3-D Shape Interrogation by Medial Axis Transform

by

Evan Conway Sherbrooke

S.M. Electrical Engineering and Computer Science, M.I.T., October 1993
S.M. Naval Architecture and Marine Engineering, M.I.T., October 1993
S.B. Mathematics, M.I.T., June 1990

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

April 1995

© Massachusetts Institute of Technology 1995. All rights reserved.

Author .....
Department of Ocean Engineering
April 25, 1995

Certified by .....
Nicholas M. Patrikalakis
Associate Professor
Thesis Supervisor

Accepted by .....
A. Douglas Carmichael
Chairman, Departmental Committee on Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUL 28 1995

3-D Shape Interrogation by Medial Axis Transform

by

Evan Conway Sherbrooke

Submitted to the Department of Ocean Engineering
on April 25, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The *Medial Axis Transform* is a representation of an object which has been shown to be useful in design, interrogation, animation, finite element mesh generation, performance analysis, manufacturing simulation, path planning, and tolerance specification. In this thesis, the theory of the Medial Axis Transform for 3-D objects and an algorithm to compute such transforms are considered. For objects with piecewise C^2 boundaries, relationships between the curvature of the boundary and the position of the Medial Axis are developed. For n-dimensional submanifolds of $\Re^n$ with boundaries which are piecewise C^2 and completely G^1, a deformation retract is set up between each object and its Medial Axis, which demonstrates that if the object is path connected, then so is its Medial Axis. In addition, it is proven that path connected polyhedral solids without cavities have path connected Medial Axes. An algorithm for determining the Medial Axis Transform is developed first for convex 3-D polyhedral solids and then extended to general 3-D polyhedral solids of arbitary genus without cavities, with non-convex vertices and edges. The algorithms are based on a classification scheme which relates different pieces of the Medial Axis to one another even in the presence of degenerate Medial Axis points. Vertices of the Medial Axis are connected to one another by tracing along adjacent edges, and finally the faces of the Axis are found by traversing closed loops of vertices and edges. Systems of governing equations for different types of Medial Axis points are established using the classification scheme, and the computation of solutions of these systems is discussed. Representation of the Medial Axis and associated radius function is addressed, and pseudocode for the algorithms is given along with recommended optimizations. A connectivity theorem is proven to show the completeness of the algorithm. A further extension of the algorithm to objects with curved boundaries is also outlined. Complexity estimates and stability analysis for the polyhedral algorithms are presented. Finally, examples illustrate the computational properties of the algorithm for convex and non-convex 3-D polyhedral solids with polyhedral holes.

Thesis Supervisor: Nicholas M. Patrikalakis
Title: Associate Professor

Acknowledgments

This thesis is dedicated to the many people who have made it possible. First to my parents, who have provided love, support, and inspiration during my life. They also produced me, for which I will always be grateful. Secondly to my brother, who has always been a model in finding a balance between schooling and other activities. In my family there has always been a creative version of the traditional work ethic which has always been an inspiration.

During my tenure in the Ph. D. program, several people and organizations I have been in contact with have provided tremendous support. First and foremost, my advisor, Professor N. M. Patrikalakis, has provided patient guidance throughout the entire process, and even before, when I was an undergraduate researcher. His tolerance for my rather unusual work habits has enabled the work to proceed smoothly. Second, Dr. H. N. Gursoy, whose thesis dealt with the 2-D Medial Axis Transform, provided initial inspiration. The other members of my thesis committee, Dr. E. Brisson, Professor E. Getzler, and Dr. X. Ye provided useful, detailed comments on the manuscript, which I greatly appreciate. Also, Professor D. C. Gossard gave useful comments in the early phases of this work. Professor F.-E. Wolter was of great help in critiquing the mathematics in Chapter 3 underlying the Medial Axis Transform. Dr. J. R. Rossignac made useful comments concerning the offsets of polyhedral solids. Prof. P. D. Sclavounos assisted in analyzing some of the properties of the Singular Value Decomposition. Prof. L. J. Guibas and Prof. G. T. Toussaint provided information about the literature.

During my tenure as a graduate student, several co-workers helped in various ways; I would like especially to acknowledge Mr. S. L. Abrams, Mr. M. S. Drooker, Mr. C.-Y. Hu, Dr. T. Maekawa, Dr. S. T. Tuohy, and Ms. J. Zhou. Mr. Abrams in particular helped generate examples for the thesis, for which I am extremely grateful. I am also indebted to the NSF Graduate Fellowship program which has made it possible for me to find food, and to enjoy independence in my work.

In addition, acknowledgments are due to the various people and organizations who kept me sane during the process; in particular, the Chorallaries, Testostertones, the volleyball people, residents of Random, certain Musical Theater Guild people, and my housemate, A. Velez, who didn't do anything thesis-related, but did install my CD-ROM drive. Finally, to Erin, whose love and support have made the end of this process painless.

This work was supported, in part, by the M.I.T. Sea Grant College Program and the Office of Naval Research under grant numbers NA90AA-D-SG-424; N00014-91-J-1014 and N00014-94-1-1001; and by an NSF Graduate Fellowship. Earlier work on nonlinear polynomial solvers on which the branch point computation is based was supported, in part, by the M.I.T. Sea Grant College Program; the Office of Naval Research; and the National Science Foundation under grant numbers NA90AA-D-SG-424; N00014-91-J-1014; DMI-9215411, and IRI-9224640; and by an NSF Graduate Fellowship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Outline

1.1 Shape Representation and Interrogation by MAT

Not too long ago, the design of even simple objects was complex and cumbersome. Starting with a rough design, a model of the object would be built and subjected to various tests; based on the results of these tests, modifications would be made to the initial design, and the process would begin again. Thanks to modern computer systems, much of the modeling and analysis may be performed *inside* the computer using software that integrates concept generation, modeling, and analysis capabilities into a comprehensive whole. Of course, increased capability leads to increased complexity, and one of the major problems modern design systems have to handle is data representation. To be useful for all phases of the design process, an integrated design system must be accurate as well as easy to manipulate. With these criteria in mind, most systems have selected either a *Boundary Representation* (B-Rep) or a *Constructive Solid Geometry* (CSG) approach to modeling three-dimensional objects.

