Coarse-to-Fine Medial Surfaces

by

Svetlana Stolpner

Department of Computer Science
McGill University
Montréal

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Science

Your file Votre référence
ISBN: 978-0-494-32791-3
Our file Notre référence
ISBN: 978-0-494-32791-3

Abstract

We study the problem of computing the medial surface of a polyhedral solid. As this problem is difficult to solve exactly, we consider an approximate solution that locates regions of space through which the medial surface passes. The algorithm we design is based on the measure of the average outward flux of the gradient of the Euclidian distance transform of the polyhedron through the surface of a spherical region shrinking to a point. We extend previous methods by developing a completeness criterion that adopts this measure for spheres of non-zero size. As a result, we obtain a coarse-to-fine algorithm that reveals those voxels that contain sections of the medial surface. Such an implementation has the advantage that digital thinning with homotopy type preservation can be borrowed from past work. We argue that the error due to discrete flux computations is bounded.

Résumé

Nous étudions le problème du calcul de la surface médiane, ou du squelette 3D, d'un polyèdre. Puisque la solution exacte est difficile à obtenir, nous recherchons une solution qui localise les régions d'espace où la surface médiane passe. L'algorithme quel nous développons est fondé sur la mésure du flux moyen du gradient de la transformée du distance euclidienne à travers une région se rétrécissant vers un point. Nous améliorons les solutions existantes par un critère qui permet l'application de cette théorie dans les régions de grandeur non-nulle. En conséquence, on obtient un algorithme qui révèle, récursivement, les voxels qui contiennent la surface médiane. Cela nous permet d'utiliser l'algorithme d'amincissement numerique homotopique introduit ailleurs. On démontre que l'erreur due aux calculs discretes est bornée.

Acknowledgements

I would like to acknowledge the immeasurable amount of support I received from Sasha, who checked all the proofs and shared all my difficulties while creating this thesis. This thesis would not have been possible without the motivation, kind guidance, and expertise of Kaleem Siddiqi. I would like to extend gratitude to Sue Whitesides, who found time away from many important duties to help in the creation of this thesis, and also shared with me the experience of Barbados. Thanks to Mom, Dad and Nadia for having tolerated me being so far away and for continuing extending their love at a distance. I would also like to thank my colleagues and friends for making CIM such a positive place.

I am grateful to Sylvain Bouix and Pavel Dimitrov for having done such interesting work that I could build on in this thesis. I would like to thank Sylvain for providing me with his source code, Dinesh Manocha for making the PQP package available, and Nina Amenta for providing access to the PowerCrust code. This research was funded by FQRNT and NSERC.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Statement of Originality

Chapter 3, and Sections 4.1.1, 4.1.2, 4.1.4, and 4.3 are based on results published by the author in a joint article with Kaleem Siddiqi [51]. Of these, Sections 3.2, 4.2 and 4.3 are the original work of the author, Section 3.1 is a review of work appearing in [8, 18, 15], Section 4.1.1 surveys the results presented in [43], Section 4.1.2 surveys a number of sources, particularly [27], and Section 4.1.4 is based on the results of [8, 32].

Chapter 5 and Sections 4.1.3 and 4.2 are original unpublished work of the author.

All these sections were conceived and written with the inspiration and guidance of the author's supervisors Sue Whitesides and Kaleem Siddiqi.

Chapter 1

Introduction

We can see only what we know how to look for.

– U. Neisser

1.1 Seeing Shape

One of the most difficult problems in computer science is modelling human behaviour. We want, at the same time, to understand the extent to which computers can be useful in achieving this task, and to make models maximally effective. Particularly challenging is the task of modelling human visual perception. Our world is filled with visual information, the individual interpretation of which depends on the physical viewing apparatus, as well as on the conditioning of the viewer. "The real voyage of discovery consists not in seeking new landscapes but in having new eyes," wrote Marcel Proust [39]. The literal interpretation of this quote is equally appropriate as altering the functionality of our eyes or our visual perception will make for an entirely novel sensory and cognitive experience. Scientists have been searching for algorithms that model the way the brain processes visual information, although it is commonly believed that no two people experience the world visually in the same way.

The area of computer science concerned with the processing of visual information is computer vision. The input to a computer system represents the visual state of the world, and an algorithm is applied to interpret the input. Suppose the input to our computer system is a representation of a shape, which we consider to be the bounding contour (in 2D) or surface (in 3D) of an object. We wish to interpret this shape in a manner similar to how a human observer would. According to Ulric Neisser, one of the founders of cognitive psychology, "the cognitive structures crucial for vision are the anticipatory schemata that prepare the perceiver to accept certain kinds of information rather than others and thus control the activity of looking [36]." From this theory, it follows that there exist certain predefined procedures that a human carries out in the task of interpreting a shape. In the next section, we introduce one such possible procedure. The remainder of this thesis shall be concerned with the correct and efficient implementation of such a procedure using a computer.

1.2 Medial Representations

The physical world consists of a great number of naturally occurring (biological) shapes. Biological objects have a tendency to "flex, grow, or otherwise modify [10]". When a human observer interprets a shape in all of its modified forms via some cognitive process, he or she draws the conclusion that all derivative shapes describe the same shape-concept, and is able to classify shapes into categories and discriminate among shapes, despite the variability among shapes in the same category. For example, when presented with a variety of images of horses and cows, where some of the animals are lying down, standing and running, the human observer is easily capable of organizing the images into groups of cow and horse shapes, despite the enormous variability among the images in each category. It follows that in interpreting shape, humans "compute" some descriptor that is invariant to the natural kinds of modifications. Harry Blum proposed a description of shape based on the medial axis (surface) transform that captures the invariance to natural modifications that the human perceptual system is known to exhibit [9]. This shape descriptor has the intuitive interpretation of the set of reflective symmetries of the object.

Consider a closed, orientable object Ω in $\mathbb{R}^n$, with boundary B. We provide two equivalent definitions for the set of symmetries, or the medial surface, of Ω. Figure 3.1 provides a visual description of the grassfire flow. Figures 1.1 and 1.2 present examples of medial surfaces.

Definition 1.2.1. *Suppose the interior of our object Ω is made of dry grass. Then if we light its boundary B on fire instantaneously, the fire will propagate inwards in the direction normal to the boundary at a uniform rate. The locus of those locations where 2 or more fire fronts meet is the* ***medial surface*** *$\mathcal{MS}$ of Ω. The flow*[1] *generated by the advancing fire fronts is called the* ***grassfire flow*** *and is given by $W(0) = B$, $\frac{\partial W(t)}{\partial t} = \hat{\mathbf{N}}$, where W is the moving front, t is the time parameter, and $\hat{\mathbf{N}}$ is the normal to W.*

Definition 1.2.2. *The* ***medial surface*** *$\mathcal{MS}$ of Ω is the locus of centres of maximal inscribed balls in Ω.*[2]

The process of extracting the medial surface of Ω is reversible given that for each point of $\mathcal{MS}$ one notes the radius of its maximal ball. Ω can then be reconstructed as the union of balls having the necessary radii.

[1] A *flow* formalizes the idea of a time-dependent variable. We shall be studying vector flows, or flows generated by vector fields.

[2] A *maximal inscribed ball* in Ω is one that is completely contained inside Ω and which is not contained in any other such ball.

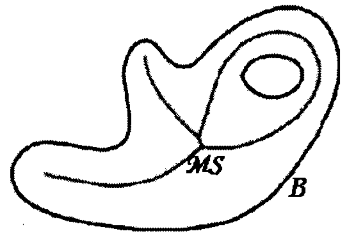


Figure 1.1: A 2D object with boundary shown in black and its medial surface (axis) in pink.

We shall be studying the medial surface of a 3D object $\Omega \in \mathbb{R}^3$. In this case, the (non-degenerate) medial surface is composed of 2D *sheets* meeting along 1D and 0D *junctions*. The 2D sheets are loci of points equidistant from 2 locations on the boundary, 1D junctions (or curves) are loci of points equidistant from 3 locations on the boundary and 0D junctions (or vertices) are loci of points equidistant from 4 locations on the boundary. Figure 1.2 illustrates all the different types of medial points. Rather than giving a formal definition of each type of medial point, we hope that the images are intuitive. This classification of medial points shall be important in the coming discussion. For certain *degenerate* objects, such as the cube and circular cylinder, there exist locations inside the objects that are equidistant from 6 (in the case of the cube) or an infinite number (in the case of the cylinder) of boundary locations. When we discuss degeneracy with respect to the computation of the medial surface, this is the kind of degeneracy we have in mind.

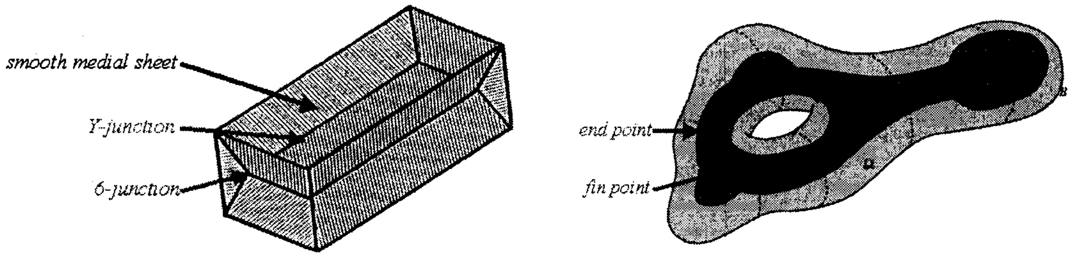


Figure 1.2: All the different types of medial points of the $\mathcal{MS}$ when $\Omega \in \mathbb{R}^3$. On the left is the medial surface of a box and on the right is the medial surface of a more complicated smooth object. Adopted from [14].

In the literature, the term medial *axis* is used instead of medial *surface* to refer to the medial locus of an object embedded in space of arbitrary dimensionality [47]. In this thesis, we shall be studying the case when Ω is a 3D object and thus, its medial locus is composed of (non-degenerate) surfaces. Our use of the term medial surface is consistent with that used in the work of [48] that we build on in this thesis.

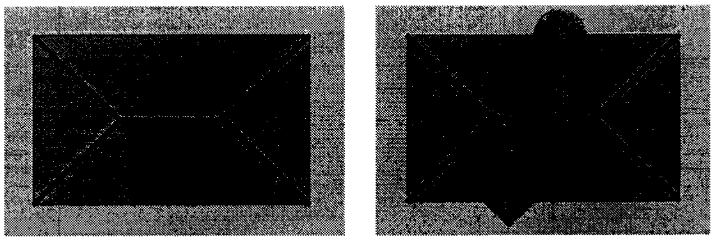


Figure 1.3: The effect of noise on the medial surface of a rectangle. The addition of 2 kinks on the right to the rectangle on the left introduces 2 new medial sheets.

The medial surface is highly sensitive to noise. A small kink in the boundary of the object can produce an additional medial sheet, as is illustrated in Figure 1.3. It thus seems apparent that not all sections of the medial surface are equivalent. A way in which points on the medial surface may be distinguished is by means of their *object angle*.

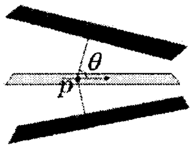


Figure 1.4: The object angle at a smooth medial point p is θ.

Definition 1.2.3. *For a particular point p on a sheet of $\mathcal{MS}$, the* ***object angle*** *θ is the angle made by the vector from p to any of its two closest points on B and the tangent plane to $\mathcal{MS}$ at p. Such a point p is called a* ***smooth medial point****.*

The concept of the object angle is illustrated in Figure 1.4. The idea that sections of medial surface with high object angle represent the most perceptually salient parts of an object's boundary is the basis of many techniques in the literature for removing "unwanted" medial surface sheets [1, 2, 23]. In fact, [23] shows that if we keep only those medial points whose object angle is above θ, then the ratio of the original volume of Ω and the volume of the object reconstructed from the simplified medial axis is at most

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

for the 3D case. The 2D case is studied by [18]. Such bounds provide the justification for the simplification of the medial surface based on object angle.

The medial surface $\mathcal{MS}$ is a lower dimensional representation of Ω. In fact, it has recently been shown that any open bounded subset of $\mathbb{R}^n$ has the same homotopy type as its medial surface, regardless of the smoothness of the boundary [31]. For $\Omega \in \mathbb{R}^3$, this means that there is a one-to-one corresponence between the number of connected components, cycles, holes, tunnels, and cavities of Ω and $\mathcal{MS}$ [31].[3] Although generally considered a simplification of Ω, in some pathological cases the medial surface may have infinitely many sheets [11]. However, for the planar case, if the planar boundary is composed of a finite number of real analytic pieces, then the medial surface has finitely many pieces [11]. Thus, we see that the medial surface is a simpler description of Ω that captures its topology.

We have seen that $\mathcal{MS}$ has a number of useful properties – it is of the same homotopy type as Ω and is a (generally simpler) lower-dimensional representation of Ω. We shall strive to capture these qualities of *homotopy equivalence* and *thinness* in our algorithm for computing an approximation to the medial surface.

So far we have been talking about the medial surface in a continuous setting.

[3] Figure 1.2 (left) shows a cavity and a tunnel.

In a discrete setting, the "cousin" of the medial surface is the Voronoi diagram.

Definition 1.2.4. *The* ***Voronoi diagram*** *of a set of points sites $s_i \in \mathbb{R}^n, 1 \leq i \leq N$, is a partitioning of space into n convex polytopes called* ***Voronoi cells*** *about each site s_i such that each polytope contains those regions of space that are closer to s_i than to any other point site. The dual of the Voronoi diagram is called the* ***Delaunay triangulation.***[4]

One approach for computing the medial surface of an object is to compute the Voronoi diagram of a set of points distributed on its boundary. The Voronoi vertices that lie inside the object are then used to approximate the medial surface. Refer to Figure 1.5. The convergence of the Voronoi vertices to the medial surface of a 3D solid is explored in [3].

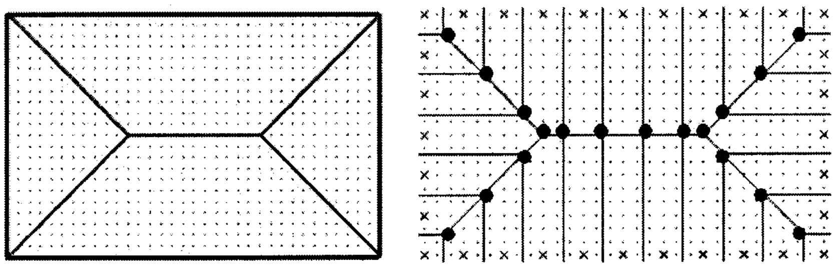


Figure 1.5: LEFT: The medial surface of a box. RIGHT: Voronoi diagram of points sampled on the box boundary. Voronoi vertices are shown as dark circles.

The *generalized* Voronoi diagram in 3D is a Voronoi diagram where sites can be faces, edges, or vertices. The generalized Voronoi diagram of a polyhedron is a superset of its medial surface. The difference between the two is that the generalized Voronoi diagram contains bisectors between a face and its edges and vertices as well as between faces meeting concavely (forming a reflex edge or vertex).

In computer graphics, an object is often represented by a triangulation of points on its boundary, a so-called *triangle mesh*. Such a mesh is a polyhedral approxi-

[4] In 3D, the Delaunay triangulation (tetrahedrization) is made by connecting those point sites that have a bisector in the Voronoi diagram with an edge.

mation of the object. The medial surface for a polyhedron is composed of bisectors between its faces, edges, and vertices, rather than surfaces of arbitrary algebraic degree for a smooth object. When Ω is a polyhedron, the medial sheets are quadratic surfaces, non-degenerate seams are intersections of 3 quadratic surfaces, and non-degenerate vertices are intersections of 4 such surfaces. Thus, the seams are solutions of equations of fourth degree and vertex junctions solutions of equations of eighth degree [33]. For this reason, even for a polyhedral object, the medial surface is difficult to compute exactly.

1.3 Problem Statement and Thesis Overview

> *The goal of this thesis is to propose an algorithm that computes the approximate location of the medial surface of a polyhedron. We seek an algorithm that may yield an arbitrarily fine approximation to the location of the medial surface and that is correct in detecting all regions of space through which the medial surface passes. The algorithm we shall design must be efficient in its usage of CPU cycles and storage. As well, the error introduced by discretizing the flux computations should be bounded.*

Chapter 2 surveys the existing approaches for computing the medial surface of 3D objects in a variety of boundary representations that are grouped into Voronoi methods, tracing methods, methods based on shocks of the grassfire flow, and spatial subdivision methods. It concludes with a summary of remaining challenges for the methods surveyed that have motivated the algorithm developed in this thesis.

Chapter 3, Section 1 studies the vector field induced by the grassfire flow. A relationship is established between the average outward flux of this vector field through a region shrinking to a point and the presence of medial sheets at that point. It turns out that the value of this average outward flux is proportional to the sine of the smallest object angle of the medial sheets, if any, meeting at this point. The constant of proportionality depends on the type of medial point we encounter and is zero in the case that no medial point is present. Thus, the value of the average outward flux about a point can be used to decide the presence of a medial sheet at that point. Section 2 extends the results of Section 1 on regions shrinking to points to the case where the region queried is a sphere of non-zero radius. This relationship is required to make the relationship with average outward flux computationally useful. We learn that in the case that the queried region is of non-zero size, deciding the presence of a medial sheet is not the same as in the case of zero-sized regions. Further, the relationship of the average outward flux through a sphere with object angle of the medial sheets passing through the sphere is not the same. For the case of an object with a polyhedral boundary, we devise a novel condition for deciding the presence of medial points in a region of non-zero size using a variant of the average outward flux measure. Further, we develop the relationship with object angle of the medial sheets passing through spheres of non-zero radius. These results now allow us to query voxels inside our object for presence of medial sheets by placing spheres about them and carrying out the appropriate computations. We conclude the chapter with a discussion of the effect of this approach on thickness and homotopy type of the set of voxels whose circumscribing spheres are said to contain medial sheets.

