. Consider a vertex which we can assume to be located at the origin, and a particular slice located at some depth z. Let $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ be the vertices of a mesh triangle that has been projected into the plane of the slice. Then a point in the triangle can be expressed in barycentric coordinates, that is, as a sum $\sum t_i \mathbf{p}_i$ with t_i chosen so that $\sum t_i = 1$. The true unit vector pointing towards the vertex, $\mathbf{v}$, and the estimated vector $\tilde{\mathbf{v}}$, are given by

$$\mathbf{v} = -\frac{\sum t_i \mathbf{p}_i}{\|\sum t_i \mathbf{p}_i\|}, \qquad \tilde{\mathbf{v}} = -\frac{\sum t_i \hat{\mathbf{p}}_i}{\|\sum t_i \hat{\mathbf{p}}_i\|},$$

where $\hat{\mathbf{p}} = \mathbf{p}/\|\mathbf{p}\|$. Then the error in the direction is given by the angle $\cos^{-1}(\mathbf{v} \cdot \tilde{\mathbf{v}})$. We do not have a good bound on this expression other than to say that it is bounded by the largest angle subtended by the triangle from the point of view of the primitive. We also note that, for triangles, which are treated as primitives separately from their edges and vertices, there is no interpolation error because the direction vector is constant. Thus, for a voxel inside a convex polyhedron, the only source of error in direction is the fact that each component of the vector can only be expressed with 8 bits of precision in the color buffer.

An alternative approach is to encode the full vector from each point on the slice to the given primitive, rather than a unit-length direction vector. This approach raised concerns with discretization error. However, with the advent of floating-point color buffers, that objection may not be a concern in the future.



a. **b.**

Figure 10: Shotgun shell "charge" with 4460 triangles. The grid resolution is 126x128x126. (a) The model. (b) Cross section, showing different sheets in different shades.

6.2 Performance

We applied our algorithm to polygonal models of various sizes, ranging from 2,000 triangles to more than 1 million. Some of the models were triangulations of scanned data, and others were CAD models. In general, scanned models have triangles with good aspect ratios and uniformly distributed over its boundary. However, many of these models have a high genus. On the other hand, the CAD models tend to have many sharp edges and uneven or high-aspect-ratio triangles.

In our tests (Table 6.1), the bulk of the computation time is taken by the computation of distance fields. Comparing running times for different resolutions shows an increase that is more than linear but less than cubic in the number of voxels along one dimension.

Except where otherwise specified, the separation angle $\theta = 60^\circ$, and the θ-SMAs have been smoothed. The resolution is specified in terms of the dimensions of the scene. The relative dimensions of the volume were determined by slightly enlarging a tight bounding box for the model.

7. COMPARISON WITH OTHER APPROACHES

There are by now a large array of approaches for computing as well as simplifying the medial axis. Performance comparisons between them are difficult, because they make different assumptions about the input, and generate different kinds of medial axis approximations as output.

The two main features of our approach are, first, that it computes the θ-SMA and not the entire medial axis, and, second, that we use a fast algorithm based on uniform spatial subdivision to compute the distance field and its gradient. As a result, we are able to compute good approximations of the θ-SMA for complex models in a few minutes.

Tracing algorithms [11, 23, 24] are much more time consuming and have only been applied to models consisting of a few hundred triangles. The adaptive subdivision algorithms of [30] and [16] computed the generalized Voronoi diagram,

a. b.

Figure 11: Buddha model with 1,087,474 triangles. The grid resolution is 55x128x55.

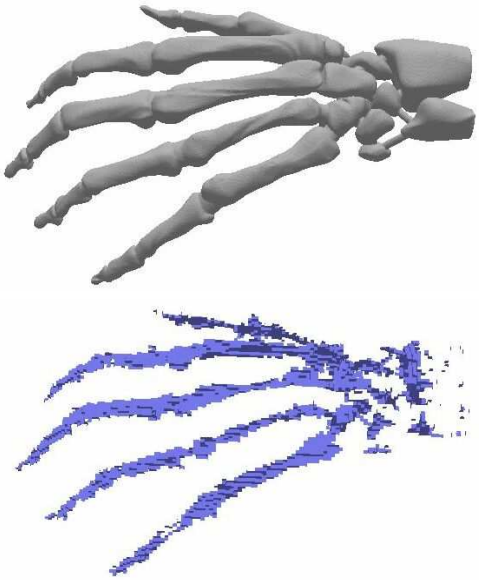


Figure 12: Skeleton hand with 654,666 triangles. The grid resolution is 79x106x127. No smoothing was performed.

rather than the medial axis. Their methods were only applied to models with up to a few hundred polygons. Our algorithm with adaptive subdivision is similar in approach to that of [30], but it creates a simplified medial axis rather than a generalized Voronoi diagram.