In this thesis, we study a third alternative, based on the *Medial Axis Transform* (MAT) (or skeleton, as it is sometimes called). The Medial Axis (MA) of an object is the locus of centers of balls which are maximal within the object, together with the limit points of this locus (a ball is maximal within an object if it is contained in the object but is not a proper subset of any other ball contained in the object). The MAT is the Medial Axis together with the associated *radius function* which is the radius of the maximal ball around any given point on the MA. The MA of a 2-D bracket is shown in figure 1-1. The maximal disc associated with a point at the intersection of three MA branches is also shown. Originally

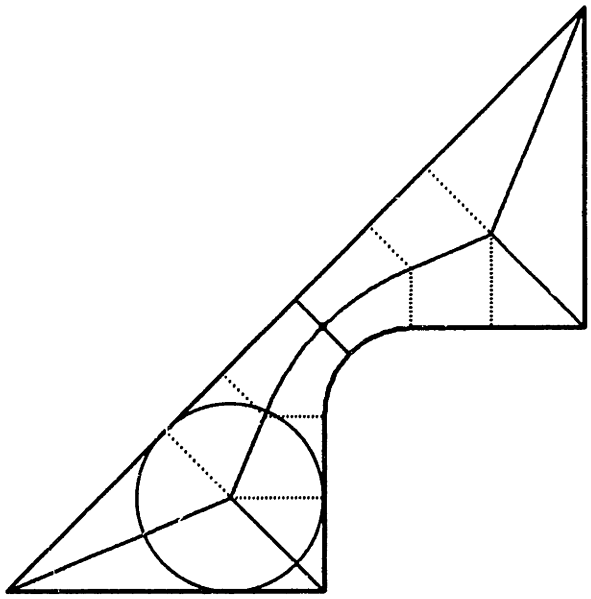

Figure 1-1: A 2-D bracket and its Medial Axis. A maximal disc on the Axis is also shown.

proposed by Blum [6] [5], the MAT has been developed extensively since then. It has several properties which neither B-Rep nor CSG directly provide. First, because it elicits important *symmetries* of an object, it facilitates the design and interrogation of symmetrical objects [5]. Second, the MAT exhibits *dimensional reduction* [19]; for example, it transforms a 3-D solid region into a connected set of points, curves, and surfaces, along with an associated *radius function* described in more detail below. Third, once a region is expressed with the MAT, the skeleton and radius function themselves may be manipulated, and the boundary will deform in a natural way, suggesting applications in computer animation. Fourth, the skeleton may be used to facilitate the creation of coarse and fine finite element meshes of the region [35] [36] [37] [38] [61] [80] [1] [83] [66]. Fifth, the MAT determines constrictions and other global shape characteristics that are important in mesh generation, performance analysis, manufacturing simulation, and path planning [59] [61]. Sixth, the MAT can be used in document encoding [15] [14] and other image processing applications [10]. Finally, the MAT may be useful in tolerance specification [40].

Up to this point, work on the Medial Axis Transform and its applications has been somewhat limited by the difficulty of developing an algorithm which is robust, accurate, and efficient to carry out the transformation. Most recent work on the 3-D problem has tried to generate a discrete approximation to the actual MAT; while such algorithms may be well-suited for some meshing applications, particularly if they involve some degree of

user interaction, they are less satisfactory for modeling since they do not capture the topological structure of the shape accurately. However, because the skeleton is typically path connected, and *because branch points on the skeleton may be expressed as solutions of a set of simultaneous nonlinear polynomial equations*, we believe that a *continuous* approach to the problem is both possible and practical.

1.2 Objectives of this Thesis

Because the problem of determining the Medial Axis Transform of 3-D objects is so little understood, there are several objectives which we must pursue. First and foremost, the problem must be formulated mathematically, and the theory surrounding the Medial Axis and its associated sets must be developed. Second, we must design an algorithm for determining the Medial Axis for some class of 3-D objects. Finally, we need to demonstrate how to extend the algorithm to a more general class of objects, and investigate the problems that might arise.

Our first objective, the development of the mathematics surrounding the Medial Axis, requires a consistent set of definitions which consolidates several disparate concepts in the literature. Accordingly, Chapter 2 presents a set of definitions which attempt to be cohesive while remaining as true to the literature as possible. Once a mathematical framework is established, various theorems concerning the Medial Axis and its related sets may be proven.

For our second objective, we design an algorithm for one of the simplest classes of 3-D shapes, the convex polyhedral solids. As we shall see, even this class exhibits some interesting characteristics and degeneracies which must be dealt with in our algorithm. A classification scheme critical to the algorithm is also developed, and a related scheme turns out to be applicable to a far more general class of objects. Since the convex polyhedral solids are overly simplistic, the algorithm is later extended to handle general polyhedral solids of arbitrary genus (a finite number of holes may pierce through the solid) without cavities (volumes subtracted from the interior of the solid), with non-convex vertices and edges. The resulting algorithm is significantly modified to handle the new curved components of the Medial Axis which arise. Both algorithms are implemented and tested in a variety of examples.