Having established a criterion for the detection of medial sheets in areas of non-zero size, we design a coarse-to-fine algorithm that "zooms in" on those regions of space that contain medial sheets in Chapter 4. We present a basic version of our algorithm and then discuss a number of implementation details. These implementation details are problems in their own right. Specifically, we look at the task of distributing points uniformly on a sphere, necessary for approximating the flux integral on spheres. Having shown that it is impossible to distribute many points on a sphere uniformly, we present a popular method for creating an approximately uniform distribution. Further, we study the problem of point-to-mesh distance, necessary for the computation of the vector field induced by the grassfire flow. Upon surveying various methods in the literature, we select one that seems appropriate. Next, we develop in detail some of the crucial steps of our basic algorithm. Finally, we discuss an approach that preserves homotopy type when deleting voxels from voxelized data sets. In addition, we present a number of suggestions for the acceleration of our algorithm and for a reduction in storage used. The chapter concludes with experimental results which demonstrate the output of our algorithm on a number of examples and provide a comparison with some state-of-the-art methods in the literature.

Chapter 5 studies the effect of the finite, non-uniform sphere sampling rate on the correctness of the algorithm. We argue that those sections of the medial surface that are missed are bounded in terms of one of three parameters: their object angle, the width of their generating boundary feature, or the amount of "dip" of the medial surface inside the sphere. We show that a more exact approximation, in terms of these three parameters, is achievable by increasing the sphere sampling rate.

Chapter 6 concludes the thesis with a summary of goals achieved and a list of directions for future work.

Chapter 2

Previous Work

The problem of computing the medial surface of a 3D object has been the subject of extensive research. Unlike its 2D counterpart, which has been studied for about three decades and is generally considered to be well-understood, this problem in three dimensions is still, in many ways, open. The reason for the added difficulty in the three dimensional case is the enormous computational burden of obtaining exact solutions, and the challenge of correctly capturing the relationships between "significant" medial sheets in approximate solutions.

The 3D object has been represented by a cloud of points sampled on its boundary, by a polygonal mesh, as a B-rep solid[1], as a set of internal voxels, and as one bounded by parametric surfaces. In examining the existing approaches, we consider the following division of methods that locate medial loci of such 3D objects: Voronoi methods, tracing methods, methods based on shocks of the grassfire flow, and recursive spatial subdivision methods. In the first section of this chapter, we provide a brief overview of the concepts, challenges and notable investigations for

[1] A *B-rep* is a representation of a solid's boundary using a collection of surface patches of various algebraic degrees that together preserve the Euler characteristic of the solid.

each of these existing approaches. In the second section, we briefly summarize the remaining challenges of these approaches and pose them as motivation for the work described in this thesis. This chapter reviews only a selection of methods that compute the medial surface and a more complete review appears in Part 2 of [47].

2.1 Existing Approaches

2.1.1 Voronoi Methods

Voronoi methods rely on the relationship between the medial surface of an object and the Voronoi diagram of a set of points sampled on the object boundary. While in 2D the set of Voronoi vertices for points sampled on a C^3 curve approximates the medial axis as the sampling density goes uniformly to infinity [44], this relationship is not true in 3D because of the possibility of 4 nearby sample points forming a very small Delaunay tetrahedron, called a *sliver* [16], whose circumcentre is a Voronoi vertex not part of the medial surface.

In [45], an algorithm is proposed for computing the medial surface for the class of those B-rep solids for which the maximal sphere can only be tangent to the surface at a finite number of points. The algorithm proceeds via a topology respecting Delaunay triangulation generated from sparsely sampled points on the solid boundary. Tentative medial points are classified into types by solving simultaneously a system of non-linear equations describing the medial point type using a modified Newton iteration method and considering if the system of equations admits a solution. The medial surface topology is assembled by searching among neighbouring tetrahedra in order to establish the connectivity between medial surface vertices.

The method of [2] approximates the medial surface by a triangulation of a subset of the vertices of the Voronoi diagram, called the poles, of points distributed appropriately on the surface of the solid. In [3], the authors show that this set of *poles*, or the furthest vertices in sample points' Voronoi cells, converges to the medial surface as the sampling density becomes infinite[2]. The intuition behind this is that for a dense enough point sample, the Voronoi cells are long and narrow and perpendicular to the surface. To decide which poles fall inside the shape, a graph on the power diagram[3] of the poles is traversed, and "inside" and "outside" labels are propagated. Several simplification techniques are proposed based on a measure of noise and redundancy. Research is carried out to prune sections of this medial surface to obtain a simpler reconstructed surface in a topology-preserving way by first segmenting the medial surface into sheets and then iteratively peeling them from the outside based on the measure of number of triangles and volume of generating boundary features [54].

When Ω is a 2D object, $\mathcal{MS}$ has very simple connectivity, and for an object with no holes, it is a topological tree. The pruning order for the medial surface of a 2D object is simply to recursively delete leaf branches, while preserving the topology (homotopy type), and is unique for non-degenerate configurations ([47], Chapter 6). However, when Ω is a 3D object, the Voronoi graph does not have the hierarchical organization that gives the pruning order for 2D objects, and usually contains vertices, the deletion of which from one layer makes another vertex from the same layer nondeletable [34]. Addressing this problem using the theory of cell complexes, [34] proposed a deletion sequence that processes layers of Delaunay tetrahedra for deletion in a topology preserving way, attempting to delete the

[2] Consider a sample point $w \in B$. Let $LFS(w)$ be the distance from w to the nearest point of the medial surface of B. Then $S \subset W$ is an $r - sample$ if the distance from any point $w \in B$ to its closest sample in S is at most $rLFS(w)$. When we say that a sample S has *infinite sampling density*, we mean that S is a $0 - sample$.[3]

[3] A *power diagram* is a weighed Voronoi diagram.

maximum number of tetrahedra satisfying a significance threshold. Finding an optimal pruning order for medial sheets of 3D objects remains an open problem [38].

Reference [16] proposes an alternative algorithm that approximates the medial surface directly from the Voronoi diagram of the set of points sampled on the object boundary in a way that is independent of scale and sampling density. The resulting Voronoi subcomplex is shown to converge to the true medial surface as the sampling density become infinitely fine. The authors construct the Delaunay triangulation and choose those faces whose dual Voronoi edges they want to keep. While alternative methods have suffered from the need to select pruning parameters individually for each data set, this method uses two fixed pruning parameters that produce satisfactory results for all models. The first parameter is the angle between the tangent plane at a sample point and Delaunay edges at that point, i.e. it is related to the object angle θ. The second parameter is the ratio of the length of the Delaunay edges and the circumradii of the surface triangles.

Although these methods are able to compute exact locations of medial vertices and may be efficient because of the availability of fast and robust Voronoi diagram algorithms, they produce many spurious branches that must be pruned. Further, the theoretical guarantees that these methods provide often require the sampling density on the object boundary to be infinite.

2.1.2 Tracing Methods

These methods start with known medial surface vertices and recursively find the seam curves using various approximate and exact techniques.

Reference [33] proposes an algorithm to construct the Voronoi diagram of a 3D polyhedron using approximate arithmetic. The tracing algorithm visits the vertices and edges of the Voronoi diagram in a breadth-first fashion. The algorithm considers if a set of boundary elements has a bisector inside a tetrahedron. By considering the quality of this approximate medial point, subsequent points are found by taking a step in the direction that improves this quality, found by solving a linear program. Voronoi edges (medial seams) are traced by fitting a sequence of tetrahedra whose size is adjusted using binary search. Tracing stops when a tetrahedron contains a bisector of a new site. Binary subdivision of space is proposed as a tool to speed up the search for closest sites for each tetrahedron.

Reference [41] computes the generalized Voronoi diagram of a 3D polyhedron by computing instead its dual, the abstract Delaunay triangulation, a simpler structure than the non-linear Voronoi diagram. Two boundary representations are considered: points sampled on the polyhedron's boundary and vertices, edges, and faces of the polyhedron. In order to make use of the first representation, one needs to be able to select a "well-chosen" set of points, something the authors don't know how to do *a priori*. The computation of the Delaunay triangulation proceeds incrementally from an initial Delaunay triangle, adding tetrahedra until the graph captures all the non-redundant half-spaces. However, the task of selecting a next boundary entity to build the next tetrahedron is computationally expensive and requires the solution of a system of non-linear equations. Moreover, the convergence of the resulting subset of the Voronoi diagram to the medial surface is questionable [47].

In [46], an algorithm for determining the medial surface of a general 3D polyhedral solid is proposed, with the possibility of allowing holes, tunnels, yet no cavities. This algorithm traces seam curves implicitly using a simple differential equation describing the evolution of the seams and thus avoids explicitly finding the seams' equations. Once a junction point is reached, the algorithm is applied recursively to determine feasible seams emanating from the junction point, and then traces them until a seam-endpoint is reached, or until another junction point is found, which is added to a priority queue. The result of the recursion is an adjacency graph representing the relationship of the elements of the medial surface, or its piecewise-linear approximation [13]. Sheets are detected by finding closed loops of vertices and edges.

Reference [13] presents a tracing algorithm that uses exact arithmetic to compute the seam curves accurately. Given that the vertex coordinates of the polyhedron are rational, the authors compute rational representations of the quadric surfaces making up the medial surface of the polyhedron. Starting from an initial seam curve equidistant from three boundary entities, one seeks the boundary element giving the next seam, recursing at junctions. The authors propose three algorithms to trace seams chosen according to the type of equidistant boundary elements to the seam. Another contribution is the possibility of efficiently finding the intersection of three quadric surfaces making a junction point. The algorithm is able to compute the medial surface for polyhedra having several hundred faces. At present, this method is limited by sensitivity to degeneracies.

Because tracing is done globally, these methods are especially sensitive to degeneracies and starting conditions. Compared to other methods, these methods are able to detect provably accurate locations of medial vertices. As a trade-off, they seem to be quite computationally expensive.

2.1.3 Methods Based on Shocks of the Grassfire Flow

Methods in this group consider the medial surface as the locus of shocks of the grassfire flow (recall Definition 1.2.1) on the object boundary[4]. These methods locate sources and sinks of the grassfire flow, apply the differential equation giving

[4] A *shock* of a flow is a location (in time and space) where the flow intersects itself.

rise to it, and perform analysis of the resulting vector field.

References [29, 30] compute a medial hypergraph of a boundary represented as a point cloud. The 1D summary of this graph, called "the shock scaffold," contains nodes corresponding to special medial points classified according to the number of points of contact on the boundary of their respective maximal spheres and the type of this contact. The nodes are found using exact bisector computations between point clusters. Numerous suggestions are presented for reducing the computational complexity of these computations. Nodes, links, and sheets are classified according to the rate of change of their radius function. Starting from initial medial nodes, subsequent junctions are found using the classification of nodes, and the search continues in the direction of increasing maximal sphere radius.

Reference [19] computes the medial surface of a shape bounded by a polygonal mesh by solving a diffusion PDE along the time axis. The boundary is evolved inwards in the direction of the surface normal and smoothed by a Laplacian operator. Medial surface points are selected whenever the curvature of the evolving boundary exceeds a given threshold. Those parts of the evolving front that collide with each other are considered part of the medial surface. In the numerical simulation, partial derivatives are discretized using umbrella operators. Users can freeze chosen points on the evolving boundary as medial points. This technique may not be as effective for irregular meshes.

Reference [17, 48] study the shocks that arise when the boundary of an object is evolved according to the grassfire flow equation. It turns out that the limiting value of the average outward flux of the vector field generated by this flow is non-negative only in those areas of space containing medial points. Reference [17] considers a discretization of this notion to a square lattice for computing the medial axis of a 2D shape. Reference [48] extends this work to a 3D voxelized object using approximate distances and coarse sampling. Both approaches lack theoretical guarantees on the correctness of the results. We examine this work in more detail in the next chapter.

Methods based on shocks of the grassfire flow are natural alternatives that capture the physical nature of the medial locus in relation to the original object. However, their applicability for fine-resolution solutions to three-dimensional problems is limited by the computational complexity and limited precision of the solutions.

2.1.4 Spatial Subdivision Methods

This group of methods locates areas of space, or cells, that contain sections of the medial surface. The criterion for finding such cells is applied recursively until the desired resolution is reached. These methods differ in terms of the guarantees made about the relationship between the cells found and the true medial surface, in terms of the boundary representation considered, and their ability to preserve homotopy type.

Reference [21] uses space subdivision to find the adjacency relationships between components of the generalized Voronoi diagram of 3D polyhedra. Subdivision continues until all points in the cell share the same four or less nearest elements (vertex, edge or face) on the object boundary, with the interpretation that each such cell contains one Voronoi vertex. This part of the computation produces an approximate Voronoi graph containing symbolic information. Separately, geometric information about the location of Voronoi vertices is obtained by solving locally univariate fourth-order polynomial equations.

Reference [55] uses spatial subdivision to approximate the generalized Voronoi diagram of a set of disjoint convex sites in any dimension. This approach avoids the difficulties involved with finding exact solutions of high-degree equations by only requiring distance computations. The cells kept by the algorithm are those whose vertices have different closest points on the convex sites and are proven to intersect the Voronoi diagram. Because only a finite number of locations on the cells are considered, it is possible that certain sections of the Voronoi diagram have been missed. Though this method does not directly apply to 3D polyhedra, it clearly illustrates the issues involved in spatial subdivision techniques.

Reference [6] constructs a polyhedral approximation of the generalized Voronoi diagram for sites of various shapes under different metrics using an octree data structure. The novelty of this method is its use of a propagation strategy to allow fast computation of distances between cells and sites. Subdivision continues until cells have closest boundary elements that are adjacent in the polyhedron.

In a similar spirit, reference [23] uses spatial sampling to approximate the medial surface sections of a polyhedron having a chosen object angle with axis-aligned facets. The necessary gradient of the distance field is computed quickly using interpolation-based graphics hardware in the version of the algorithm that uses uniform spatial sampling. The authors also propose the use of octree-like spatial subdivision, in theory only, as the necessary computations do not support the use of fast distance extraction. Reference [52] extends this method to allow for preservation of homotopy type during simplification. The authors demonstrate bounds on the object angle given the size of the cells in the subdivision. The simplification algorithm proceeds by iteratively removing frontier sheets with small object angle if their removal does not alter the homotopy type of the medial surface.

In general, the spatial subdivision methods that perform distance computations between the cells and the boundary at a small number of locations are limited in that they cannot guarantee completeness in detecting all sections of the medial surface at any resolution. When coupled with exact computations, like in [21], these methods may be able to locate the complete medial surface, though at a higher computational cost.

2.2 Remaining Challenges

It is a significant challenge to combine efficiency and correctness considerations in designing an algorithm to compute medial surfaces. Furthermore, exact methods (the Voronoi and tracing methods in particular) are burdened by the extreme sensitivity of the exact medial surface to noise and degeneracies. On the other hand, methods that seek to approximate the medial surface (the methods based on shocks of the grassfire flow and the spatial subdivision ones above) seldom can guarantee completeness and lack stopping conditions.

In this thesis we seek an efficient algorithm to find the arbitrarily precise discrete locations of space where the medial surface passes. We assume the input to be a closed, orientable polyhedron that approximates the boundary of a 3D object. Our method extends the work of [17, 48] by providing a completeness criterion that allows us to decide the presence of a medial sheet at any resolution for objects having a mesh boundary. The method combines aspects of techniques based on shocks of the grassfire flow and spatial subdivision approaches and is designed to have the following properties: 1) it is robust against noise or degeneracies, 2) it provides guarantees on completeness and 3) the errors incurred due to discretization are bounded.

In the following chapter we review the relationship with shocks of the grassfire flow and the average outward flux through a region shrinking to a point. We find that this measure of average outward flux is related to the object angle of the medial sheets, if any, passing through the shrinking region. Having established this useful relationship, we explore the challenges of using the same criterion for regions of non-zero size. For the case of an object with a polyhedral boundary, we are able to develop a completeness criterion that allows to adopt the average outward flux condition for deciding the presence of medial sheets in regions of non-zero size.

Chapter 3

Flux-Based Medial Surfaces

In this chapter we review the connection between shocks of a grassfire flow and the average outward flux of the gradient of the Euclidean distance function through the surface of a shrinking region, which is the basis of the skeletonization algorithm introduced in [49, 17] and further developed in [48]. We then use the results of [14] to provide the relationship between the average outward flux values and the type and significance of the medial sections, if any, passing through that point. In order to make this relationship computationally useful, we would like to adopt it for regions of non-zero size, which is a non-trivial and neglected problem. In Section 3.2, we present original results of a completeness criterion for detecting the presence of medial sheets passing through spheres of non-zero size for objects with a polyhedral boundary. We show bounds on the relationship of the average outward flux through a sphere with the object angle (significance) of the medial sheets, if any, passing through the sphere and the size of the sphere. Finally, we suggest using a modified average outward flux measure to detect voxels of space containing medial sheets and discuss the homotopy type and thickness of the resulting voxelized approximation to the medial surface.