The surface sampling approaches such as those of [6], [1], and [13], take point samples on the surface as input, rather than the boundary features of a polyhedron. The accuracy and topology of the resulting medial axis varies considerably based on the sampling criterion used to generate the point samples. This makes it difficult to compare the approaches. Amenta et al. [1] report times of roughly six minutes for models of around 30,000 points, and Dey et al. [13] process around 122,000 points in a little over five minutes.

8. CONCLUSION

We have presented a medial axis approximation, the θ-SMA, based on the idea of the separation angle for a point on the medial axis. The criterion characterizing the θ-SMA is easy to understand and analyze, and it results in a more stable structure than Blum's medial axis. In practice, it is able to detect and capture most of the sharp features of the original model. We have presented a formal characterization of the simplification of θ-SMA as a function of θ.

We have described two algorithms for fast approximating the θ-SMA. One, using a uniform grid, is well-suited for implementation using the parallel features of modern graphics hardware. We have highlighted its performance on a number of complex benchmarks. The other algorithm uses an octree decomposition, in order to reduce the memory expense and make the time efficiency more output-dependent. Both algorithms fit into a consistent framework; both produce approximations that remain within a specified distance of some part of the full medial axis. We have analyzed the approximation errors produced by our algorithm.

There are a number of areas of future work. The key to our use of graphics hardware is that the direction vector field, stored as RGB triples, is associated to the scalar distance field, represented as depth values. This approach could be applied more generally to other pairs of associated vector and scalar fields. The use of graphics hardware for general computing purposes is currently an active area of research. We would like to compute the θ-SMA of solids with curved boundaries as well as procedural models. Moreover, we would like to use θ-SMA for different applications including mesh generation and shape analysis.

9. ACKNOWLEDGMENTS

We would like to thank Nancy Amato, the Stanford University Computer Graphics Laboratory, the Georgia Tech Large Models Archive, and NASA, for the use of polyhedral models. We also thank Kenny Hoff for his HAVOC software and for many helpful discussions, and we thank the reviewers for a number of suggestions on the exposition. Finally, this work has been supported in part by ARO Contract DAAD19-99-1-0162, NSF awards ACI 9876914, IIS-982167, ACI 0118743, ONR Contracts N00014-01-1-0067 and N00014-01-1-0496, NLM Contract N01-LM-0-3501, and by Microsoft Corporation and Intel Corporation.

10. REFERENCES

[1] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proc. ACM Solid Modeling*, pages 249–260, 2001.

[2] D. Attali and J. Boissonnat. A linear bound on the complexity of the Delaunay triangulation of points on polyhedral surfaces. In *Proc. ACM Solid Modeling*, pages 139–146, 2002.

[3] D. Attali and A. Montanvert. Computing and simplifying 2d and 3d continuous skeletons. *Computer Vision and Image Understanding*, 67(3):261–273, 1997.

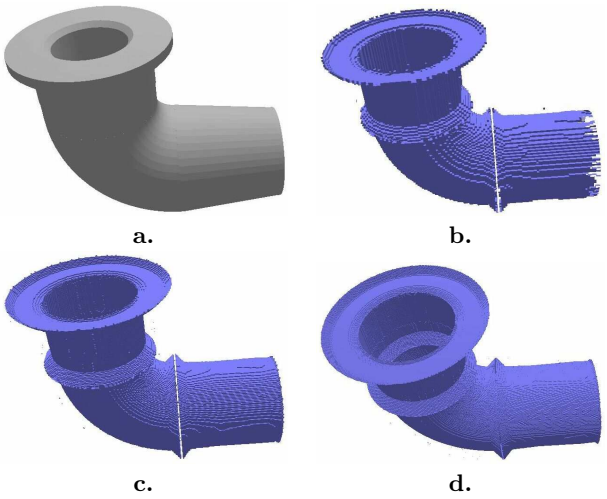


Figure 13: Elbow pipe, at varying resolutions. (a) The model. Figures (b), (c) and (d), correspond to 128, 256, and 512, voxels along the longest side, respectively. The gap visible in (b) and (c) shows where the interior of the pipe model is separated by a surface into two compartments. The gap is not visible in (d) only because the angle of the scene is slightly different. The θ-SMA is not smoothed.