Our last objective combines the first two to postulate a continuous algorithm for ob-

jects with curved boundaries. Although we did not implement such a generally applicable algorithm, we believe that it is theoretically feasible and present a template as a starting point for future work. We also try to identify areas of concern for both mathematician and computer scientist in the extension of this work.

1.3 Organization of this Thesis

1.3.1 Definitions and Review

Chapter 2 begins with a list of definitions, not merely of the MAT and its related sets, but also of some topological terms we use throughout the thesis. In addition, there is a section which briefly covers the notation used. Following the definitions is a review of the existing literature.

1.3.2 Mathematical Properties of the MAT

Because the theoretical understanding of the three-dimensional MAT is relatively incomplete, we address a number of mathematical issues in Chapter 3 which have not until now been addressed in depth in extant literature. Because our algorithms require the Medial Axis to be connected in order to work, this chapter also shows that in a wide number of shapes, including polyhedra, the Medial Axis is in fact connected.

1.3.3 Point Classification

In an attempt to reduce complexity, we describe a Medial Axis point classification scheme in Chapter 4 which enables us to trace out skeleton branches. Our scheme represents an extension of a scheme due to Brandt [11].

1.3.4 An Algorithm for Determining the MAT of Convex 3-D Polyhedral Solids

In Chapter 5, we use our classification scheme to develop an algorithm to construct the MAT of convex 3-D polyhedral solids. Our method is similar in spirit to that of Hoffmann [42] but is more closely related to the connectivity of the skeleton. The complexity and robustness of this algorithm are analyzed, and examples of the method are also given.

1.3.5 An Algorithm for Determining the MAT of General 3-D Polyhedral Solids

Chapter 6 covers the extension of the algorithm to general 3-D polyhedral solids of arbitrary genus without cavities, with non-convex vertices and edges. Because, as we shall see, parts of the Medial Axis of such polyhedra may be curved, we describe how these parts may be traced. An analysis of the algorithm and examples are also given.

1.3.6 Extension of the Algorithm to Solids with Curved Boundaries

Chapter 7 outlines how the algorithm might be extended to more complex 3-D solid objects with curved boundaries. Also, possible problems in extending the algorithm are identified in this chapter.

1.3.7 Conclusions and Recommendations

Chapter 8 is the final chapter of this thesis, containing conclusions and a summary of the main contributions of this work. Recommendations for further research are also presented.

1.3.8 Offset Sweeps of Simple Geometric Objects

A short appendix is provided, which gives implicit equations for the offset sweeps of points, lines, and planes. These implicit equations are used in the algorithm for general 3-D polyhedral solids to trace out pieces of the MA.

1.3.9 Subdividing Multiply Connected Polygonal Regions

The final appendix in this thesis discusses the subdivision of multiply connected polygonal regions in the plane into a set of simply connected regions, which is a preprocessing step for the algorithm discussed in Chapter 6. Different approaches are considered, and an algorithm is presented for performing the decomposition quickly.

Chapter 2

Definitions and Review

2.1 Introduction

Because of the wealth of research on MATs and associated symmetry sets, as well as the related work on equidistantial point sets and Voronoi diagrams, it is difficult to give a complete review of all the literature. We will instead briefly identify some of the research which is most directly related to this thesis. The review is split into three parts: the first considers some of the algorithms which have been developed to determine MATs and associated symmetry sets; the second examines known topological and analytic properties of such sets; and the third part reviews solvers of simultaneous nonlinear polynomial equations. Although the third portion of the review does not concern the Medial Axis and its related sets directly, such solvers are needed to determine certain important points on the Medial Axis, as we shall see in chapter 4.

Before we begin the review, however, we present the notation which will be used throughout this thesis, as well as a few definitions of the Medial Axis and some related sets.

2.2 Notation

1. The lower-case letters i, j, k, l, m, and n denote nonnegative integers.

2. The other lower-case letters denote real-valued functions or real numbers.

3. Lower-case letters in boldface, such as $\mathbf{x}$ and $\mathbf{y}$, will denote vectors of real numbers, points, or vector-valued functions. It should be clear from the context what the

dimension of any given vector is. When we work with the components of the vector $\mathbf{x}$, we will assume that x_i is the ith component of $\mathbf{x}$.

4. Upper-case letters such as A and B denote sets or matrices. Also, the element in the ith row, jth column of a matrix A is denoted a_{ij}.

5. The *closed interval* $[a, b]$ denotes the set of all real x such that $a \leq x \leq b$. The *open interval* (a, b) denotes the set of all real x such that $a < x < b$.

6. $\Re$ denotes the real numbers; $\Re^n$ denotes the set of points in n-dimensional Euclidean space.

7. When pseudocode is given, the conventions used are those found in *Introduction to Algorithms* by Cormen, Leiserson, and Rivest [22]. Indentation is used to indicate blocks of code, rather than curly braces as in the C language, or the **begin-end** construct found in Pascal.

2.3 Definitions

This section is split up into two subsections. The first covers definitions from topology and analysis, and the second gives definitions of the Medial Axis and related sets.

2.3.1 General Definitions

We will use a number of concepts from topology, analysis, and differential geometry in this thesis, so we now briefly state some of the more important definitions. We refer the reader to Munkres [54] for a comprehensive introduction to topology, to Rudin [70] for an introduction to mathematical analysis, and to Warner [86] for an advanced treatment of differential geometry.

We begin with some important topological definitions from [54] and [70]:

Definition 2.1 Let $\mathbf{x} \in \Re^n$ and let $r > 0$. The *open ball* of radius r at $\mathbf{x}$, $B_r(\mathbf{x})$, is defined as the set $\{\mathbf{y} \in \Re^n : d(\mathbf{x}, \mathbf{y}) < r\}$, where $d(\mathbf{x}, \mathbf{y})$ represents the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$. The *closed ball* $\overline{B}_r(\mathbf{x})$ is the set $\{\mathbf{y} \in \Re^n : d(\mathbf{x}, \mathbf{y}) \leq r\}$.