3.1 From Flow to Points

3.1.1 Average Outward Flux

Recall Definition 1.2.1 of the medial surface via the grassfire flow. When W is the advancing front, t the time parameter, $\hat{\mathrm{N}}$ the normal to W, and B the object boundary, the mathematical equation of the grassfire flow is

$$\frac{\partial W(t)}{\partial t} = \hat{\mathrm{N}}$$
$$W(0) = B.$$

As the fire fronts advance inside the shape, those locations where the fire fronts meet are the medial surface. If we note for each location inside the shape the time it takes for the first fire front to get there, then the local maxima of these values correspond to locations where the medial surface passes. Equivalently, let us consider the signed Euclidean distance transform of our surface B.

Definition 3.1.1. *The **signed Euclidean distance transform** $D : \mathbb{R}^3 \rightarrow \mathbb{R}$ for a shape with surface B is given by $D_B(p) = \chi(p) \inf_{q \in B} d(p,q)$, where $d(.,.)$ denotes the Euclidean distance, and $\chi(p) = -1$ if p is outside or on B and 1 if p is inside B.*

For an example of the signed Euclidean distance transform of a shape, see Figure 3.1, taken from [18].

The fire front advances in the direction normal to the boundary at all times, and the distance from a point inside a shape to the closest point on the shape boundary is along a perpendicular to the shape boundary. Therefore, at each point inside the shape, the distance to its closest point on the boundary is the same as the time it takes for the first fire front to arrive at that point. As the two concepts are equivalent, the medial surface corresponds to the local maxima (or ridges) in the signed

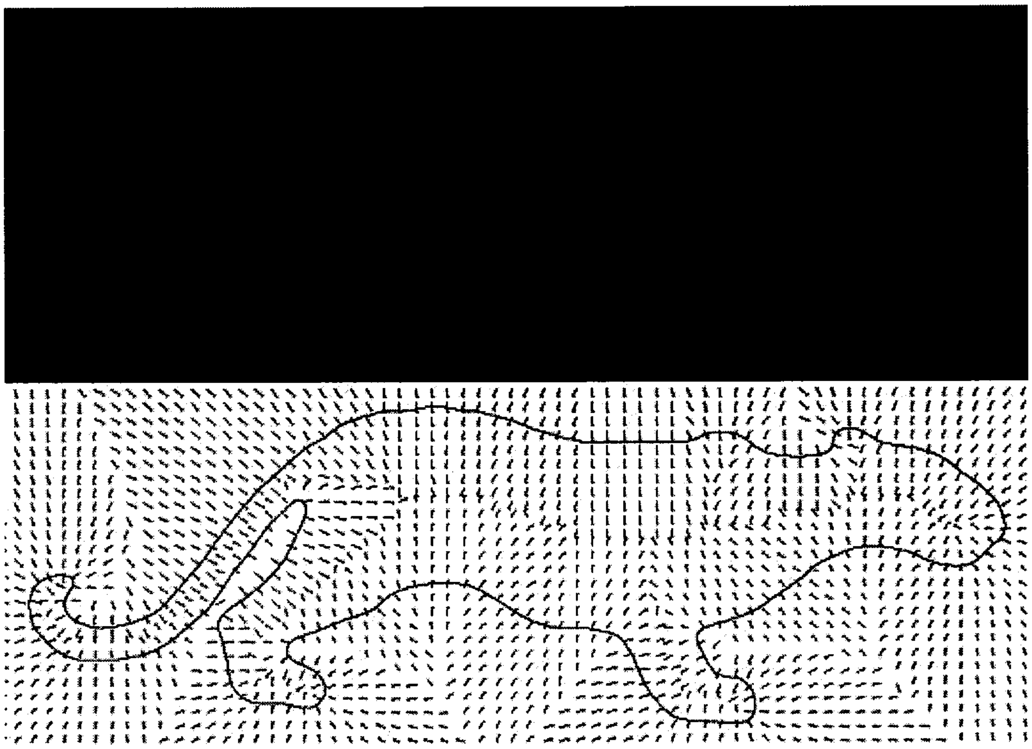


Figure 3.1: TOP: The signed Euclidean distance transform of a panther shape. Brighter yellow denotes greater positive values, while brighter blue denotes greater negative values. BOTTOM: The gradient of the signed Euclidean distance transform of the panther shape. Adopted from [18].

Euclidean distance transform of the shape. Examination of Figure 3.1 illustrates this relationship for the case of the panther shape.

Rather than finding the local maxima in the signed Euclidean distance transform of a shape, we shall be studying its gradient. The gradient of the distance transform $\nabla D_B : \mathbb{R}^3 \to \mathbb{R}^3$ is a vector field, with vectors inside B pointing away from B. Figure 3.1 shows the gradient of the signed Euclidean distance transform of the panther shape.

As we have just mentioned, $\hat{\mathbf{N}} = \nabla D$ in the grassfire flow equation. Therefore, the locations where the advancing fire fronts meet, or the medial surface, may be interpreted as the locus of those points inside the shape for which ∇D is multi-valued, or locations that are equidistant from two or more parts of the boundary.

These locations are the local maxima of D. Rather than finding local maxima of D, we want to find the locations where ∇D is multi-valued.

As we are going to be studying the vector field ∇D in a region, consider the measure of outward flux through the surface S of a small sphere $\mathbb{S}$. The outward flux of ∇D through S, with outward normal N_S, is

$$OF_S(\nabla D) = \iint_S \nabla D \cdot N_S dS,$$

and can be interpreted as the degree to which the flow produced by ∇D is volume preserving. If ∇D models the movement of water through S, then the outward flux of ∇D is negative if more water is coming out than coming in to the region S, positive if more water is coming in than coming out, and zero if the amount of water coming in and out of S is the same. As the region S shrinks to a point, the flux of ∇D through S tends to zero. Therefore, studying $OF_S(\nabla D)$ in regions of small size to detect the presence of medial sheets at those locations is not conclusive.

However, as we shall see in the coming discussion, the value of the average outward flux through S

$$\mathcal{AOF}_S(\nabla D) = \frac{\iint_S \nabla D \cdot N_S dudv}{\iint_S dS} \tag{3.1}$$

as S shrinks to a point is related to the object angle of points on the medial surface sheet, if any, passing inside S. Specifically, we want to be able to study the $\mathcal{AOF}_S(\nabla D)$ in regions through which passes a section of the medial surface $\mathcal{MS}$. We would like to apply the Divergence Theorem and interpret the flux surface integral as a volume integral. However, for those regions of space where the medial surface passes, ∇D is multi-valued and hence, the regular version of the Divergence Theorem does not apply. Exactly for this purpose, Dimitrov *et al.* [17] have developed the Extended Divergence Theorem:

$$\mathcal{OF}_S(\nabla D) = \iint_S \nabla D \cdot N_S dS = \iiint_{\mathbb{S}} div(\nabla D) d\mathbb{S} - 2 \iint_C \nabla D \cdot N_C dC. \tag{3.2}$$

Here, since our object is embedded in $\mathbb{R}^3$ and $\nabla D : \mathbb{R}^3 \to \mathbb{R}^3$, $\mathbb{S}$ is a solid sphere with boundary S and C is the region of $\mathcal{MS}$ intersected by the interior of S. The more general treatment and derivation of an extended divergence theorem for a vector field with a locus of discontinuities is provided in [14].

When considering the $\mathcal{AOF}$ through the surface S shrinking to a point x, the first term on the right side of Equation 3.2 tends to zero, while the second term does not:

$$\begin{aligned}
\lim_{area(S)\to 0} \mathcal{AOF}_S(\nabla D) &= \lim_{area(S)\to 0} \frac{\iint_S \nabla D \cdot N_C dC}{\iint_S dS} \\
&= \lim_{area(S)\to 0} \frac{\iiint_{\mathbb{S}} div(\nabla D) d\mathbb{S} - 2\iint_C \nabla D \cdot N_C dC}{\iint_S dS} \\
&= -2 \cdot \lim_{area(S)\to 0} \frac{\iint_C \nabla D \cdot N_C dC}{\iint_S dS} \\
&\leq -2 \cdot \lim_{area(S)\to 0} \frac{\iint_C dS}{\iint_S dS} \min(\nabla D \cdot N_C) \\
&= -\mu \cos(\min_i(\frac{\pi}{2} - \theta_i)) \\
&= -\mu \sin(\min_i \theta_i).
\end{aligned}$$

The θ_i are the object angles of points on medial sheets meeting at x, if any. Since $\theta_i \in [0, \pi/2]$, $sin(min_i\theta_i) \geq 0$, and for $\mu \geq 0$, the limiting value of $\mathcal{AOF}_S \leq 0$. A more detailed discussion can be found in Chapters 3 and 4 of [47].

3.1.2 Medial Densities for Shrinking Spheres

The values of μ are summarized in Table 3.1 for different types of points inside Ω when the region S is the surface of a sphere. For a review of what these types are, refer to Figure 1.2. Observe that the value of the $\mathcal{AOF}$ for non-medial points is zero. For points lying on the medial surface that are not endpoints, the value of the $\mathcal{AOF}$ is at most $-\frac{1}{2}\sin(\theta)$. In the literature, μ is refered to as the "medial density" and captures the relationship between the limiting value of the average outward flux, object angle and type of medial point at the limiting location.

The reason why non-medial points have $\mathcal{AOF} = 0$ is that as we zoom in on a section of ∇D that does not contain a medial sheet, ∇D looks like a parallel vector field, despite possible expansions at a greater scale. However, whenever a medial sheet does pass through our infinitesimal region S, ∇D is contracting inside, i.e., more water comes in than comes out, and the value of $\mathcal{AOF}$ is negative. The magnitude of the $\mathcal{AOF}$ is proportional to the "degree of contraction" or the object angle of the medial sheet. Refer to Figure 3.2 for an illustration.

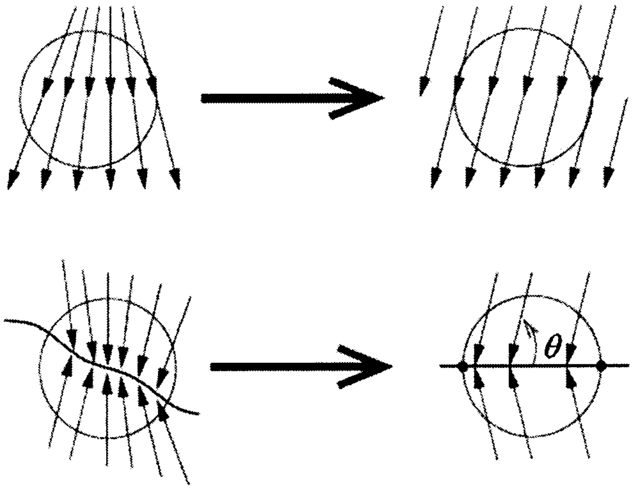


Figure 3.2: TOP: Zooming in on a section of ∇D that does not contain a medial sheet BOTTOM: Zooming in on a section of ∇D where a medial sheet passes. Modified from [8].

We now derive the values for the constants in Table 3.1 adopted from J. Damon [14] for the special case of shrinking spheres. A similar derivation in 2D has been presented in [17]. For smooth medial points, C is just a circle with the same radius r as S, so μ is $2 \cdot \frac{\pi r^2}{4\pi r^2} = \frac{1}{2}$. In the case of a fin point, the sheet that supports the fin intersects the sphere in a circle and so the value of μ is the same as for smooth medial points. In case of a Y-junction point, there are 3 sheets meeting along a seam, each sheet in the limit contributing a semi-circle of area $\frac{\pi}{2}r^2$. Therefore, the

smooth medial point	$\frac{1}{2}$
fin point	$\frac{1}{2}$
Y-branch point	$\frac{3}{4}$
6-junction point	$\geq \frac{1}{2}$
end point	0
non-medial point	**0**

Table 3.1: Value of constant μ for a spherical region for different types of points [14].

total contribution is $2 \cdot 3 \cdot \frac{\frac{\pi}{2} r^2}{4\pi r^2} = \frac{3}{4}$. For a 6-junction point, the 6 sheets contribute at least a circle in total. As the object angle is not defined for edge points, and neither is the tangent plane, the smallest value of μ for edge points is 0. Therefore, for medial points that are not edge points, the value of μ is at least $\frac{1}{2}$.

We have reviewed a theory that starts with the grassfire flow and analyzes the vector field induced by this flow at individual points inside our object Ω. We have shown that the value of the Average Outward Flux ($\mathcal{AOF}$) of this vector field at a point is related to the object angle of that point if it lies on a medial sheet. As our goal is to locate medial sheets algorithmically, we will adopt this measure for use by a computer in regions of non-zero size in the following sections. We shall consider the discretization of this measure for an object with a polyhedral boundary and shall raise and address some significant computational issues neglected by previous approaches.

3.2 From Points to Spheres

3.2.1 Discretizing the Average Outward Flux Measure

We seek to discretize the measure of $\mathcal{AOF}$ through a convex, closed 2D surface S

$$\mathcal{AOF}_S(\nabla D) = \frac{\iint_S \nabla D \cdot N_S du dv}{\iint_S dS}.$$

The following discretization has been proposed in [17, 48]:

$$AOF_S(\nabla D) = \frac{\sum_{i=1,\ldots,N} \nabla D(p_i) \cdot N_S(p_i) da_i}{Area(S)},$$

where p_i are points sampled uniformly on S and da_i is the area of the patch of S corresponding to p_i.

A way to compute the values of da_i is to consider the areas of the Voronoi regions on S for each point p_i. In case one chooses to compute these, algorithms that construct the Voronoi diagram of a set of points on the surface of a sphere exist [42]. However, for N large and the distribution of points p_i roughly uniform, the areas of these patches da_i will be roughly the same. In this case, a simpler version of the discrete formula is:

$$AOF_S(\nabla D) = \frac{\sum_{i=1,\ldots,N} \nabla D(p_i) \cdot N_S(p_i)}{N}. \tag{3.3}$$

The above equation has been used to approximate the continuous values of the $\mathcal{AOF}$ [48] with the interpretation that large negative values indicate that a medial surface passes inside S, and small negative and all positive values mean that no piece of the medial surface passes inside S or that the object angle of points on the medial surface passing inside S is small. This interpretation is based on the assumption that the results of the case when S shrinks to a point apply to the case when S bounds an the area of non-zero size.

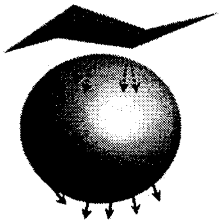


Figure 3.3: Inside the region S ∇D is both contracting and expanding. The AOF of ∇D over S may be greater than the negative value related to the object angle of the medial sheet that is introduced by the convex edge.

However, no matter how small the area bounded by S is, given an arbitrary boundary B, it is always possible for ∇D to be both contracting and expanding in the interior of S. For a practical example, consider Figure 3.3. The arrows denote ∇D vectors. In this example, a medial surface sheet passes inside S because of the convex edge. However, its negative contribution to AOF is reduced by the expansion in ∇D away from the concave edge in the vicinity. This poses a serious issue for the correctness of discrete algorithms that find the medial surface based on the AOF computed using Eq. 3.3. Furthermore, as we will see later, even if we were assured that a medial sheet passed inside S, in the case of a non-zero sized region, the relationship with the object angle may not be as stated in Table 3.1.

In the following section, we develop results that allow us to correctly apply a variant of the AOF measure in regions of non-zero size to detect medial sheets and to reason about the object angle of these surfaces for objects with a polyhedral boundary. An abridged version of these results was presented in [51].

3.2.2 Detecting Medial Sections

Recall that B is the boundary of a closed, orientable polyhedral solid. This boundary may also be described as a closed, orientable, polygonal mesh. Also, S is the surface of a sphere lying inside B. Let D_B be the distance transform of the boundary B. $\nabla D_B : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a unit vector field defined uniquely at each point p inside the region B, with the exception of points on the medial locus where it is multi-valued. Let B_p be the closest point on B to p, which is also the first intersection of a ray cast from p in the direction $-\nabla D(p)$ with B. In case of ambiguity, we pick one closest point on B arbitrarily.

Let $N_S(p)$ be the unit outward normal to S at p. Let N_f be the unit inward normal to a polygonal face f of B.

Definition 3.2.1. *A* ***dihedral angle*** *between two faces of a closed orientable mesh is the internal angle made by the planes containing the faces.*

Definition 3.2.2. *A* ***concave edge*** *e of B is one such that the dihedral angle is greater or equal to π. A* ***convex edge*** *is one that is not concave.*

Definition 3.2.3. *A* ***concave vertex*** *v of B is one such that at least 2 of the faces that share this vertex have a dihedral angle strictly greater than π. A* ***convex vertex*** *is one that is not concave.*

Figure 3.4 provides an illustration of the following definitions.

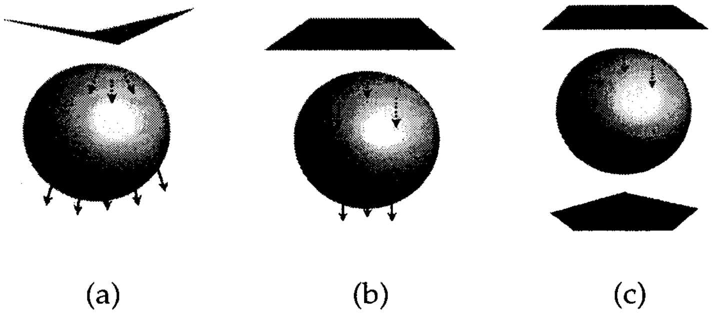


Figure 3.4: (a) 2-sided fan (b) 2-sided parallel section (c) 1-sided fan and 1-sided parallel section

Definition 3.2.4. *A* ***complete fan*** *$F_e(S)$, or simply* ***fan*** *of a concave edge $e \in B$ through*

S is defined as $F_e(S) = \{a|a \in S, B_a \in e\}$. A ***fan*** *$F_v(S)$ of a concave vertex $v \in B$ through S is defined as $F_v(S) = \{a|a \in S, B_a = v\}$.*

We use $F_e(S)$ to refer to all fans, unless greater specificity is required.