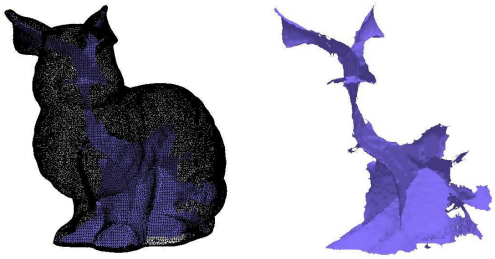


Figure 14: Bunny. 69,451 triangles, 128x126x100. (a) The bunny in wireframe, with the medial axis. (b) The θ-SMA.

[4] J. August, A. Tannenbaum, and S. Zucker. On the evolution of the skeleton. In *Proc. Int. Conf. on Computer Vision*, 1999.

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

[6] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.*, 3(4):266–286, 1984.

[7] J.-D. Boissonnat and F. Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *ACM Symp. Computational Geometry*, pages 223–232, 2000.

[8] C.-S. Chiang. *The Euclidean distance transform.* Ph.D. thesis, Dept. Comput. Sci., Purdue Univ., West Lafayette, IN, Aug. 1992. Report CSD-TR 92-050.

[9] H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical theory of medial axis transform. *Pacific J. Math.*, 181(1):56–88, 1997.

[10] S. W. Choi and H.-P. Seidel. Linear onesided stability of MAT for weakly injective 3D domain. In *Proc. ACM Solid Modeling*, pages 344–355, 2002.

[11] T. Culver, J. Keyser, and D. Manocha. Accurate computation of the medial axis of a polyhedron. In *Proc. ACM Solid Modeling*, pages 179–190, 1999.

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

[13] T. K. Dey and W. Zhao. Approximate medial axis as a Voronoi subcomplex. In *Proc. ACM Solid Modeling*, pages 356–366, 2002.

[14] T. K. Dey and W. Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. In *European Symp. Algorithms*, pages 387–398, 2002.

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

[16] M. Etzion and A. Rappoport. Computing the Voronoi diagram of a 3-D polyhedron by separate computation of its symbolic and geometric parts. In *Proc. ACM Solid Modeling*, pages 167–178, 1999.

[17] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manocha. Fast computation of generalized Voronoi diagrams using graphics hardware. *Proc. ACM SIGGRAPH*, pages 277–286, 1999.

[18] C. M. Hoffmann. How to construct the skeleton of CSG objects. In *Proc. 4th IMA Conf. on The Mathematics of Surfaces*, Bath, UK, 1990. Oxford University Press.

[19] L. Lam, S.-W. Lee, and C. Y. Chen. Thinning methodologies—A comprehensive survey. *IEEE Trans. PAMI*, 14(9):869–885, 1992.

[20] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha. Fast proximity queries with swept sphere volumes. Technical Report TR99-018, Dept. Comput. Sci., University of North Carolina, 1999.

[21] V. Milenkovic. Robust construction of the Voronoi diagram of a polyhedron. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 473–478, 1993.

[22] I. Ragnemalm. The euclidean distance transformation in arbitrary dimensions. *Pattern Recognition Letters*, 14:883–888, 1993.

[23] J. Reddy and G. Turkiyyah. Computation of 3D skeletons using a generalized Delaunay triangulation technique. *Computer-Aided Design*, 27:677–694, 1995.

[24] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. Computation of the medial axis transform of 3-D polyhedra. In *Proc. ACM Solid Modeling*, pages 187–199. ACM, 1995.

[25] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. The Hamilton-Jacobi skeleton. In *International Conf. Computer Vision*, pages 828–834, 1999.

[26] D. W. Storti, G. M. Turkiyyah, M. A. Ganter, C. T. Lim, and D. M. Stal. Skeleton-based modeling operations on solids. In *Proc. ACM Solid Modeling*, pages 141–154, 1997.

[27] M. Styner, G. Gerig, S. Joshi, and S. Pizer. Automatic and robust computation of 3D medial models incorporating object variability. *International Journal of Computer Vision*, to appear.

[28] G. Taubin. A signal processing approach to fair surface design. In *Proc. ACM SIGGRAPH*, pages 351–358, 1995.

[29] G. M. Turkiyyah, D. W. Storti, M. Ganter, H. Chen, and M. Vimawala. An accelerated triangulation method for computing the skeletons of free-form solid models. *Computer-Aided Design*, 29(1):5–19, Jan. 1997.

[30] J. Vleugels and M. Overmars. Approximating generalized Voronoi diagrams in any dimension. Technical Report UU-CS-1995-14, Dept. Comput. Sci., Utrecht University, 1995.

[31] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. Motion planning for a rigid body using random networks on the medial axis of the free space. *ACM Symp. Computational Geometry*, pages 173–180, 1999.

[32] Y. Y. Zhang and P. S. P. Wang. Analytical comparison of thinning algorithms. *Int. J. Pattern Recognit. Artif. Intell.*, 7:1227–1246, 1993.