Note that the distance function may be defined for sets as well as for points:

Definition 2.2 The *distance* between two sets A and B in $\Re^n$ is defined as

$$d(A,B) = \inf_{\mathbf{x}\in A,\mathbf{y}\in B}\{d(\mathbf{x},\mathbf{y})\}$$

where $d(\mathbf{x},\mathbf{y})$ is the Euclidean distance from $\mathbf{x}$ to $\mathbf{y}$ and inf represents the greatest lower bound.

Definition 2.3 Let $A \subset \Re^n$. Let $\mathbf{x} \in \Re^n$.

1. $\mathbf{x}$ is an *interior point* of A if there exists an $\epsilon > 0$ such that $B_\epsilon(\mathbf{x}) \subset A$. The set of all interior points of A is called the *interior* of A and is denoted A^0.

2. A is called an *open* set if every point in A is an interior point.

3. $\mathbf{x}$ is a *limit point* of A if for any $\epsilon > 0$, $B_\epsilon(\mathbf{x}) \cap A \neq \emptyset$. (Here $\emptyset$ denotes the empty set.) Note that $\mathbf{x}$ may or may not be an element of A.

4. The *closure* of A, denoted $\overline{A}$, is the union of A with its limit points. If $A = \overline{A}$, then we say that A is a *closed* set.

The definitions above are valid even for more general spaces than $\Re^n$, although we do not consider such spaces in this thesis. As long as open and closed balls are defined in a space, they all apply as written. Since the existence of open and closed balls depends only on the existence of a distance function, or *metric*, the definitions above are applicable in any *metric space.*

Definition 2.4 A *metric space* (X,d) (or sometimes simply X when d is understood) is a space together with a continuous function $d : X \times X \longrightarrow \Re$ satisfying the following properties:

1. $d(\mathbf{x},\mathbf{x}) = 0$ for all $\mathbf{x} \in X$.

2. $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$ for all $\mathbf{x},\mathbf{y} \in X$ (symmetry property).

3. $d(\mathbf{x},\mathbf{z}) \leq d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{z})$ for all $\mathbf{x},\mathbf{y},\mathbf{z} \in X$ (triangle inequality).

For simplicity, we shall concern ourselves only with $\Re^n$ in this thesis, and merely note that some of our results as well as our definitions are also applicable in more general spaces. For more information on metric spaces, see Munkres [54].

The *inward offset* and the *outward offset* may be defined in terms of the distance to sets. The following definition is adapted from Rossignac and Requicha [69]:

Definition 2.5 Let $A \subset \Re^n$ be a regular set (a set such that the closure of the interior of A equals A). Then the *outward offset* of A at a distance r is the set $\{\mathbf{x} : d(\mathbf{x}, A) \leq r\}$. The *inward offset* of A at a distance r is the set $\overline{\Re^n - \{\mathbf{x} : d(x, \Re^n - A) \leq r\}}$.

Definition 2.6 Let $X \subset \Re^k$ and $Y \subset \Re^n$. Let $f : X \longrightarrow Y$ be a mapping. We say that f is *continuous of class* C^r or simply C^r if all of the derivatives of f up to rth order are continuous at each $\mathbf{x} \in X$. If f is C^0, in particular, it means that f is continuous but that its derivatives need not be. We say that f is G^1 if f is continuous and the tangent space at every point $\mathbf{x}$ is also continuous. f is said to be a *homeomorphism* is both f and its inverse $f^{-1} : Y \longrightarrow X$ are defined and continuous. If a homeomorphism exists between X and Y, we say that X and Y are *homeomorphic.*

Note that G^1 continuity need not imply C^1 continuity; consider the function $\mathbf{f} : \Re \longrightarrow \Re^2$ defined by $\mathbf{f}(\mathbf{x}) = (\mathbf{x}, \mathbf{x})$ when $x < 0$ and $\mathbf{f}(\mathbf{x}) = (2\mathbf{x}, 2\mathbf{x})$ when $x \geq 0$. $\mathbf{f}$ is only C^0 at $x = 0$ but the tangent direction is continuous at $x = 0$, so $\mathbf{f}$ is G^1.

Definition 2.7 Let $A \subset \Re^n$.

1. A *separation* of A is a pair of disjoint, nonempty subsets of A, X and Y, such that $X \cup Y = A$, and $\overline{X} \cap Y = X \cap \overline{Y} = \emptyset$. (Here $\overline{X}$ denotes the closure of X.) We say that A is *connected* if no separation of A exists.

2. Let $\mathbf{x}$ and $\mathbf{y}$ be points in A. A *path* $f : [0, 1] \longrightarrow A$ between $\mathbf{x}$ and $\mathbf{y}$ is a continuous function such that $f(0) = \mathbf{x}$ and $f(1) = \mathbf{y}$. A is said to be *path connected* if any two points $\mathbf{x}, \mathbf{y} \in A$ can be joined by a path. Note that if A is path connected, it is automatically connected, although the converse is not necessarily true (see Munkres [54]).

3. The set A is said to be *bounded* if there exists a real number $M > 0$ such that for any points $\mathbf{x}$ and $\mathbf{y}$ in A, $d(\mathbf{x}, \mathbf{y}) < M$.

4. A is said to be *compact* if it is closed and bounded.

For more general definitions of compactness and connectedness for spaces other than $\Re^n$, see Munkres [54].