Definition 3.2.5. *Let S' be a section of S. Then $S' = out(S') \cup in(S')$, where*

$$\begin{aligned} out(S') &= \{a|a \in S', \nabla D(a) \cdot N_S(a) \geq 0\} \\ in(S') &= \{a|a \in S', \nabla D(a) \cdot N_S(a) < 0\}. \end{aligned}$$

Definition 3.2.6. *Consider a point a on S. Then* ***opp****(a) is the point of intersection of an open ray, cast from a in the direction $\nabla D(a)$ or $-\nabla D(a)$, with S. In the case that $\nabla D(a)$ is tangent to S, $opp(a) = a$.*

Definition 3.2.7. *A* ***one-sided fan*** *$F_e^1(S)$ is a subset of $F_e(S)$ such that $F_e^1(S) = \{a|\nabla D \cdot N_S(a) < 0, opp(a) \notin F_e(S)\}$.*

Definition 3.2.8. *A* ***two-sided fan*** *$F_e^2(S)$ is a subset of $F_e(S)$ such that $F_e^2(S) = \{a|opp(a) \in F_e(S)\}$.*

Definition 3.2.9. *A* ***parallel section*** *$P_f(S)$ through S, of a face $f \in B$, is defined as $P_f(S) = \{a|a \in S, B_a \in f, \nabla D(a) = N_f\}$.*

Definition 3.2.10. *A* ***one-sided parallel section*** *$P_f^1(S)$ is a subset of $P_f(S)$ such that $P_f^1(S) = \{a|\nabla D(a) \cdot N_S(a) < 0, opp(a) \notin P_f(S)\}$.*

Definition 3.2.11. *A* ***two-sided parallel section*** *$P_f^2(S)$ is a subset of $P_f(S)$ such that $P_f^2(S) = \{a|opp(a) \in P_f(S)\}$.*

In the following we shall use AOF to refer to the discrete form of the average outward flux described by Eq. 3.3.

Proposition 3.2.12. *If ∇D is parallel on S, then it is parallel inside S.*

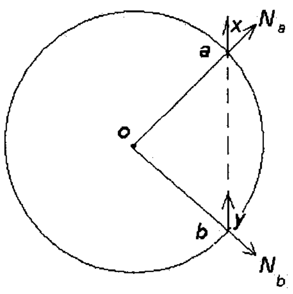


Figure 3.5: Proof of Lemma 3.2.13

Proof. Suppose ∇D is not parallel inside S. Then there are two points a and b inside S such that $\nabla D(a) \not\parallel \nabla D(b)$. Let a_s be the point of intersection of a ray projected from a in the direction $-\nabla D(a)$ with S. Define b_s similarly. Since for both a and b $-\nabla D(a)$ and $-\nabla D(b)$ are the directions to the closest point on B, then for a_s and b_s the directions towards the closest point on B are also $-\nabla D(a)$ and $-\nabla D(b)$, which are not parallel, which is a contradiction. □

Lemma 3.2.13. *Consider a pair of distinct points a and b on the surface of S such* $\nabla D(a) = \nabla D(b)$ *and such that* $a = opp(b)$. *Then these two points contribute* 0 *to the AOF, i.e.* $\nabla D(a) \cdot N_S(a) + \nabla D(b) \cdot N_S(b) = 0$.

Proof. Consider Figure 3.5. Let x be the angle between $\nabla D(a)$ and $N(a)$. Let y be the angle between $\nabla D(b)$ and $N(b)$. Let o be the center of S. Then $\|\bar{oa}\| = \|\bar{ob}\|$ and $\measuredangle oab = \measuredangle abo$ (since $\triangle$ oab is isosceles). But $\measuredangle oab = x$ and $y = \pi - \measuredangle abo = \pi - x$. In other words,

$$x + y = \pi. \tag{3.4}$$

In case x or y is π, the Lemma follows directly since the dot products are 1 and -1. For the remaining cases of $x, y \in [0, \pi)$, we may take the cosine of both sides of

Equation 3.4:

$$\begin{aligned}\cos(x+y) &= \cos\pi = -1 \\ \cos x\cos y - \sin x\sin y &= -1. \end{aligned} \tag{3.5}$$

Similarly, we may take the sine of both sides of Equation 3.4:

$$\begin{aligned}\sin(x+y) &= \sin\pi = 0 \\ \sin x\cos y + \cos x\sin y &= 0 \\ \frac{\sin x\cos y}{\cos x} &= -\sin y. \end{aligned}$$

Substituting in Equation 3.5,

$$\begin{aligned}\cos x\cos y + \frac{\sin x\sin x\cos y}{\cos x} &= -1 \\ \frac{\cos y}{\cos x}(\cos^2 x + \sin^2 x) &= -1 \\ \frac{\cos y}{\cos x} &= -1 \\ \cos y &= -\cos x \\ \cos x + \cos y &= 0 \\ \nabla D(a)\cdot N_S(a) + \nabla D(b)\cdot N_S(b) = 0, \end{aligned}$$

where the last equality follows because ∇D and N_S are unit vectors. □

Proposition 3.2.14. *A two-sided parallel section contributes* 0 *to the* AOF*. A two-sided fan contributes positively to the* AOF*.*

Proof. For all points p in parallel two-sided sections P_f^2, $\nabla D(p) = N_f$. We know that $|in(P_f^2(S))| = |out(P_f^2(S))|$. Points of $in(P_f^2)$ may be uniquely paired together with points of $out(P_f^2)$ such that the contribution of each pair is exactly 0 for two-sided parallel sections, by Lemma 3.2.13. In the case of two-sided fan sections F_e^2, $|in(F_e^2)| < |out(F_e^2)|$. Points of $out(F_e^2)$ may not be uniquely paired up with points of $in(F_e^2)$. The total contribution of F_e^2 is positive, since the contribution $\nabla D(a)\cdot N_a$ of points a of $out(F_e^2)$ is by definition positive. □

Proposition 3.2.15. *One-sided parallel sections $P_f^1(S)$ and one-sided fans $F_e^1(S)$ contribute negatively to the AOF.*

Proof. By definition, $|out(P_f^1(S))| = |out(F_e^1(S))| = 0$. Therefore, the contribution to the AOF is a sum of negative $\nabla D(a) \cdot N_S(a)$ terms, and so the total contribution is negative. □

Proposition 3.2.16. *Points in S belong either to parallel sections or fans.*

Proof. For points a in S, some B_a lie on the faces of B, others lie on edges, while others lie on vertices. For those points a for which B_a lie on faces but not on edges, $\nabla D(a)$ must be normal to these faces.

We consider two kinds of edges and vertices: convex and concave. Consider a point p strictly inside B. For a convex edge or vertex of B, there is no such point p such that B_p lies on it, because there is always a point within $\epsilon > 0$ of the convex edge/vertex that is closer to p. To see this, suppose the contrary. For some convex edge or vertex, suppose the closest point on B to a point p inside B was on this edge or vertex. But we know that the shortest path from a point to a plane is along a perpendicular to that plane. The distance along a perpendicular to one of the planes meeting at the convex edge or vertex is shorter than the distance to the convex edge or vertex. Therefore, for a convex edge or vertex, no B_p lies on it. Those points p whose closest points on B lie on concave vertices or edges are part of fans, by definition. □

Theorem 3.2.17. *The medial surface of the solid with boundary B passes inside the region S if and only if the AOF due to all the regions of S, except for the two-sided fans, is negative.*

Proof. [$\Rightarrow$] Suppose the medial surface passes through the region S inside B. Then S cannot be a two-sided parallel section, for if it were, by Proposition 3.2.12, ∇D

would also be parallel inside S and, provided that the sphere sampling rate N is large enough, no medial surface would pass through it. Note also that the situation of multiple two-sided parallel sections without a fan section is impossible. From Proposition 3.2.16, S contains points that belong to (two-sided and 1-sided) parallel sections and fans only. Since we know that there is a section of points in S for which ∇D is not a two-sided parallel section, there is at least one fan or one-sided parallel section. When N is large enough, we are assured to find some of these points. By Proposition 3.2.14, we may ignore the two-sided parallel sections, if any, since they don't contribute to the AOF. Also by Proposition 3.2.14, by ignoring the two-sided fans, we ignore a positive contribution to the AOF. The remaining regions – one-sided fans and parallel sections – contribute negatively, by Proposition 3.2.15. These regions exist because ∇D is multi-valued inside S.

[$\Leftarrow$] Suppose that the AOF due to all regions except for two-sided fans is negative. The remaining regions are one-sided fans and two-sided and one-sided parallel sections. By Proposition 3.2.14 we know that two-sided parallel sections contribute zero to the AOF. Therefore, the negative value must have come from a one-sided fan or one-sided parallel section by Proposition 3.2.15 and we are assured of their existence. For points a in these sections, $\nabla D(a) \neq \nabla D(opp(a))$ because otherwise both a and $opp(a)$ would share a closest point on B and would then be part of a two-sided fan or parallel section. Thus ∇D is multi-valued inside S, meaning that the medial surface passes inside S. □

In the above proof, note that the "if" portion of the proof did not rely on N being large. So, if for some region S, AOF' is negative, we can be sure that a medial surface passes inside. However, if the medial surface does pass inside, we cannot be assured that the AOF' will be negative, except for sufficiently large N. Therefore, for small N, negative AOF' is a sufficient but not necessary condition for the presence of a medial surface sheet. We shall examine the effect of the sphere sampling rate N on correctness in Chapter 5.

In the ensuing discussion, we shall adopt the following notation:

Definition 3.2.18. *Let $AOF'_S(\nabla D)$ be the value of the $AOF_S(\nabla D)$ discounted by the contribution of two-sided fans.*

In the coming discussion, we divide space inside Ω into regions, place spheres in those regions and use their AOF' values to decide the presence of medial sheets inside those regions. This approach will prove very useful in designing a coarse-to-fine algorithm that approximates the medial surface using discrete regions with arbitrary precision and also allows to apply existing digital homotopy type preservation algorithms to the resulting approximation.

3.2.3 Relationship between AOF′ and Object Angle θ

Consider the cubic lattice $\mathbb{L}_\sigma = \sigma\mathbb{Z} \times \sigma\mathbb{Z} \times \sigma\mathbb{Z}$ with spacing σ. Place at each point $x \in \mathbb{L}_\sigma$ a cube $C_\sigma(x)$ of size σ^3 (also called *voxel*). Also, let $S_\sigma(x)$ be the (surface of a) sphere of radius σ centered at $x \in \mathbb{L}_\sigma$. The subscript σ will be dropped unless required.

Definition 3.2.19. *The voxelized interior* $\textbf{Inside}_\sigma\textbf{(B)}$ *is the set of lattice points $x \in \mathbb{L}_\sigma$ such that $S_\sigma(x)$ is contained in the interior of B.*

We shall be evaluating the $AOF'_S(\nabla D)$ for spheres $S_\sigma(x)$ at points $x \in Inside_\sigma(B)$. We know that when the sphere's surface is sampled finely enough, those $S_\sigma(x)$ for which $AOF'_{S_\sigma(x)}(\nabla D)$ is negative contain sections of the medial surface. We now wish to reconcile the relationship of AOF' with the object angle θ in the case when $\sigma > 0$.

Earlier in this chapter, we saw that whenever we shrink the surface of a sphere S about a smooth medial point, $AOF = AOF' = -\frac{1}{2}\sin\theta$, where θ is the object angle of the medial surface at this smooth point. This value was obtained by considering the intersection of S with the medial surface section. As S shrinks to a point, this intersection is a circle having the same radius as S. Unfortunately, in the case when S has non-zero radius, if the medial surface does indeed pass through a cell $C(x)$, it does not have to pass through the center of $S(x)$. In fact, it can just graze the corner, as shown in Figure 3.6.

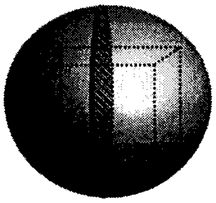


Figure 3.6: Intersections of a medial surface sheet with a sphere. The pink is the minimum area of intersection and the green is the maximum area.

Proposition 3.2.20. *If a planar section of $\mathcal{MS}$ with object angle θ passes through a cell $C(x)$ that is everywhere as wide as $S_\sigma(x)$, then AOF' through $S_\sigma(x) \in [-\frac{1}{2}\sin\theta, -\frac{1}{8}\sin\theta]$.*

Proof. The lower bound is achieved by revisiting the case where the intersection of $S_\sigma(s)$ with $\mathcal{MS}$ is a circle of radius σ. The upper bound is obtained when $\mathcal{MS}$ passes through the corner of $C(x)$ (refer to Figure 3.6). In this case, the area of the circle where the medial surface cuts $S_\sigma(x)$ is $\pi(\sigma/2)^2$ and value of AOF' is $-2\frac{\pi(\sigma/2)^2}{4\pi\sigma^2}\sin\theta = -\frac{1}{8}\sin\theta$. □

When the medial surface is curved, the constant μ will increase as the area of intersection of the medial surface with the sphere will be higher. However, we still have an important relationship with the smallest possible value of constant μ relating the negative sine of the object angle and the value of AOF':

Corollary 3.2.21. *If a section of the medial surface with object angle at least θ passes through a cube $C_\sigma(x)$ that is everywhere as wide as $S_\sigma(x)$, then AOF' through $S_\sigma(x)$ is at most $-\mu \sin\theta$, with $\mu = \frac{1}{8}$.*

The choice of the sphere's radius σ is crucial to the bound in Corollary 3.2.21. If the sphere $S(x)$ is made to circumscribe the cube $C(x)$ by having radius $\sqrt{3}/2\sigma$, then nothing can be said about the minimum value of the constant μ. If the radius is made larger than σ, then μ increases, but the amount of overlap between spheres about neighbouring cubes becomes very significant. Thus, our choice of sphere radius σ is both simple and appropriate. In fact, it is also the choice suggested by P. Dimitrov in [17]. A study into capturing the redundancies among neighbouring spheres would allow us to increase the spheres' radii.

We now illustrate, via a simple example, the effect of discretization on capturing object angle information. Figure 3.7 (top), demonstrates the x, y and z slices of the medial surface of a box, where the medial surface sheets happen to pass right through the centre of each voxel. For this synthetic example, we observe that the value of AOF' in the middle solid section (shown darkest in the Figure) is -0.5003, which is within 0.06 % of the expected value of $-\frac{1}{2}\sin(\pi/2) = -0.5$ for this medial sheet of object angle $\pi/2$. The other 8 sheets of the medial surface have AOF' value around -0.3524. For these sheets, the object angle is $\pi/4$ and the AOF' value obtained is within 0.34% of the value expected of $-\frac{1}{2}\sin(\pi/4) \approx -0.3536$. The reason why the values obtained differ from the expected values in this synthetic case is due to finite sphere sampling. If we shift the coordinates of the box mesh by $\sigma/2$, then the medial surface passes between two voxels, as in Figure 3.7 (bottom). The value of AOF' in the middle section is -0.3315, rather than -0.5. This is precisely because the medial sheet no longer passes through the center of the voxels $C(x)$ and their spheres $S(x)$.

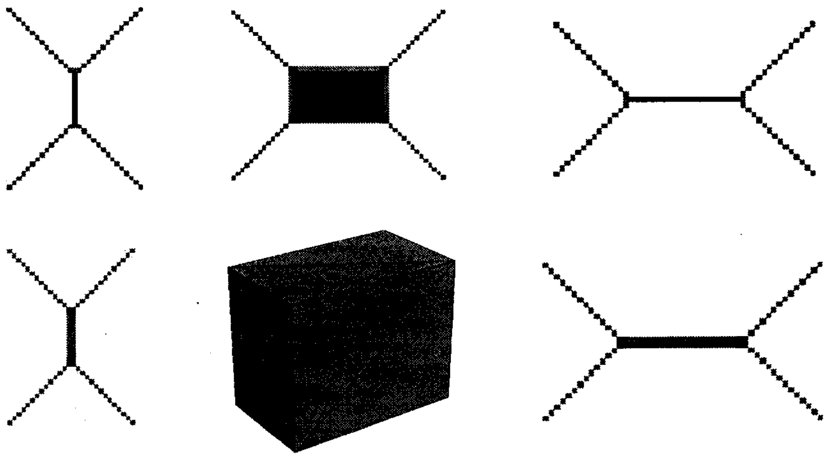


Figure 3.7: TOP: the x, y, and z slices of the medial surface for the box with darker values representing higher AOF values. BOTTOM: x and y slices when the medial surface passes between two voxels.

3.2.4 Thickness and Homotopy Type

Recall from Chapter 1 that the medial surface $\mathcal{MS}$ of the object Ω is both thin and homotopy-equivalent to Ω. As can be observed from Figure 3.7, depending on the choice of voxelization, the medial surface may be thin (1 voxel wide) or thick (more than 1 voxel wide). In this section, we show the effect of discretization on thickness and the homotopy type of the set of voxels having negative AOF'.

Definition 3.2.22. *Let* $L_\sigma(B) = \{x \in \mathbb{L}_\sigma | AOF'_{S_\sigma(x)}(\nabla D_B) < 0, x \in Inside_\sigma(B)\}$.

According to Theorem 3.2.17, $L_\sigma(B)$ is the set of voxels interior to B that contain a piece of the medial surface, or the voxelized approximation to the $\mathcal{MS}$ of the interior of B.

Definition 3.2.23. *The* ***thickness*** *of a voxelized approximation* $L_\sigma(B)$ *to the medial surface sheet at a smooth point* p *is measured as the maximum distance between the voxels used in the approximation located in the directions* $\pm N_p$ *from* p*, the normal directions to the medial sheet at* p*, that are not also used in the approximation of some other medial sheet.*

*The **thickness** of $L_\sigma(B)$ is the maximum thickness over all smooth points p.*

In the above definition, in case of sheets degenerating to a curve or point, the normal directions to points on such sheets are all the points on the unit sphere at $(0, 0, 0)$.

Proposition 3.2.24. *The thickness of $L_\sigma(B)$ is at most 2σ.*

Proof. Only those points $x \in \mathbb{L}_\sigma$ will be part of $L_\sigma(B)$ for which $S_\sigma(x)$ contains a piece of the medial surface of the interior of B. For those $S_\sigma(x)$ centered closer than σ to the medial surface, some portion of the medial surface passes inside. Consider a smooth medial point p. Then the thickness at p is the maximum distance between voxels located at positions $x \in \{p + \alpha N_p | \alpha \in \mathbb{R}\}$, such that $S_\sigma(x)$ contains p. The thickness is σ if p is located at a lattice point, since none of the spheres about the 26-neighbours of that lattice point contain p. In case p is not located at a lattice point, spheres that are located further than $p \pm \sigma N_p$ away cannot intersect p. So the furthest distance between voxels about a smooth point is 2σ. □

In the case of degenerate medial sheets, a similar argument proves the proposition. Note also that had the radius of the spheres been greater than σ, we would not be able to guarantee a thickness of at most 2σ. We have shown that the set of voxels approximating the medial surface cannot be very thick.

It remains to consider the homotopy type of $L_\sigma(B)$. It is known that the medial surface preserves the homotopy type of the original object. However, because $L_\sigma(B)$ is composed of voxels, even if we were able to guarantee that all the voxels through which the medial surface passes are part of $L_\sigma(B)$, the number of cavities, handles and connected components does not necessarily have to be correct, because the voxelization "inflates" the medial surface. Further, as we are also keeping those voxel cells $C(x)$ such that their spheres $S(x)$ intersect the medial surface but the cells $C(x)$ do not, these extra voxels may alter the topology of $L_\sigma(B)$. Luckily, there is a simple way to deal with homotopy type preservation in voxelized data sets, and we explore this method in the coming chapter.

As we have established the necessary criterion for deciding the presence of a medial surface section in spherical regions of non-zero size as well as the relationship between the object angle, the value of the AOF', and the radii of the spheres when they are centered on the cubic lattice, we may now design a coarse-to-fine algorithm for finding sections of the medial surface with a desired minimum object angle. Such an algorithm will be able to carve out sections of space that potentially contain medial surface sheets, with increased spatial resolution at each level of recursion.

Chapter 4

Coarse-to-Fine Algorithm

Iterative logical networks are not needed for implementation. Indeed, lattice organized devices can in some important ways be a deterrent to a good grassfire. For example, good circularity requires the use of a large number of element-to-element connections at all distances, or exceedingly complex communications and computations at each point.

– H. Blum

In this chapter we design a coarse-to-fine algorithm for detecting those locations inside Ω where $\mathcal{MS}$ passes. We start by presenting a basic version of our algorithm and point out those steps that require further discussion. Three of these steps are well-known problems in the literature: distributing points on a sphere, point-to-mesh distance and digital homotopy type preservation. We proceed to discuss the background of these problems, existing solutions and the particular solution we have chosen to adopt in our implementation. The fourth problem discussed is specific to our algorithm and we present a novel solution. We also discuss a number of implementation suggestions for reducing the computational cost in terms of time and space that our algorithm uses. Finally, we summarize the complete algorithm that unifies all the algorithms presented in this section. A number of experimental results are presented that illustrate qualitatively and quantitatively the strengths of the algorithm we have designed.

4.1 Existing Algorithm

Given an input polygonal mesh boundary B of a 3D object Ω, we would like to locate those points on Ω's medial surface that have relatively high object angle. We choose a desired starting resolution σ and make a 3D voxel array $Inside$ consisting of those voxels whose circumscribing spheres of radius σ fit completely inside B. The overview of the algorithm COARSETOFINE is given by Algorithm 1.

The following subsections study the steps 3, 6, 11, 14, 15 and 18 of Algorithm 1 in further detail, providing theoretical background for the subproblem solved in each step and details of the implementation we have chosen.

4.1.1 Sphere Sampling

In step 3 of Algorithm 1, we would like to sample a sphere with N points, such that these points are distributed as uniformly as possible in order for Equation 3.3 to be a good approximation to Equation 3.1.

Suppose we could distribute N points uniformly on a sphere. Consider the Voronoi diagram of these points. Construct a polyhedron with vertices at locations on the sphere where 3 or more Voronoi faces meet and edges connecting vertices that are part of Voronoi faces. Then the faces of such a polyhedron must be congruent and all the angles and vertices equal. This condition implies that the

```
Input: Polygonal mesh boundary $B$, its voxelized solid shape $Inside$ with voxel
    size $\sigma^3$, sphere sampling rate $N$, resolution increment $d$, number of levels of
    subdivision required $levels$, object angle $\theta$.
Output: The voxels of $Inside$ subdivided $levels$ times by $d$ through which passes
    a piece of the medial surface wider than $\sigma/d^{levels}$ with object angle at least $\theta$.
for all voxels $v$ of $Inside$ do
    Let $x$ be the location of $v$
    Place a sphere $S$ of radius $\sigma/(d^{levels})$ at $x$
    $AOF(x) \leftarrow 0$
    for all points $p_i$ on the surface of the sphere, $i = 1 \ldots N$ do
        Find the closest point $y$ on $B$ to $p_i$
        Let $\nabla D(p_i)$ be the unit direction from $y$ to $p_i$
        Let $N_S(p_i)$ be the unit outward normal to $S$ at $p_i$
        $AOF(x) \leftarrow AOF(x) + \frac{\nabla D(p_i) \cdot N_S(p_i)}{N}$
    end for
    Let $AOF'(x)$ be the contribution to $AOF(x)$ of those points $p_i$ that are not
    part of two-sided fans
end for
if $levels > 0$ then
    Delete those voxels of $Inside$ located at $x$ that do not contain a one-sided
    section
    Subdivide the remaining voxels by $d^3$
    CoarseToFine($B, Inside, \sigma, N, d, levels - 1, \theta$)
else
    Delete all voxels located at $x$ of $Inside$ for which $AOF'(x) > -\frac{1}{8}\sin(\theta)$
    Return $Inside$
end if
```

Algorithm 1: CoarseToFine($B, Inside, \sigma, N, d, levels, \theta$)

polyhedron is *regular*. If we were able to distribute uniformly N points on a sphere, for N an arbitrary number, then we would be able to create a regular (or *Platonic*) polyhedron using the above procedure having an arbitrary number of faces. We summarize the well-known argument showing that this is not possible.

The following formula observed by Descartes in 1640 and rediscovered by Euler in 1752 relates the number of vertices V, edges E, and faces F of a simple polyhedron (or an object topologically equivalent to a sphere) [12]

$$V - E + F = 2. \tag{4.1}$$

Furthermore, for a regular polyhedron, each of whose faces is n sided, and for which r edges meet at each vertex, the following conditions must be satisfied:

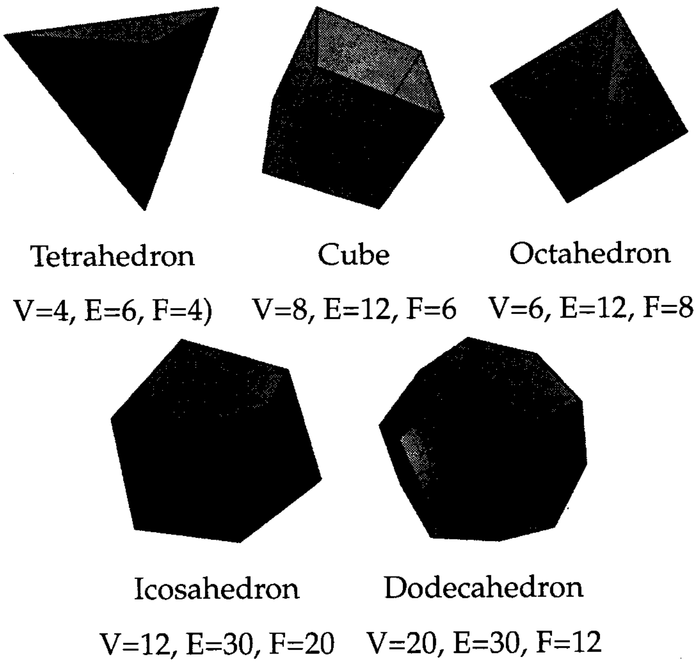
$$nF = 2E$$
$$rV = 2E.$$

Combining these with Equation 4.1, we obtain

$$\frac{1}{n} + \frac{1}{r} = \frac{1}{2} + \frac{1}{E}. \tag{4.2}$$

We know that $n \geq 3$ and $r \geq 3$. In order for E to be positive, both n and r cannot be greater than 4. Whenever $n = 3$, r can equal $3, 4$, or 5 and vice versa. These cases describe the tetraheron, cube, octahedron, icosahedron and dodecahedron shown in Figure 4.1. These are the only 5 possible regular polyhedra. The 20 vertices of the dodecahedron is the configuration distributing the maximum number of points on a sphere uniformly. The (dual) Voronoi polyhedron corresponding to this configuration is the icosahedron.

As distributing N points "as uniformly as possible" on a sphere is desired for so many applications, a large body of work exists looking at variants of this problem [43]. A popular approach is that of icosahedral dissection, which proceeds by iteratively subdividing the triangles of an icosahedron into 4 new triangles and projecting their centers radially onto the sphere. This approach is limited in a number



Tetrahedron V=4, E=6, F=4)

Cube V=8, E=12, F=6

Octahedron V=6, E=12, F=8

Icosahedron V=12, E=30, F=20

Dodecahedron V=20, E=30, F=12

Figure 4.1: The 5 Platonic solids.

of ways: the quantities of points N that may be generated in this way are restricted and points near the equator are spaced further apart than the rest [43]. We have adopted the method of Rakhmanov, Saff, and Zhou [40] which distributes points in the following way:

The points, given in spherical coordinates $(\theta_i, \phi_i), 0 \leq \theta_i \leq \pi, 0 \leq \phi_i \leq 2\pi$, are computed as described below.

$$\theta_i = \arccos(h_i), \quad h_i = -1 + \tfrac{2(i-1)}{N-1} \quad \text{for} \quad 1 \leq i \leq N.$$

$$\phi_i = \begin{cases} \left(\phi_{i-1} + \frac{3.6}{\sqrt{N}} \frac{1}{\sqrt{1-h_i^2}}\right) \mod 2\pi & \text{for} \quad 2 \leq i \leq N-1 \\ 0 & \text{for} \quad i = 1, N \end{cases}$$

The set of generated points forms a spiral, in the following sense: Points are placed starting at the South pole, and continuing to the next lattitude line located $2/(N-1)$ units up along a great circle and across this lattitude line $\frac{3.6}{\sqrt{N}}$ units counter-clockwise to the next placed point. The procedure repeats with one point placed

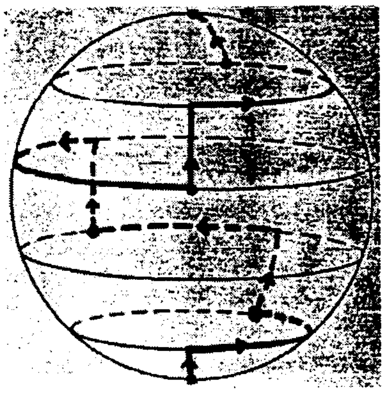


Figure 4.2: The distribution of 6 points on a sphere using the method of [40], adopted from [43].

on each of the N lattitudes, including the poles, as can be seen in Figure 4.2.

The method of [40] is a popular one. In particular, it has been proposed as the method for generating "uniformly distributed" points on a sphere by a leading textbook in Computational Geometry [37].

4.1.2 Point-to-Mesh Distance

The most important operation of COARSETOFINE is finding the closest point on B to a point on the sphere in step 6 of Algorithm 1. This must be done as efficiently as possible. Proximity queries of this sort have been studied extensively. Popular approaches use Bounding Volume Hierarchies (BVHs) organizing sets of geometric primitives using a hierarchy of bounding primitive shapes, like boxes. An alternative kind of hierarchy organizes convex patches, where each node is the convex hull of its children. The central idea behind these methods is to allow rejecting uninteresting portions of the mesh quickly, while focusing on those parts that are closest to the query point. This is achieved by building a hierarchy where the primitive shape associated with each parent node bounds the primitive shapes fitted about its children. Another set of approaches uses Binary Space Partitions

(BSPs), such as k-d trees and octrees.

Bounding Volume Hierarchies

The different kinds of shapes used in Bounding Volume Hierarchies (BVHs) are Axis-Aligned Bounding Boxes (AABBs)[5], Oriented Bounded Boxes (OBBs) [24] and Sphere-Swept Volume (SSVs) [27, 25]. The bounding hierarchies for the two objects whose distance is desired are traversed in tandem, and each bounding volume gives lower and upper bounds on the distance between the objects. These bounds allow pruning of portions of the hierarchy. A special feature of these methods is that they incorporate heuristics for reducing computation in case of two subsequent proximity queries for two objects that have not greatly changed their positions. This is useful for our purposes, since we are interested in computing distances between the mesh and point objects distributed on a sphere. A heuristic ordering of points on the sphere is considered in Section 4.2.

OBBs are bounding boxes of arbitrary orientation in 3-space [24]. Bounding boxes are fitted around primitives by considering principal directions obtained by spectral analysis of the covariance matrix of the convex hull of the mesh triangles and centered at their mean point. Large OBBs are split and new oriented boxes are fitted around the children. OBBs approximate a shape using less levels than AABBs and can be tested for overlap nearly as quickly.

SSVs include point swept spheres, line swept spheres, rectangle swept spheres [27], and triangle swept spheres [25]. In [27], the bounding volumes are fitted to the object in a similar way as OBBs above. The simplicity of the shapes allows for simple proximity tests among primitives. Use of Rectangle Swept Spheres (RSSs) is demonstrated to provide more efficient query time. Furthermore, it is shown that while RSSs and OBBs have the same approximate pruning power, the distance test for OBBs is slower than that for RSSs. In [25], an algorithm that explicitly computes the distance from a point to a triangle mesh is presented, although its details are very similar to [27], except for the choice of bounding volume. Triangles are fitted to the shape by the process of mesh simplification, with the radius of the sphere given by the error accumulated during the simplification.

Convex Surface Decomposition

Reference [20] proposes building a hierarchy of convex surface patches (leaf elements) and their convex hulls (internal nodes) to facitilate distance computation. First, the surface is decomposed into convex pieces using graph search methods. Then, the hierarchy is constructed top-down where each piece is a convex polyhedron. The authors claim that this hierarchy has fewer levels than hierarchies that we will see in the next section. For testing proximity between 2 convex polyhedra in such hierarchies, a Voronoi marching subroutine is used. Reference [20] features the most interesting of the query acceleration ideas based on exploiting the coherence between subsequent queries called "generalized front tracking". It works by caching parts of the current hierarchy for future queries.

Binary Space Partitions

BSPs are standard data structures that partition space with cuts made orthogonal to the coordinate axes. They are used to perform nearest neighbour search. A method that uses k-d trees [26] proceeds by successively making cuts along the x, y, and z coordinate axes through the midpoint in that direction. The cut axis is chosen as one that minimizes the product of the number of mesh elements in each resulting subproblem. A query mesh triangle is processed by checking to which side of the cut plane this triangle falls and recursing in that direction. Once a potential closest triangle is found, cells of the tree at least that far away are checked. It is, however, likely that many triangles in the tree will have to be examined before the true nearest neighbour is found. A suggestion that enforces some restrictions on properties of the k-d tree cells and then considers a priority ordering on their traversal is presented in [4].

In our implementation, we use a BVH of [27] called the Proximity Query Package (PQP) for finding the closest location on the mesh boundary B to points distributed on the sphere by performing exact distance computations between the polyhedron with boundary B and (degenerate) point objects. This method appears to have favourable properties in terms of computation speed compared with the other methods described above for our particular application. For selected datasets, the SWIFT package of [20] has been shown to outperform PQP. On the other hand, PQP is suggested as the appropriate package for point-to-polygonal mesh computations in [23]. In our implementation, the calls to PQP are the most costly of COARSETOFINE. Therefore, it would be highly desirable to compare the timings of SWIFT with PQP for our application.

4.1.3 Finding Two-Sided Fans and One-Sided Sections

In step 11 of Algorithm 1, we need to locate those points p_i that are part of a two-sided fan section in computing AOF'. By definition, p_i is part of a two-sided fan section if $\nabla D(p_i) = \nabla D(opp(p_i))$ and $B_{p_i} \in e$, where e is either an edge or vertex of B. Because of the finite sphere sampling, it may so happen that for a particular p_i we did not query $opp(p_i)$. For this reason, we consider the Voronoi diagram of the points p_i. Algorithm 2 present that algorithm for finding AOF'.