Occasionally we will talk about subspaces of $\Re^n$:

Definition 2.8 A *subspace* $S \subset \Re^n$ is a subset of $\Re^n$ with a topological structure induced by $\Re^n$. Specifically, if V is an open set of $\Re^n$, then $V \cap S$ is an open set of S. When we speak of a *neighborhood* U of $\mathbf{x} \in S$, we assume that U is an open set in S containing $\mathbf{x}$.

Finally, we will need the definition of a *manifold.* A manifold is essentially a generalization of curves and surfaces to $\Re^n$ and is characterized by the property that there is a local parameterization of the manifold around any point on the manifold. The following definition is adapted from Wolter [88]:

Definition 2.9 Let $M \subset \Re^n$. M is a *an unbordered k-dimensional submanifold of* $\Re^n$ if every point $\mathbf{x} \in M$ has a neighborhood in M which is homeomorphic to $\Re^k$. M is a *k-dimensional submanifold of $\Re^n$ with boundary* if every point $\mathbf{x} \in M$ has a neighborhood in M which is homeomorphic to either $\Re^k$ or the k-dimensional halfspace $H^k = [0, \infty) \times \Re^{k-1}$. If the neighborhood is homeomorphic to H^k, then $\mathbf{x}$ is said to be a *boundary point.* The set of boundary points make up the manifold's *boundary* ∂M.

2.3.2 Definitions of the Medial Axis and Related Sets

Now that we have some basic definitions from analysis and topology, we can define the Medial Axis and some of its related sets.

Definition 2.10 Let D be a subset of $\Re^n$. A closed ball (or disc in 2-D) is said to be *maximal* in D if it is contained in D but is not a proper subset of any other ball (disc) contained in D.

Definition 2.11 The *Medial Axis* (MA), or skeleton, of a subset D of $\Re^n$, denoted $M(D)$, is the locus of points which lie at the centers of all closed balls which are maximal in D, together with the limit points of this locus. The *radius function* of the Medial Axis of D is a continuous, real-valued function defined on $M(D)$ whose value at each point on the Medial

Axis is equal to the radius of the associated maximal ball. The *Medial Axis Transform* (MAT) of D is the Medial Axis together with its associated radius function.

When D is a n-dimensional submanifold of $\Re^n$ with boundary ∂D, the points of intersection between a maximal ball centered at $\mathbf{x} \in D$ and ∂D are called *footpoints* of $\mathbf{x}$. The set of footpoints of $\mathbf{x}$ is also sometimes called the *projection* of $\mathbf{x}$ on ∂D.

These two definitions are adapted from Blum [6]. As in Wolter's paper [88], definition 2.11 has been modified from Blum's original definition by the inclusion of limit points of the locus of centers of maximal balls. The inclusion of these limit points makes the Medial Axis closed and even compact if D is bounded. Notice also that although we have defined the MAT for $\Re^n$ only, it generalizes naturally to any metric space; all that is needed is the existence of balls in the space, which depends on the existence of a metric.

It is clear that the Medial Axis is closely associated with equidistantial point sets; since maximal balls usually touch the boundary of D in more than one point (although not always), a dense subset of points on the Medial Axis are *equidistant* from more than one point on the boundary. Thus there are relationships between the Medial Axis and equidistantial point sets which are further explored in other work (see sections 2.4 and 2.5). The most well known of these sets is the Voronoi Diagram (see Aurenhammer [2] or Fortune [30] for a review):

Definition 2.12 Consider a collection of sets $\{E_i\}$ in a metric space with a distance function d. The *Voronoi region* associated with E_i, denoted $\mathrm{Vor}(E_i)$ is the locus of points whose distance to a point of E_i is less than the distance to any point in any other set E_j where $j \neq i$. The *Voronoi Diagram* $\mathrm{VoD}(\{E_i\})$ of the collection is the locus of points in the metric space which do not lie in the Voronoi region of any set in the collection.

In the case of polygonal regions in 2-D and polyhedral regions in 3-D, the Voronoi Diagram is generally a superset of the Medial Axis. A well-known result from Wolter [88] states that the set of points equidistant from two or more closest points on the boundary is dense in the Medial Axis. Typically, for a polygonal shape, the boundary elements $\{E_i\}$ consist of all edges and all non-convex vertices, and for a polyhedral solid the boundary elements are all faces and non-convex edges and vertices. Since any maximal ball can only be tangent to a boundary element in one point for polygonal or polyhedral shapes (due

to the linearity of the boundary elements) and since points on the Voronoi Diagram are equidistant from two or more boundary elements, it is easy to see that the Voronoi Diagram is a superset of the Medial Axis for such shapes. The sets need not be equal; consider the behavior of both sets due to a non-convex vertex in a 2-D polygon. The Voronoi Diagram has two edges meeting at the vertex which are equidistantial from the vertex and each of its adjacent edges. However, there are no maximal balls in a neighborhood of the vertex, so the Medial Axis does not contain these edges.

For objects with non-linear boundary elements, the relationship is not always so obvious. Because curved boundary elements (a parabola, for example) may have maximal balls tangent in more than one point of the element, the Voronoi Diagram may not be a superset of the Medial Axis. The fundamental difference between the two sets is that the Medial Axis is an intrinsic representation of an object, whereas the Voronoi Diagram depends on a decomposition of the object.

Another important equidistantial point set is the *Cut Locus* examined by Wolter [87] [88]:

Definition 2.13 Let S be a subset of $\Re^n$. For any $\mathbf{x} \in \Re^n$, let $d(\mathbf{x}, S) = \min_{\mathbf{y} \in S}\{d(\mathbf{x}, \mathbf{y})\}$ be the minimum distance from $\mathbf{x}$ to the set S. A *minimal join* (or *shortest path*) from a point $\mathbf{x} \in \Re^n$ to S is a line segment joining $\mathbf{x}$ to some point $\mathbf{x}' \in S$ such that $d(\mathbf{x}, S) = d(\mathbf{x}, \mathbf{x}')$. The *Cut Locus* of S is the closure of the locus of points in $\Re^n$ which have more than one minimal join to S.