There are a number of notes to be made. First, finding AOF' is only necessary at the last level of the recursion in Algorithm 1, as its exact value is only required

```
Input: The mesh boundary $B$, points $P = \{p_i \in S(x), 1 \leq i \leq N\}$, and all their
    associated $\nabla D(p_i)$ values, $AOF(x)$.
Output: $AOF'(x)$, which is the value of $AOF(x)$ discounted by the contribution of
    two-sided fans.
Let $AOF'(x) = AOF(x)$
for all points $p_i \in S$ do
  if $B_{p_i}$ is within $\epsilon$ of an edge or vertex of $B$ then
    Find $j$ such that $p_j$ is the closest point on $S$ to $opp(p_i)$
    if $\nabla D(p_i) = \nabla D(p_j)$ then
      $AOF'(x) = AOF'(x) - \frac{\nabla D(p_i) \cdot N_S(x)}{N}$
    end if
  end if
end for
Return $AOF'(x)$
```

Algorithm 2: COMPUTE$AOF'(B, P, \nabla D(P), AOF(x))$

for using the relationship with object angle. Also, it would eliminate redundant computation to not compute AOF at all and simply add up the contributions of points not part of two-sided fans in step 11. We feel that both of these changes would hinder the presentation in Algorithm 1.

Computing the Voronoi diagram of a set of points on a sphere is possible using the method of [42]. However, performing queries on this 3D structure is a challenging task. Currently $O(n \log n)$ space and $O(\log^2 n)$ queries, where n is the number of faces of the spatial subdivision, are achievable [50], but the problem of $O(n)$ space and $O(\log n)$ time queries remains open [35]. In our implementation, we are using the brute-force solution.

In step 14 of Algorithm 1, we need to decide whether any one-sided sections (fans or parallel) pass through S. We motivate and provide a method by which to confirm that no medial surface passes through S given the values of ∇D sampled on the sphere.

In Section 4.1.1, we have seen that when $N > 20$, it is not possible to distribute N points uniformly on a sphere. For this reason, when approximating the flux integral in Equation 3.1 with Equation 3.3 when $N > 20$, even in the simple case when all the points on the sphere have their closest points located on the same triangle of B, the result produced by Equation 3.3 is different from zero. For this reason, nonnegative value of $AOF'(S)$ may not be interpreted to mean that no medial surface surface provably passes through S. We suggest using Algorithm 3 to decide if a section of the medial surface passes through a sphere $S(x)$ for positions $x \in Inside$ or if a one-sided fan section exists for points sampled on $S(x)$.

In step 2 of Algorithm 3, one may do a Voronoi diagram look-up as explained above. Note that the error introduced by the finite sphere sampling in the case that $\nabla D(p_i) = \nabla D(p_j)$, where p_j is the site of the Voronoi cell where $opp(p_i)$ lies,

```
Input: The mesh boundary $B$, points $P = \{p_i \in S(x), 1 \leq i \leq N\}$, and all their
    associated $\nabla D(p_i)$ values.
Output: TRUE if no one-sided fans detected and FALSE otherwise.
 1: for all points $p_i \in S$ do
 2:   Find $j$ such that $p_j$ is the closest point on $S(x)$ to $opp(p_i)$
 3:   if $\nabla D(p_i)$ and $\nabla D(p_j)$ are not parallel then
 4:     Find $\nabla D(opp(p_i))$
 5:     if $\nabla D(p_i)$ and $\nabla D(opp(p_i))$ are not parallel then
 6:       Return FALSE
 7:     end if
 8:   end if
 9: end for
10: Return TRUE
```

Algorithm 3: DECIDE$\mathcal{MS}(B, P, \nabla D(P))$

yet $\nabla D(p_i) \neq \nabla D(opp(p_i))$, is studied in Chapter 5. On the other hand, as we want to avoid false positives, we check explicitly in step 5 that indeed a one-sided fan passes through the sphere. However, even if Algorithm 3 returns TRUE for a particular voxel location x, we must make sure that deleting x from $Inside$ does not alter the homotopy type of $Inside$. This issue is addressed in the next section.

4.1.4 Homotopy Type Preservation

Recall that $\mathcal{MS}$ is of the same homotopy type as Ω. Often, applications that use $\mathcal{MS}$ to study the shape of Ω couple the extraction of $\mathcal{MS}$ with preservation of homotopy type.

Our method performs operations on the voxelized interior of Ω, $Inside$. At each stage of the recursion, $Inside$ corresponds to $Inside_\sigma(B)$. Therefore, in Step

15 of Algorithm 1, we not only must subdivide those voxels of $Inside_\sigma(B)$ that do not contain a medial sheet, but we must also add those voxels that are not part of $Inside_\sigma(B)$ but are part of $Inside_{\sigma/d}(B)$. Therefore, the set of voxels of $Inside$ is updated at each level of the recursion to fit most closely the shape of Ω.

If this is desired, we delete voxels in steps 14 and 18 of Algorithm 1 from $Inside$ only if their removal does not change the homotopy type of $Inside$. To do so, we adopt the approach of [8] that uses the condition of [32] to decide if a voxel is removable.

Consider a voxel positioned at x. For $n \in \{6, 18, 26\}$, we use the standard notions of n-connectivity, n-adjacency and n-neighbourhood $N_n(x)$. Let $N_n^*(x) = N_n(x) \setminus \{x\}$. Consider the following two numbers:

- C^*: the number of 26-connected components that are 26-adjacent to $C(x)$ in $Inside \cap N_{26}^*$
- $\bar{C}$: the number of 6-connected components that are 6-adjacent to $C(x)$ in $\overline{Inside} \cap N_{18}$.

Then the following theorem due to Malandain and Bertrand describes those points that may be removed without altering the homotopy of $Inside$ [32]:

Theorem 4.1.1. *A voxel positioned at x is removable from $Inside$ without change in $Inside$'s homotopy type if $C^*(x) = \bar{C}(x) = 1$. Such a voxel is called* simple.

Algorithm 4 is a modified version of the algorithm in [8].

In step 1 of the algorithm, we use the popular method of [7]. The order in which we consider points for removal is key to obtaining a minimal set of voxels from which no other voxel with AOF' value above the given threshold may be

```
Input: The set of voxels $Inside$, their $AOF'$ values,threshold $thresh$
Output: If $thresh = 0$, deletes those voxels of $Inside$ that do not contain a section
    of the medial surface while preserving homotopy type. If $thresh < 0$, deletes
    those voxels of $Inside$ positioned at $x$ such that $AOF'(x) > thresh$ while pre-
    serving homotopy type.
Compute the approximate Euclidean distance transform $E$ of $Inside$
Sort the voxels by their $E$ values to give $InsideSorted$
for all voxels at location $x$ of $InsideSorted$ do
  if The voxel at $x$ is simple then
    if $thresh = 0$ and DECIDEMS(x) then
      Delete the voxel at location $x$ of $Inside$
    end if
    if $thresh < 0$ and $AOF' > thresh$ then
      Delete the voxel at location $x$ of $Inside$
    end if
  end if
end for
Return $Inside$
```

Algorithm 4: HOMOTOPYPRESERVINGDELETE(*Inside*, *AOF'*, *thresh*)

further removed. The heuristic ordering of points for removal that we consider is that given by an approximate Euclidean distance transform where points closer to the outside of the object are considered for removal before those further inside.

Unlike [52], our method preserves the homotopy type of the original object regardless of whether the complete medial surface (with $\theta > 0$) is of the same homotopy type as the original object. This is due to the fact that our homotopy preservation algorithm is applied to the voxelized interior of Ω, rather than to medial sheets. However, our approach assumes that the homotopy type of $Inside$ at the highest resolution desired is the same as that of Ω.

4.2 Time and Space Improvement Suggestions

At present our algorithm runs quite slowly: the time to compute the medial surface with final resolution of $177 \times 129 \times 361$ voxels and 3 levels of subdivision by 27 for a mesh having 5672 triangles is around 2 hours on a computer with a Pentium IV, 3.40 GHz processor having 3 Gb of RAM. Also, the size of the file that stores the final AOF' information is 38 Mb. None of these values is acceptable for practical use. In this section, we consider a number of suggestions for speed-up and reduction of storage used by our algorithm.

We have observed that the most expensive step in Algorithm 1 is step 6. This is a call to the package PQP which computes the closest point on a mesh to a point object. PQP is able to capture coherence between distance queries by "triangle caching": remembering the closest triangle pair from the previous query to initialize the distance between objects to to the distance between these triangles [27]. Such an approach works well when the the distance between the mesh and subsequent point objects changes only slightly. This happens when the change in relative position of the objects (the mesh and the point) is small. The spiral ordering of the points we consider is actually not the ultimate processing order because the path wraps many times around the sphere. An ordering that explores points in one locality before it moves on to the next provides, at least heuristically, a better ordering.

Algorithm 5 describes the order of traversal of points. The local regions considered in step 8 consist of roughly 6 points, as Voronoi cells of points distributed on a sphere in a nearly uniform way are hexagonal [43]. Algorithm 6 summarizes how Algorithms 2, 3, 4 and 5 may be used together to improve Algorithm 1.

In addition to capturing the coherence between distance queries from neighbouring points on the spheres, there are a number of additional suggestions for capturing the overlap between neighbouring spheres and caching information from previous levels of subdivision. First, by placing spheres of radius σ around voxels with side length σ, there is a significant amount of overlap between spheres of neighbouring voxels. In fact, for a particular voxel at location x, the overlap of spheres around its 6-neighbours covers completely the voxel at location x. Therefore, one would not need to perform any computation at x. Using this trick, one could save half of the computational cost. However, to capture the relationship with object angle, one would have to perform the computation at x as well, although this must only be done at the last level of the recursion. Second, it is wasteful to dismiss the information computed at the earlier levels of recursion in the later levels. Therefore, it would be advisable to use the distance information obtained for a sphere $S_\sigma(x)$ to initialize the distance queries of spheres $S_{\sigma/d}(y)$, where $C_{\sigma/d}(y)$ are cells resulting from subdividing $C_\sigma(x)$

Currently, the algorithm stores the AOF' values in a 3D array corresponding to the bounding box of the mesh. This is highly wasteful, although useful for easy visualization of the results. A much more reasonable solution would be to use oc-

```
Input: Points $P = \{p_i | 1 \leq i \leq N\}$ distributed on the surface of a sphere, the mesh
    boundary $B$
Output: $\nabla D(P)$.
Compute the Delaunay Triangulation $DT$ of points $p_i$
Start at a random point $q \in P$
Compute $\nabla D(q)$ and let the closest triangle on $B$ to $q$ be $tri(q)$
Mark $q$ as visited
Enqueue $(q, tri(q))$ in queue $Q$
while $Q$ is not empty do
    Dequeue $(k, tri(k))$
    for all points $s$ adjacent to $k$ in $DT$ do
        if $s$ is not marked as visited then
            The distance to $s$ is at least the distance to $tri(k)$
            Compute $\nabla D(s)$ and let the closest triangle on $B$ to $s$ be $tri(s)$
            Mark $s$ as visited
            Enqueue $(s, tri(s))$ in $Q$
        end if
    end for
end while
Return $\nabla D(P)$
```

Algorithm 5: ORDEREDPROXIMITYQUERY(P, B)

```
Input: Polygonal mesh boundary $B$, its voxelized solid shape $Inside$ with voxel
    size $\sigma^3$, sphere sampling rate $N$, resolution increment $d$, number of levels of
    subdivision required $levels$, object angle $\theta$.
Output: The voxels of $Inside$ subdivided $levels$ times by $d$ through which passes
    a piece of the medial surface wider than $\sigma/d^{levels}$ with object angle at least $\theta$.
for all voxels $v$ of $Inside$ do
    Let $x$ be the location of $v$
    Sample points $P = \{p_i | 1 \leq i \leq N\}$ on $S_{\sigma/(d^{levels})}(x)$
    $\nabla D(P) \leftarrow$ ORDEREDPROXIMITYQUERY($P$, $B$)
    $AOF(x) \leftarrow 0$
    for all points $p_i$ on the surface of the sphere, $i = 1 \ldots N$ do
        Let $N_S(p_i)$ be the unit outward normal to $S$ at $p_i$
        $AOF(x) \leftarrow AOF(x) + \frac{\nabla D(p_i) \cdot N_S(p_i)}{N}$
    end for
    COMPUTE$AOF'(B, P, \nabla D(P), AOF(x))$ .
end for
if $levels > 0$ then
    $Inside \leftarrow$ HOMOTOPYPRESERVINGDELETE($Inside$, $AOF'$, 0)
    Subdivide the remaining voxels by $d^3$ and add new internal voxels, if any
    COARSETOFINE+$(B, Inside, \sigma, N, d, levels - 1, \theta)$
else
    Return HOMOTOPYPRESERVINGDELETE($Inside$, $AOF'$, $-\frac{1}{8}\sin(\theta)$)
end if
```

Algorithm 6: COARSETOFINE+$(B, Inside, \sigma, N, d, levels, \theta)$

trees to represent our data. Octrees are spatial data structures that allow variable resolution at each spatial location. In 3D, the root of the tree corresponds to a cube bounding the mesh and has d^3 children, where d is the number of pieces into which we divide each dimension of the cube. If a child node contains information that needs to be further resolved, then it is subdivided further, etc. Not only would this data structure save space by only using high resolution cubes for areas that lie inside the mesh, but it would also precisely capture the coarse-to-fine nature of our algorithm. Those sections of the interior the mesh that do not contain a part of the medial surface do not need be resolved. By only subdiving those cells that contain a piece of the medial surface, at each level of subdivision, the amount of storage necessary would be proportional to the surface area of the medial surface. This would provide an enormous improvement in terms of storage. In fact, algorithms that compute the medial surface using recursive spatial subdivision often use octrees [6, 21, 28].

In our experiments, we used the subdivision parameter $d = 3$ to subdivide each voxel by 27 at each step of the recursion. If d is made smaller, it may take many subdivisions to achieve a good approximation to the medial surface, requiring more storage, and more voxels may need to be subdivided each time, requiring more computation. However, depending on the nature of the medial surface, the choice of smaller d may dismiss sections of space at a lower computational cost. On the other hand, if d is made bigger, a good approximation to the medial surface may be achieved after a few subdivisions, requiring less storage, and fewer voxels may need to be subdivided each time, requiring less computation time. On the other hand, the computation at each level of subdivision is more expensive for larger values of d. An investigation into the optimal value of d could potentially allow a speed-up in computation and a reduction in storage.

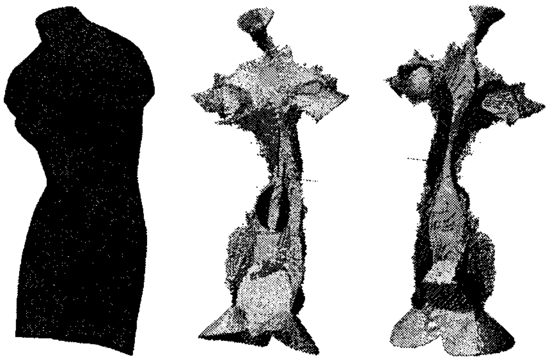


Figure 4.3: The "Venus" mesh (5672 triangles) and its medial surface at a resolution of $177\times129\times361$ voxels: back and front approximated using voxels with AOF' values below -0.15.

4.3 Experimental Results

This section illustrates the performance of Algorithm 1 on a number of examples.

Having already seen the results of applying our algorithm to a simple box shape, whose medial surface in cross-section is shown in Figure 3.7, we now study the behaviour of our algorithm on the more complicated shape of the "Venus" sculpture shown in Figure 4.3. This particular mesh is smooth and fine enough so that by thresholding the internal voxels by AOF' value, we are able to recover the smooth sheets of the medial surface of the smooth sculpture, as shown in Figure 4.3. We applied our algorithm with 3 levels of recursive subdivision, where the last level had resolution of $177 \times 129 \times 361$ voxels. Figure 4.4 illustrates the voxels kept by our algorithm at the 3 stages of recursive spatial subdivision. The sphere sampling is 70, 60 and 50, for the low, middle and high resolutions respectively. The subdivision factor d used was 3. The computational savings of using recursive spatial subdivision for this and other models are captured in Table 4.1. The voxels in the finest resolution are thresholded by their AOF' values to locate the medial sheets having desired object angle. Figure 4.5 demonstrates the effect of increasing this threshold on the set of voxels remaining. We see that many spurious branches

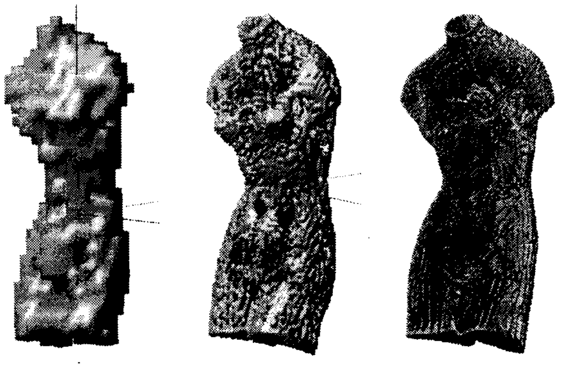


Figure 4.4: Revealing the medial surface of "Venus": voxels that contain medial sheets at the three levels of recursive subdivision

Object	Level 1	Level 2
Box	45.9%	45.5%
Venus	2.1%	12.6%
3-hole torus	3.9%	20.6%
Cow	8.5%	6.3%

Table 4.1: Percentage of object voxels at the highest resolution that were dismissed at each lower level of subdivision.

are present when the threshold is low and the simplification of the medial surface as the threshold is increased. Being able to remove sections of the medial surface in 3D by thresholding alone is very desirable, as much effort is spent on finding a pruning order for the complete medial surface produced by various exact methods [54].

We compare our output with that of some known medial surface algorithms: the PowerCrust [2] and an exact medial surface solver for polyhedra [13]. The output for these two algorithms on the "Venus" mesh is displayed in Figure 4.6 (a) and (b). The input to PowerCrust is the set of vertices of "Venus" and the input to the exact solver is a simpler polyhedron with 250 faces. We also compare against the output of an earlier *AOF* method of [48] that uses an approximate distance transform of a voxelized object in Figure 4.6 (c) and (d). When applied to the vox-

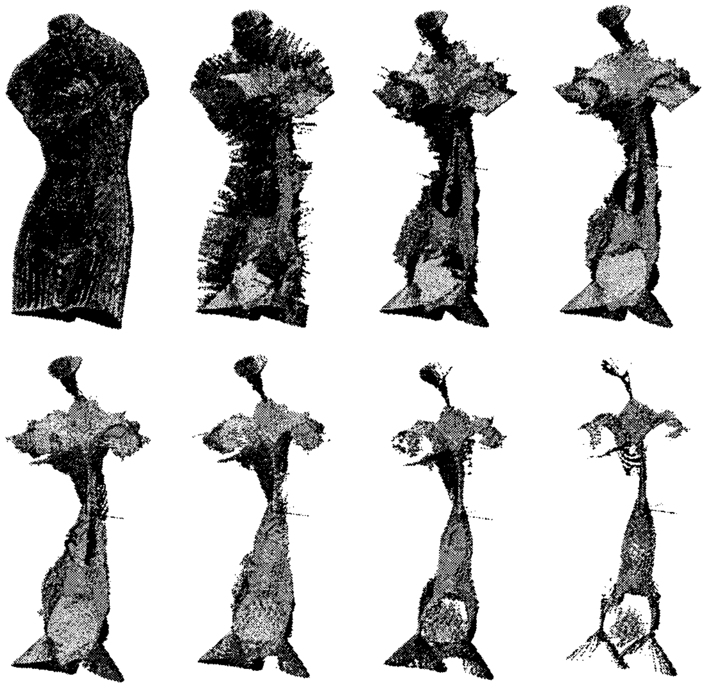


Figure 4.5: Thresholding the high resolution medial surface of "Venus": those voxels having AOF' values below 0, -0.05, -0.1, -0.15, -0.2, -0.25, -0.3, -0.35 (left to right) shown.

elized interior of "Venus" with the highest resolution we considered in our experiments, those voxels whose AOF values are below -0.15 are shown in Figure 4.6 (c). Figure 4.6 (d) demonstrates the effect of using homotopy type preserving voxel deletion. The spurious branches are caused by insufficient smoothness of the voxelized interior of the "Venus" mesh. As a quantitative comparison of the outputs would require the existence of an exact solution, something researchers are not yet able to compute accurately and in reasonable time, we invite the reader to inspect the images in Figure 4.6 qualitatively.

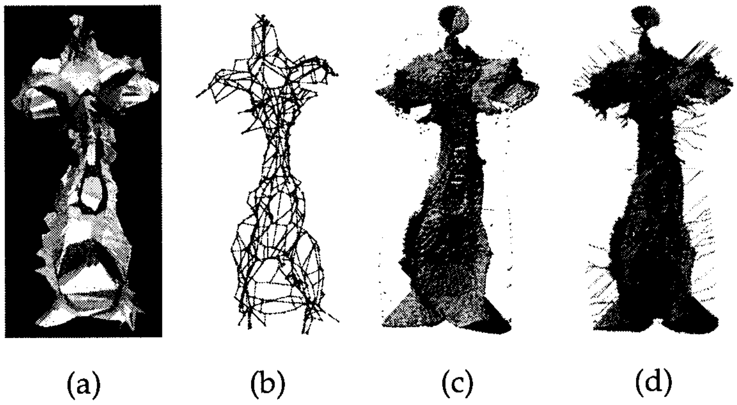


Figure 4.6: The medial surfaces computed by (a) [2] (b) [13] and (c), (d) [48] for "Venus".

We know that thresholding does not always preserve the homotopy type of the medial surface. In Figure 4.7, middle, we see that using thresholding produces a disconnected set of voxels. Applying Algorithm 4 when deleting voxels produces a set of medial surface voxels that preserves homotopy type, shown in Figure 4.7, right. We note, however, that the use of Algorithm 4 is not crucial for objects with complicated topology. In Figure 4.8, we show the medial surface of a 3-hole torus obtained by thresholding. It appears quite smooth and homotopy equivalent to the original object.

We compare our output for the cow and torus models with that of the PowerCrust [2] in Figure 4.9. The input to the PowerCrust is the set of mesh vertices. The output is a triangulation of the "poles": Voronoi vertices that lie on the medial

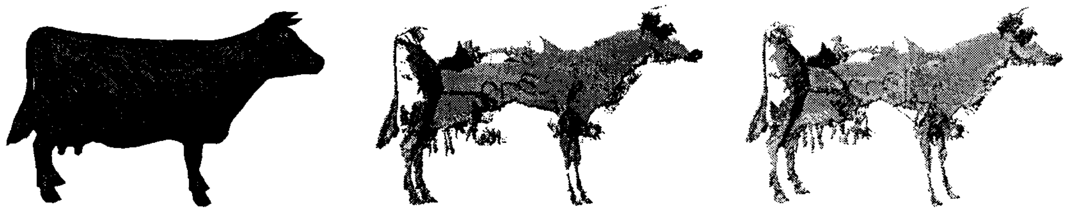


Figure 4.7: A cow mesh (5804 triangles); its $\mathcal{MS}$ with resolution 369 x 227 x 121 voxels obtained by thresholding the AOF' values at -0.2; its $\mathcal{MS}$ obtained using homotopy type preserving deletion.

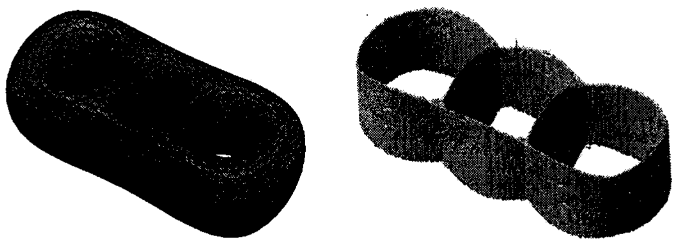


Figure 4.8: A three-hole torus (4000 triangles) and its $\mathcal{MS}$ with resolution 369 x 173 x 128 consisting of those internal voxels whose AOF' is below -0.22.

surface. We see that our results are comparable.

These examples show that our algorithm can yield smooth manifolds due to the fact that the voxel grid can be subdivided to the desired level of resolution. In the following chapter, we shall examine issues of correctness with regard to the finite sphere sampling rate. Specifically, we will argue that the amount of the medial surface potentially missed is bounded in terms one of three parameters.

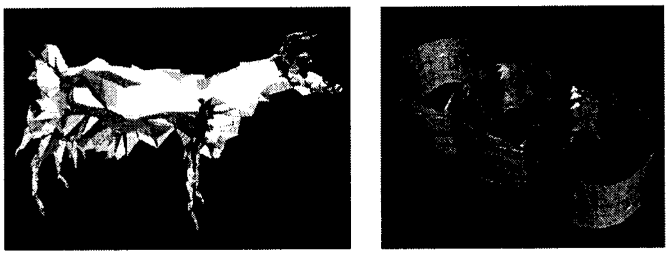


Figure 4.9: The medial surfaces of the 3-hole torus and cow mesh produced by the PowerCrust [2].

Chapter 5

Correctness Issues

So far in our algorithm we have assumed that the sphere sampling rate N is big enough such that Theorem 3.2.17 holds. Specifically, in the design of our algorithm, we have assumed that if the medial surface passes through a voxel $C_\sigma(x)$, then necessarily some of the points sampled on $S_\sigma(x)$ form a one-sided section, so that algorithm DECIDE$\mathcal{MS}$ returns TRUE. However, in certain scenarios, our sphere sampling may not have been fine enough to find a one-sided fan or parallel section. In this chapter, we discuss the relationship between the sphere sampling rate N and the correctness of the algorithm COARSETOFINE+. Specifically, we argue that any sections of the medial surface missed as a result of the finite sphere sampling rate are bounded as a function of N. Therefore, either the information we gathered by finitely sampling the sphere leads us to conclude using Algorithm 3 that no section of the medial surface passes through S or, if a section does pass and we missed it, then there is a bound in terms of one of three parameters on the quality of the medial surface section missed. The arguments presented rely on the reader's intuition in places and a number of claims are left without proof. Nonetheless, we feel that an examination of the relationship between the sphere sampling rate and correctness is important and any kind of progress toward showing this relationship is novel and interesting.

When points $\Phi = \{\phi_i | 1 \leq i \leq N\}$ are sampled on a sphere of radius σ as discussed in Section 4.1.1, to get from point ϕ_k to the next point on the spiral ϕ_{k+1}, we go up $\frac{2\sigma}{N-1}$ to the next horizontal plane and $\frac{3.6\sigma}{\sqrt{N}}$ across along the meridian to ϕ_{k+1}. So the distance between ϕ_k and ϕ_{k+1} is at most $\frac{2\sigma}{N-1} + \frac{3.6\sigma}{\sqrt{N}}$ by the triangle inequality. Let A denote this distance. It is a function of N and σ.

Let $DT(\Phi)$ be the Delaunay Triangulation of Φ. For each Delaunay edge (ϕ_k, ϕ_l) of $DT(\Phi)$ and for each sheet of the medial surface missed, consider 2 cases:

Case 1 The medial surface sheet missed enters $S_\sigma(x)$ between ϕ_k and ϕ_l and exits elsewhere, passing through $C_\sigma(x)$ (refer to Figure 5.1).

Case 2 The medial surface sheet missed enters and exits $S_\sigma(x)$ between ϕ_k and ϕ_l (refer to Figure 5.4).

It may also so happen that the medial surface sheet missed does not intersect any of the Delaunay edges. If this does happen, the sheet must enter and exit S inside the same Delaunay triangle. Let ϕ_k and ϕ_l be the vertices of such a Delaunay triangle having the largest length edge. For an upper bound on the quality of the medial sheet missed, shift the medial sheet such that it enters and exits between ϕ_k and ϕ_l. Now Case 2 applies.

We consider each case separately.

5.1 Medial Sheet Leaves

The distance between ϕ_k and ϕ_l, two vertices of a Delaunay edge, is at least A. Using A and the length B of the medial surface segment missed through S we show that either the "width" of the boundary feature whose medial surface is missed is small or the object angle of points on the missed medial surface section is small. There are two subcases, illustrated by Figure 5.1:

Case 1(a) We missed a sheet of the medial surface with arbitrary object angle points generated by a "narrow" feature.

Case 1(b) We missed a sheet of the medial surface with small object angle points generated by a "wide" feature.

For a pair of points ϕ_k and ϕ_l, consider the point ϕ_m located between these two points along the great circle connecting them. Let $N_S(\phi_m)$ be the outward normal to S at ϕ_m. Call F the feature composed of those 2 boundary elements (vertices, edges, faces) whose bisector is the medial surface sheet missed in Case 1.

Definition 5.1.1. *The **width** of F is the greatest perpendicular distance to between points on both of the boundary elements whose bisector is part of the medial surface of the shape $N_S(\phi_m)$. Let w be the line segment connecting those points along which the width is measured, or **width segment**.*

Next we need a definition of distance from the sphere S to the feature F missed.

Definition 5.1.2. *The distance **d** from the sphere S to the feature F with width segment w is $\chi(S, w) dist(\phi_m, w)$, where $\chi(S, w) = -1$ if w intersects S and 1 otherwise and $dist(\phi_m, w)$ is the smallest Euclidean distance from ϕ_m to w.*

We want to find the maximum possible feature width w and object angle θ of the medial surface section missed given the sphere sampling rate N and sphere radius σ.

Let B be the length of the medial surface sheet through S, extending the sheet beyond any junctions with other sheets, measured as the shortest distance between

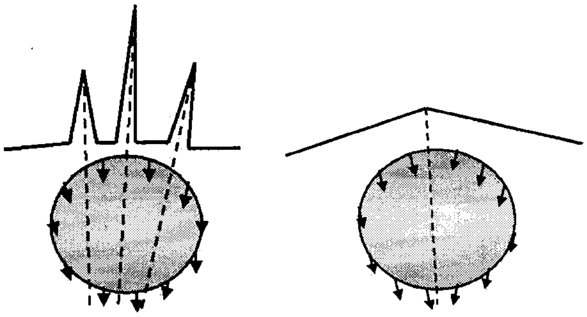


Figure 5.1: Missing sections of the medial surface due to a finite sphere sampling rate: Case 1(a) and Case 1(b)

the points of entry and exit of the medial surface sheet into S. We assume that the medial surface sheet missed enters S normal to ϕ_m (WLOG, because of our use of perpendicular distance). Also, the maximum value of object angle and boundary feature width is obtained when the point of entry into S of the medial surface section missed is at ϕ_m. Then the relationship between A, w, d, B and object angle θ that gives the maximum value of w and θ is as in Figure 5.2. From Figure 5.2, the object angle of the missed medial surface sheet varies from θ' to θ. We solve for w, θ and θ' given A, B and d as follows:

$$w = A \cdot \frac{B+d}{B}$$
$$\theta = \arctan(\frac{w}{2d})$$
$$\theta' = \arctan\left(\frac{w}{2(B+d)}\right).$$

Recall from Section 3.2.3 that the medial surface passes through a voxel $C_\sigma(x)$ if a planar section of the medial surface passing through $S_\sigma(x)$ passes at least through the corner of $C_\sigma(x)$ where it has (minimum) diameter $\frac{\sigma}{2}$. In practice, the maximum value of A, $\frac{2\sigma}{N-1} + \frac{3.6\sigma}{\sqrt{N}}$, given by the spiral sphere sampling is too much of an overestimate. When $N = 100$, this value is 0.3802. Experimentally, we find that for 100

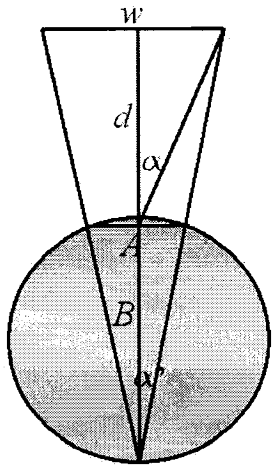


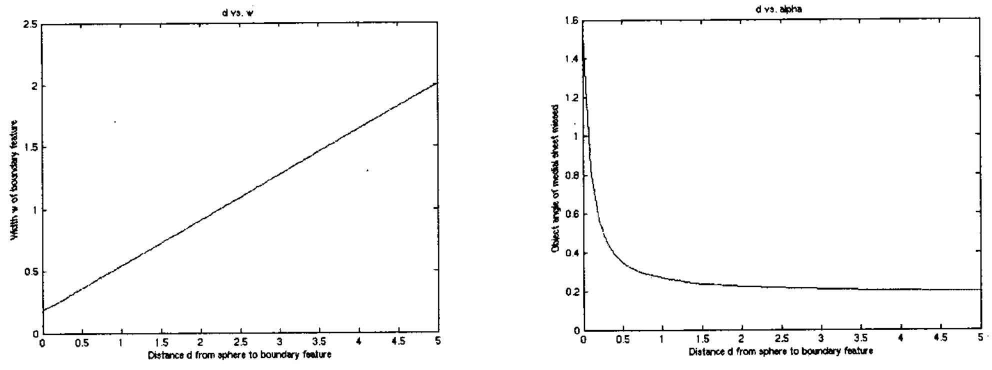
Figure 5.2: Relationship between $A, w, d, B,$ and θ

points distributed in the spiral fashion on the sphere, the greatest shortest distance between points on a unit sphere is 0.1824. For this value of A and values $\sigma = 1$ and $B = \frac{\sigma}{2}$ [1], Figure 5.3 shows the relationship between d and w, and d and θ.

In case 1(a), θ is the maximum object angle of the medial surface section missed. In case 1(b), θ' is this object angle. As we observe from Figure 5.2, this angle is always smaller than θ and so in finding the upper bound on object angle and width of the boundary feature missed, we shall only be concerned with finding the values of θ.

We observe that for values of d below 0.5, while the object angle of the boundary feature is large, the boundary feature width w is fairly small. For values of d above 0.5, this width becomes large, while the object angle θ falls off to about 0.2 radians.

[1] Whenever we consider a curved section of $\mathcal{MS}$, it appears that B may be smaller than $\frac{\sigma}{2}$. However, in our discussion we have not investigated the relationship between A and the maximum allowed curvature of the medial sheets missed. If the medial sheets are allowed to be maximally curved, then $B \geq 2 \cdot (1 - \frac{\sqrt{3}}{2})\sigma \approx 0.2679\sigma$, but such a situation is not allowed, for otherwise it would not be possible to pair up ∇D values at sampled points in Algorithm 3. However, using this bound on B reveals a similar relationship between A, B, w, d and θ.