It is shown by Wolter [88] that when B is a closed n-dimensional manifold of $\Re^n$ with boundary ∂B, then the Medial Axis of B is equal to the intersection of the cut locus of ∂B with B.

The last and most general of the commonly encountered equidistantial sets is the *Symmetry Set*, see Scott *et al.* [71]. This set has a definition similar to that of the Cut Locus; however the restriction on minimal joins that they must be minimal over all of S is relaxed to the condition that they need be minimal only over some neighborhood of x':

Definition 2.14 Let S and d be the same as in definition 2.13. A *locally minimal join* from $\mathbf{x} \in \Re^n$ to S is a line segment joining $\mathbf{x}$ to some point $\mathbf{x}' \in S$ such that $d(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}, U)$ for some neighborhood $U \subset S$ of $\mathbf{x}'$. The *Symmetry Set* of S is the closure of the locus of

points in $\Re^n$ which have more than one locally minimal join to S of the same length.

The only other definition we have seen for the symmetry set is the set of centers of circles tangent to two or more points on a plane curve [71]. Definition 2.14 covers a more general class of boundaries and is more clearly related to the Cut Locus.

2.4 Algorithms for Determining the MAT or Related Sets

The MAT was introduced and explored by Blum [6] and further explored by Blum [5] and Blum and Nagel [7] to describe biological shape. Soon after it was introduced, continuous algorithms for computing the MAT were developed for special planar regions. Montanari [53] developed an algorithm to compute the MAT of a multiply-connected polygonal figure. His algorithm proceeds by identifying significant *branch points* and propagating the boundary contour inward, while connecting the branch points with appropriate linear or parabolic segments. A more efficient algorithm for computing the MAT of a convex polygonal figure in $\mathcal{O}(n \log n)$ was presented by Preparata [65] along with an $\mathcal{O}(n^2)$ algorithm for a non-convex polygon. Lee [48] developed an $\mathcal{O}(n \log n)$ algorithm for polygons with non-convex corners. Srinivasan and Nackman [79] presented an $\mathcal{O}(nh + n \log n)$ algorithm for multiply connected polygons with h holes. More recently, Gursoy [35] and Gursoy and Patrikalakis [61] [36] [37] [38] developed an algorithm to compute the MAT of a multiply connected planar region bounded by line segments and circular arcs, and used this algorithm to generate finite element meshes automatically and determine global shape characteristics. Chou [20] develops an algorithm to compute the Voronoi Diagram of a planar shape bounded by arbitrary closed curves. Guibas and Stolfi [34] investigate the relationship between the Voronoi Diagram and the Delaunay Triangulation, and develop the quad-edge data structure to represent them. Sugihara [82] investigates the use of Voronoi Diagrams to approximate various types of generalized Voronoi Diagrams. Rosenfeld [68] considers different representations of shapes based on an axis and a generation rule. Farouki and Johnstone [28] investigate the bisector between a point and a plane curve. Held's book [39] contains a comprehensive review of Voronoi Diagram algorithms, which he uses in the context of pocket machining. Another comprehensive review of the state of the art in Voronoi Diagram algorithms has been compiled by Aurenhammer [2].

Other work has concentrated on discrete and approximate approaches to determine the

MAT or its related sets. Nackman [55] proposes a 3-D algorithm to use a polyhedral approximation of a smooth boundary and produce a polyhedral approximation to the skeleton. The algorithm is an extension of Bookstein's line skeleton approach [9] to 3-D. It takes as input a polyhedral surface made up of convex polygons and generates a connected graph of convex polygons approximating the MA of the original object; since the input polyhedron is assumed to be an approximation to a smooth curved object, the output is not the skeleton of the polyhedron itself but rather a collection of polygons approximately tangent to the skeleton of the true object. Lavender *et al.* [47] use an octree-based approach to determine the Voronoi Diagram. Their algorithm works on set-theoretic solid models, composed of unions, intersections, and differences of primitive regions represented by a collection of polynomial inequalities, and produces an octree (or quadtree in two dimensions) which divides space into Voronoi regions at some specified resolution. Scott *et al.* [71] discuss a method for determining the Symmetric Axis which is based on a combined wave/diffusion process in the plane. Their algorithm proceeds by assigning each boundary pixel a unit displacement above the plane and every other pixel a zero displacement, and then numerically propagating a wave from the boundary. The wave is attenuated by a diffusion process to reduce numerical error, and local maxima in the wave are declared to lie on the Symmetric Axis. Although useful for binary images at low resolutions, the error may be large for higher resolutions. Memory and processing requirements for this method tend to be quite high as well.

Brandt [10] [13] [12] [11] finds a continuous approximation to the skeleton in both the planar and the 3-D case by first discretizing the boundary. He samples the boundary at a given sampling density, yielding a set of discrete points which form a pixelized or voxelized approximation to the boundary. The next step is to run an efficient discrete-point Voronoi diagram on the set of points. Finally, portions of the skeleton which result from the effects of quantization are pruned away. Since it is typically true that such spurious branches of the skeleton have a rapidly changing radius function, the ratio between the radius function at two adjacent vertices and the distance between the two vertices may be computed. This ratio is an approximation to the slope of the radius function; if it is close to 1, the vertex with the smaller radius can be discarded. Another strategy for pruning such spurious branches is also discussed in [11]. This approach attempts to classify each of the vertices in the interior Voronoi diagram according to how many footpoints the vertex has. The number of

footpoints is determined by taking the associated maximal sphere at the vertex, increasing the radius slightly, and intersecting the dilated sphere with the boundary. This intersection partitions the surface of the sphere into areas which lie either inside or outside the region. Each area lying outside the boundary is assumed to correspond to a footpoint; since the most commonly occurring type of skeleton point has two footpoints, only these points are kept, and the rest are pruned away.