(a) (b)

Figure 5.3: For $A = 0.1824$, relationship between distance d and (a) width w and (b) object angle θ.

For N larger, the object angle falls off to 0 radians as d gets large fairly quickly.

Together, these relationships can be interpreted to mean that for a given N and σ, either the width of the boundary feature whose bisecting medial surface sheet is missed is small or the object angle of the medial surface section missed is bounded. For case 1, we may conclude that the error in these two parameters due to the sphere sampling rate is bounded in terms of N and σ.

5.2 Medial Sheet Takes a Dip and Comes Back

An example of this case is illustrated by Figure 5.4. Unlike case 1, here the medial surface can have very high object angle and arbitrary boundary feature width. However, in this case, we miss at most a section of the medial surface that has width A in cross-section.

We will show bounds on how far the medial surface "dips" inside S. We shall consider the 2D case of the cross-section in the plane of $N_S(\phi_m)$.

Definition 5.2.1. *The dip of a medial surface section missed between neighbouring points*

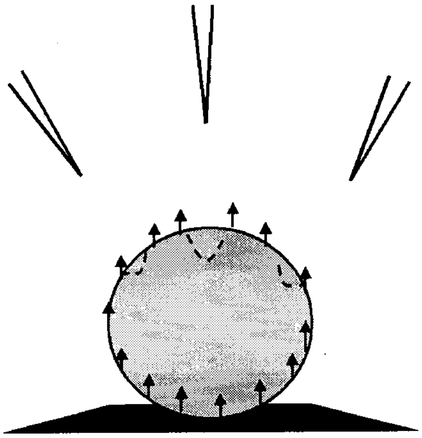


Figure 5.4: Missing sheets of the medial surface due to a finite sphere sampling rate in Case 2

ϕ_k and ϕ_l of the sphere sampling is the maximum perpendicular distance from the line segment connecting ϕ_k and ϕ_l and the medial surface section missed.

It is known that the bisector of a point and a line in the plane is a parabola. We derive the equation again here. Consider a point p located in the Euclidean plane at $(1, 0)$ and the line $L : y = -1$ as in Figure 5.5. We want to find the bisector B of this point p and line L; that is, given the x-coordinate x on the bisector B, we want to find the equation for the y-coordinate y in terms of x. Solving for y, we get:

$$\begin{aligned} dist(p,B)^2 &= dist(L,B)^2 \\ x^2 + (1-y)^2 &= (1+y)^2 \\ \frac{x^2}{4} &= y. \end{aligned}$$

Therefore, the equation of the bisector between p and L is $y = \frac{x^2}{4}$. We want to use this equation to obtain the amount of dip given the width of the parabola A.

Setting $x = \frac{A}{2}$, we solve for y:

$$\begin{aligned} \frac{(A/2)^2}{4} &= y \\ \frac{A^2}{16} &= y. \end{aligned}$$

This amount y is the amount of dip of the parabola below the line segment (ϕ_k, ϕ_l). Next, we show that $\frac{A^2}{16}$ represents the maximum possible dip given A.

We argue that to obtain the maximum dip of the medial surface section missed due to the finite sphere sampling (1) a planar cross-section of the medial surface section missed must be parabolic and (2) the greatest dip of the parabola is achieved when the segment (ϕ_k, ϕ_l) passes perpendicular to its line of symmetry, i.e. in cross-section, the medial surface missed looks exactly like the parabola in Figure 5.5.

Consider the missed section of the medial surface in cross-section. We are now in the 2D case. Here the medial surface is either planar (edge-edge, vertex-vertex bisector) or parabolic (vertex-edge bisector). In the planar case, the dip is 0. The greatest dip is achieved when the medial surface is parabolic. When segment (ϕ_k, ϕ_l) passes perpendicular to the parabola's line of symmetry, the apex of the parabola gives the greatest dip. As the curvature of the parabola at the apex is the greatest, this dip is the maximum overall.

We have shown that the amount of dip of the missed medial surface sections is bounded for the case where this medial surface section both enters and exits $S_\sigma(x)$ between ϕ_k and ϕ_l as a function of N and σ. For the case where the missed medial surface section enters but does not exit between ϕ_k and ϕ_l and intersects the interior of $C_\sigma(x)$, we have shown that either the object angle of the medial surface section missed is small or the width of the boundary feature whose medial bisector is missed is small.

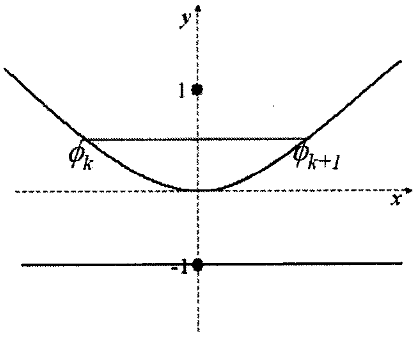


Figure 5.5: The bisector between a point and a line is a parabola.

As these cases cover all the possibilities of how a missed medial surface sheet may have passed through $C_\sigma(x)$, we conclude that either the object angle of the medial sheet, the boundary feature width, or the medial sheet's dip is bounded in terms N and σ. One may adjust the values of N to obtain a more exact approximation in terms of these three parameters.

Chapter 6

Conclusions and Future Work

Our goal in this thesis has been to design an efficient and correct algorithm that computes an arbitrarily precise approximation to the location of the medial surface of a polyhedron. We have extended a continuous theory based on the measure of average outward flux in a region of zero size [48, 17] to a theory applicable to non-zero sized domains. We have applied this extended theory to design a coarse-to-fine algorithm that reveals, at each scale, more precise locations of the medial surface of a polyhedron. The output is a set of voxel locations that is at most 2 voxels thick approximating the true medial surface. Our algorithm is able to preserve the homotopy type of the original object if a sufficiently fine resolution of the approximate medial surface is requested. Further, this medial surface may be properly simplified based on thresholding the voxels by their AOF' values alone, demonstrating the descriptive strength of this measure. We have made an effort to study the effect of discrete computations on the correctness of our algorithm that has lead us to believe that the medial surface sections missed as a result of computations at a finite number of locations is bounded.

This thesis presents an extension to the work in [18, 8]. The first of these, [18], develops the continuous theory presented in Chapter 3 and applies it in a discrete setting to 2D shapes. The second, [8], is an extension of the work in [18] to 3D voxelized objects. None of these earlier studies considers the correctness issues when average outward flux measures are applied in a region of non-zero size. We hope that our contribution of a provably complete algorithm for polyhedra compliments the contributions of [18, 8].

Next, we outline avenues for future work that allow improvement on the work presented in this thesis, extension of this work, or its application in a new context.

- To make our approach practical, the algorithm needs to be more efficient, in terms of computation time and storage. The suggestions of Section 4.2 need to be implemented and experimental evidence needs to be gathered to justify them. Further, use of a different software than PQP [27] for computing the distance between a point and a mesh, as well as a tailored version of this software to our particular problem, need to be considered. Finally, the use of GPUs for fast geometric computations is becoming very popular [53]. To stay current with developing trends, continuing research would need to investigate the merits of this kind of processor for these purposes.
- To make this work significant in a theoretical setting, the arguments of Chapter 5 need to be formalized and all claims proved. Furthermore, it is important to establish a relationship with the sphere sampling rate N and the error in the estimate of the object angle given by AOF' through a sphere.
- We would like the bounds on object angle given by Proposition 3.2.20 to be tighter. This can be done by increasing the radii of the spheres fitted around voxels. However, in this case, too much wasteful computation is done for computing the average outward flux through the large spheres. Also, the width of the medial surface that would guarantee the upper bound would need to increase. It is possible that, by combining information obtained at

different scales, we may learn more about the true object angle and the width of the medial surface passing through a voxel.

- Also, it may be possible to discover the minimum and maximum object angles of points on medial sheets inside a particular voxel by a direct examination of the grassfire flow vector field through that voxel, or to at least obtain bounds on these angles.
- The polyhedra we have considered have been approximations of smooth objects. The medial surface of smooth objects is composed of smooth manifolds. If we can determine the quality of the polyhedral approximation to the smooth shape and infer from it the object angle of medial sheets corresponding to the polyhedron but not the smooth shape, we may simplify the medial surface to obtain the medial surface of the smooth shape.
- The approximation to the medial surface we have computed is composed of voxels. Smoothing this set of voxels or using a special subvoxel refinement scheme similar to [18], may better approximate the true locations of the smooth medial sheets.
- It would also be highly beneficial to be able to segment the medial surface into its constituent sheets. This can be done using digital classification, as in [8], although prior smoothing may be required. Further, in areas where the connectivity information is not apparent, we could increase the voxel resolution to discover the connectivity.
- It has been recently shown by the James Damon [14, 15] that the differential geometry of an object and its boundary can be studied using the differential geometry of its medial surface. Specifically, there exists a mapping between a special shape operator on the medial surface and that of the boundary. To make computational use of this relationship, it is necessary to have a well defined tangent plane to the medial surface. This tangent plane needs to

vary smoothly, and so, a smoother approximation to the medial surface than the voxelized approximation we have proposed is required. Furthermore, at each (smooth) medial point, we need to be able to distinguish two vectors that connect that point to their two closest boundary points. As we want to deal with smooth points only, it is necessary to have a segmentation of the medial surface into sheets before proceeding.

It is our hope that work on finding medial surfaces using the average outward flux of the vector field induced by the grassfire flow continues so that the problems posed above, and others, may be solved. Medial representations are finding more and more applications, from shape analysis and retrieval to mesh generation, medical image analysis, bioinformatics, and motion planning. The need for correct and efficient algorithms to compute this representation is a pressing one, and we close this thesis with a wish for its proximate fulfilment.

Bibliography

[1] D. Attali, A. Montanvert. Computing and simplifying 2D and 3D semicontinuous skeletons of 2D and 3D shapes. *CVIU*, 67(3): 261-273, 1997.

[2] N. Amenta, S. Choi, R. K. Kolluri. The power crust. *ACM Symposium on Solid Modeling and Applications*, 249-260, 2001.

[3] N. Ameta, S. Choi, R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2,3): 127-153, 2001.

[4] S. Arya, D. Mount, N. Netanyahu, R. Silverman, A. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45: 891-923, 1998.

[5] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger. The r*-tree: An efficient and robust access method for points and rectangles. *SIGMOD Conference on Management of Data*, 322-331, 1990.

[6] I. Boada, N. Coll, N. Madern, J. A. Sellarès. Approximations of 3D generalized Voronoi diagrams. *EWCG2005*, 2005.

[7] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27: 321-345, 1984.

[8] S. Bouix. *Medial surfaces*. Ph.D Thesis, School of Computer Science, McGill University, 2003.

[9] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38: 205-287, 1973.

[10] H. Blum, R. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10: 167-180, 1978.

[11] H. I. Choi, S. W. Choi, H. P. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1): 57-88, 1997.

[12] R. Courant, H. Robbins, I. Stewart. *What is mathematics?* Oxford: Oxford University Press, 1996.

[13] T. Culver, J. Keyser, D. Manocha. Exact computation of the medial axis of a polyhedron. *Computer Aided Geometric Design*, 21(1): 65-98, 2004.

[14] J. Damon. Global geometry of regions and boundaries via skeletal and medial integrals. Preprint.

[15] J. Damon. Determining the geometry of boundaries of objects from medial data. *IJCV*, 63(1):45-64, 2005.

[16] T. K. Dey, W. Zhao. Approximate medial axis as a Voronoi subcomplex. *Symposium on Solid Modeling and Applications*, 356-366, 2002.

[17] P. Dimitrov, J. N. Damon, K. Siddiqi. Flux invariants for shape. *CVPR*, 2003.

[18] P. Dimitrov. *Flux invariants for shape.* M.Sc. Thesis, School of Computer Science, McGill University, 2003.

[19] H. Du, H. Qin. Medial axis extraction and shape manipulation of solid objects using parabolic PDEs. *Symposium on Solid Modeling and Applications*, 2004.

[20] S. Ehmann, M. Lin. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Eurographics*, 20(3), 2001.

[21] M. Etzion, A. Rappoport. Computing the Voronoi diagram of a 3-D polyhedron by separate computation of its symbolic and geometric parts. *Symposium on Solid Modeling*, 167-178, 1999.

[22] M. Etzion, A. Rappoport. Computing Voronoi skeletons of a 3-D polyhedron by space subdivision. *Computational Geometry: Theory and Applications*, 21:87-120, 2002.

[23] M. Foskey, M. C. Lin, D. Manocha. Efficient computation of a simplified medial axis. *Symposium on Solid Modeling and Applications*, 96-107, 2003.

[24] S. Gottshalk, M. Lin, D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *SIGGRAPH*, 171-180, 1996.

[25] A. Guériec. Meshsweeper: Dynamic point-to-polygonal mesh distance and applications. *IEEE Transactions on Visualization and Computer Graphics*,7(1), 2001.

[26] M. Held, J. Klosowski, J. Mitchell. Evaluation of collision detection methods for virtual reality fly-throughs. *CCCG*, 205-210, 1995.

[27] E. Larsen, S. Gottschalk, M. C. Lin, D. Manocha. Fast proximity queries with swept sphere volumes, Technical Report TR99-018, Department of Computer Science, University of N. Carolina, Chapel Hill, 1999.

[28] D. Lavender, A. Bowyer, J. Davenport, A. Wallis, J. Woodwark. Voronoi diagrams of set-theoretical solid models. *IEEE Computer Graphics and Applications,* 12(5): 69-77, 1992.

[29] F. Leymarie, B. Kimia. Computation of the shock scaffold for unorganized point clouds in 3D. *CVPR,* 821-827, 2003.

[30] F. Leymarie, B. Kimia. The shock scaffold for representing 3D shape. *IWVF4,* 2001.

[31] A. Lieutier. Any open bounded subset of $\mathbb{R}^n$ has the same homotopy type than its medial axis. *ACM Symposium on Solid Modeling and Applications,* 65-75, 2003.

[32] G. Malandain, G. Bertrand. Topological segmentation of discrete surfaces. *CVPR,* 444-449, 1991.

[33] V. Milenkovic. Robust construction of the Voronoi diagram of a polyhedron. *5th Canadian Conference on Computational Geometry,* 473-478, 1993.

[34] M. Näf, O. Kübler, R. Kikinis, M. Shenton, G. Székely. Characterization and recognition of 3D organ shape in medial image analysis using skeletonization. *Workshop on mathematical methods in biomedical image analysis,* 139-250, 1996.

[35] J. Mitchell, J. O'Rourke. Computational geometry column 42. *International Journal of Computational Geometry Applications,* 11(5):573-582, 2001.

[36] U. Neisser. *Cognition and reality.* Freeman Press, 1976.

[37] J. O'Rourke. *Computational geometry in C.* Cambridge: Cambridge University Press, 1998.

[38] S. M. Pizer, K. Siddiqi, G. Székely, J. N. Damon, S. W. Zucker. Multiscale medial loci and their properties. *IJCV,* 55(2-3): 155-179, 2003.

[39] M. Proust. *In search of lost time.* 1927.

[40] E. Rakhmanov, E. Saff, Y. Zhou. Minimal discrete energy on the sphere. *Mathematical Research Letters,* 1:647-662, 1994.

[41] J. Reddy, G. Turkiyyah. Computation of 3D skeletons using a generalized Delaunay triangulation technique. *Computer Aided Design,* 27(9):677-694, 1995.

[42] R. Renka. Algorithm 772: STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere. *ACM Transactions on Mathematical Software,* 23(3):416-434, 1997.

[43] E. B. Saff, A. B. J. Kuijlaars. Distributing many points on a sphere. *Mathematical Intelligencer,* 19(1):5-11, 1991.

[44] M. Schmitt. Some examples of algorithms analysis in computational geometry by means of mathematical morphological techniques. *Workshop on Geometry and Robotics*, LNCS, Springer-Verlag, 391: 225-246, 1989.

[45] D. Sheehy, C. Armstrong, D. Robinson. Shape description by medial surface construction. *IEEE Transactions on Visualization and Computer Graphics*, 2(1): 62-72, 1996.

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

[47] K. Siddiqi, S. Pizer. *Medial representations: mathematics, algorithms and applications*, Springer, 2006 (to appear).

[48] K. Siddiqi, S. Bouix, A. R. Tannenbaum, S. W. Zucker. Hamilton-Jacobi skeletons. *IJCV*, 48(3): 215-231, 2002.

[49] K. Siddiqi, S. Bouix, A. Tannenbaum, S. W. Zucker. The Hamilton-Jacobi skeleton. *ICCV*, 1999.

[50] J. Snoeyink. Chapter 30: Point location. *Handbook of Discrete and Computational Geometry*, J. Goodman and Joseph O'Rourke, eds., Boca Raton: CRC Press LLC, 559-574, 1997.

[51] S. Stolpner, K. Siddiqi. Revealing significant medial structure in polyhedral meshes. *3DPVT*, 2006.

[52] A. Sud, M. Foskey, D. Manocha. Homotopy-preserving medial axis simplification. *Symposium on Solid and Physical Modeling*, 39-50, 2005.

[53] A. Sud, M. Otaduy, D. Manocha. DiFi: Fast 3D distance field computation using graphics hardware. *Eurographics*, 23(3), 2004.

[54] R. Tam, W. Heidrich. Shape simplification based on the medial axis transform. *IEEE Visualization*, 481-488, 2003.

[55] J. Vleugels, M. Overmars. Approximating generalized Voronoi diagrams in any dimension. Technical Report UU-CS-1995-14, Department of Computer Science, Utrecht University, 1995.