Chiang [19] takes a planar region bounded by piecewise C^2 curves and performs a cellular decomposition of the plane in a neighborhood of the region. Each cell is assigned an approximate distance to the nearest point on the boundary of the region using an algorithm due to Danielsson [24] which computes the Euclidean distance transform. This information is later used to find a starting point for tracing axis branches in two dimensions and for recognizing when the tracing has passed the end of a branch. The tracing itself uses the distance information to determine on which boundary elements the footpoints of the current Medial Axis point lie. Once these elements are known, a set of simultaneous equations describing the local structure of the MA near the given point is formed. Using these equations to determine the tangent to the MA at the given point, a short distance along the tangent is traversed, the point is refined with Newton iteration, and another tracing step is taken. At each step, the distance information is used to determine whether or not the current branch has become inactive. If so, a branch point or an end point has been hit, and the tracing either proceeds along another branch or stops. Although the tracing is not extended to three dimensions, Chiang [19] notes that the same Euclidean distance transform in 3-D may be used to determine an approximation to the skeleton. One simple way of using the distance transform in this way is to identify those points which have locally maximal distance values after the distance transform is carried out. Such points are clearly close to centers of maximal disks, and so can be considered to provide an approximation to the skeleton.

Sudhalkar *et al.* [81] introduce a set called the *box skeleton* which they argue has the properties which make the MAT desirable as an alternate representation of shape. In particular, the box skeleton exhibits dimensional reduction, homotopic equivalence and invertibility. However, their skeleton is defined using the L_∞ norm (the box norm) instead of the Euclidean norm and thus may be quite different from the Medial Axis. Their algorithm for determining the box skeleton operates on discrete objects made of unit squares (or cubes,

in 3-D) and proceeds by *thinning* the object while maintaining homotopic equivalence to the original object. In order to perform the thinning, the object is transformed into a graph; in the planar case, the boundaries between adjacent pixels are considered to be edges of the graph, and the intersections of these edges are the vertices of the graph. The first thinning step proceeds by replacing the graph by that portion of the *dual* graph which is interior to the original (primal) graph. Since this procedure alone may result in a disconnected skeleton, the boundary "shrink wraps" around the skeleton as it thins. Procedures based on these concepts are developed for both 2-D and 3-D discrete objects.

Most of the 3-D algorithms in existence (such as the ones above) are fundamentally *discrete* algorithms. To our knowledge, few *continuous* approaches have been proposed, due largely to the computational complexity involved. One of the few such techniques is developed by Hoffmann [42], who proposed a method for assembling the skeleton of a CSG object. His method proceeds by determining points of closest approach between pairs of boundary elements and checking these points to make sure they are in fact on the Medial Axis. (Each of these points are on the MA if and only if the distance to the pair of elements is less than or equal to the distances to the other boundary elements.) The points are then sorted in order of increasing distance from the boundary, and then a local analysis around each point is performed, in an attempt to identify whether the point lies on a face, edge, or vertex of the Medial Axis. This determination is made by identifying all boundary elements which lie at the same minimum distance from the point and forming a set of simultaneous equations in n variables which describe the equidistantial set that the point belongs to. Based on the rank of the Jacobian of this set of equations, the point is predicted to lie on a face, edge, or vertex of the skeleton. Then neighboring faces and edges are traced out in order of increasing distance from the boundary. However, the method appears not to have been further developed and implemented. Furthermore, it is necessary in the method to intersect equidistantial sets with one another in order to trim away portions which do not belong to the Medial Axis, and consequently the algorithm carries additional computational cost. Related papers by Dutta and Hoffmann [26] [27] consider the exact representation of the bisectors which appear as skeleton branches in the skeleton of CSG objects bounded by planes, natural quadrics, and torii.

Reddy and Turkiyyah, in a recent manuscript [67] received during the final editing of this dissertation, propose an algorithm for determining the skeleton of a 3-D polyhedron

based on the generalization of the Voronoi Diagram defined above. They compute an abstract Delaunay Triangulation of the polyhedron and use the result to obtain the dual, the generalized Voronoi Diagram. The Delaunay triangulation computed is a generalization of the usual Delaunay triangulation, which connects isolated nodes together with line segments. In the generalization, the nodes represent parts of a polyhedron (specifically, either a face or a non-convex edge or a non-convex vertex of the polyhedron) and therefore the triangulation is an abstract graph. It still maintains duality with the generalized Voronoi Diagram we defined earlier in Definition 2.12, and is easier to compute since the Voronoi Diagram may contain trimmed quadric surfaces. The skeleton of the polyhedron is obtained by trimming away certain elements of the generalized Voronoi Diagram which are in the Voronoi Diagram but not the skeleton (for example, the equidistantial point set between a face and a non-convex vertex bounding it). The algorithm can explicitly determine certain critical points of the skeleton, but does not contain accurate representations of the curves and surfaces making up the skeleton. In addition, the stability and ability of the algorithm to handle degeneracy are difficult to evaluate and are not analyzed in [67].

Other work by Sheehy *et al.* [74] [75] investigates the use of a Domain Delaunay triangulation on a distribution of points on the boundary (in a manner similar to Brandt [10] [11]) to attempt to determine the topological features of the Medial Axis of a B-Rep solid. The steps of an algorithm to compute the Medial Axis from these features are outlined. This work was also received during the final editing of this dissertation, and appears to be still in progress.

In addition to the work described above which discusses the determination of the Medial Axis Transform, there has been additional work on the inverse problem of *reconstructing* the original solid from the Medial Axis and the radius function. Reconstruction of boundary surfaces from curves and surfaces of the Medial Axis in 3-D is discussed by Gelston and Dutta in [32]. Vermeer's thesis [85] considers the problem for 2-D and 3-D skeletons in great detail.

In settling on a method for computing the skeleton in 3-D, it is illustrative to compare the features of the above techniques. Many of the above discrete techniques have continuous aspects to them, and are therefore relevant to the development of a fully continuous approach. Although we feel that Brandt's approach [10] [11] is limited by quantization error, his work on the classification of skeleton points is relevant to a continuous approach.

Chiang's work [19] on the 3-D problem is preliminary, but we believe that his basic idea in 2-D of tracing branches using sets of simultaneous polynomial equations is extensible to 3-D, and his Euclidean distance transform could be a valuable tool in reducing execution time of a continuous algorithm. The papers by Reddy and Turkiyyah [67] and Sheehy *et al.* [74] [75] appear promising, but they are too recent to evaluate in greater technical depth.

2.5 Topological and Differential Properties

Wolter [88] provides a thorough analysis of topological properties of the MAT, and establishes the relationship between the MA and related symmetry sets such as the cut locus. Principal results of his paper include his proof of homotopic equivalence between an object with a C^2 boundary and its Medial Axis, the invertibility of the MAT, and the C^1 smoothness of the distance function on the complement of the cut locus. Wolter also analyzes the differentiability of the distance function in [87].

Chiang [19] and Brandt [10] [11] study many of the mathematical properties of the MAT, particularly for two-dimensional regions. Principal results of Chiang's thesis include the proof that for two-dimensional regions with piecewise C^2 boundaries, the MA is connected, the MAT is invertible, and the maximal disc of a given MA point usually divides the MA and the boundary into disjoint objects (which provides justification for divide-and-conquer approaches). Brandt computes first- and second-order differential properties of the planar skeleton, shows homotopic equivalence between the skeleton and its boundary under certain conditions, and explores the determination of the skeleton under different metrics (which may be useful for determining the skeleton of binary images). He also explores the notion of skeleton point classification [11], classifying skeleton points according to the number of footpoints.

The three-dimensional problem is studied in some detail by Nackman [55] and Nackman and Pizer [56], who also derive relationships between curvatures of the boundary, the skeleton, and the associated radius function. Curvature relationships in the planar case are considered by Blum [5].

2.6 Solvers of Nonlinear Polynomial Equation Systems

Although this portion of the review does not concern the Medial Axis and its related sets directly, solvers of nonlinear polynomial equation systems are needed to determine certain important points on the Medial Axis, as we shall see in chapter 4. Therefore a review of some recent methods is necessary.

In recent CAD-related research, three classes of methods for the computation of solutions of nonlinear polynomial systems have been favored: algebraic techniques, homotopy, and subdivision [77]. These methods may be classified as *global* because they are designed to compute all roots in some area of interest. There also exist a number of *local* numerical techniques which employ some variation of Newton-Raphson iteration or numerical optimization [23]. These methods are used in CAD applications requiring high accuracy because they are efficient (usually exhibiting quadratic convergence rates close to simple roots) and are straightforward to program. However, they typically require good initial approximations to roots; such approximations are usually obtained through some sort of global search like sampling, a process which cannot provide full assurance that all roots have been found. This lack of robustness makes the development of efficient and stable global techniques desirable. In what follows, we briefly review three classes of global methods.

Algebraic geometry techniques like elimination or Groebner basis methods [17] [18] belong to the first class of methods. These methods have many advantages; they are theoretically elegant, guaranteed to find all complex roots of a system irrespective of the dimensionality of the solution set, and well-suited for implementation in symbolic mathematical systems. However, many of them suffer from numerical instability, making implementation in floating point arithmetic difficult. Furthermore, they are inefficient in memory and processing time requirements and therefore unattractive for systems only moderate in degree or dimensionality. Recently, some work has been done in combining elimination with numerical methods. Manocha [52] presents an algorithm which uses resultants to eliminate variables from a system of polynomial equations. Since, as he demonstrates, the resultant of the system corresponds to the determinant of a matrix polynomial, the problem of finding the roots of the system corresponds to an eigenvalue problem.

The second category of methods is the class of *homotopy techniques* [31] [89]. These methods may be used to find all complex solutions of a nonlinear polynomial system if

the number of roots is finite. Unfortunately, investigation of such methods indicates that they also tend to be numerically ill-conditioned. If we try to get around this problem by implementing the algorithm in exact rational arithmetic, we end up with enormous memory requirements because we have to solve large systems of complex initial value problems. Furthermore, such techniques are excessive in many problems we encounter where we only need real roots within a bounded set.

The third class, the category of *subdivision-based techniques*, is of more interest to us. Although not as general as algebraic methods, since they are only capable of isolating zero-dimensional solutions, they are nevertheless fast and numerically stable. Most of them do have the additional drawback that after a fixed number of subdivision steps they are never absolutely certain that they have isolated each root when roots are closely spaced. Subdivision algorithms provide boxes which enclose *all* solutions, although some boxes may contain more than one solution. However, the chance that a subdivision algorithm will fail in this way drops drastically as the resolution tolerance is lowered. Our root-finding scheme is conceptually based on the results of many different researchers, and so we will review most of their work here. Lane and Riesenfeld [46] investigated the application of binary subdivision and the variation diminishing property of polynomials in the Bernstein basis to eliciting the real roots and extrema of a polynomial. Boehm [8] and Cohen *et al.* [21] extended this idea to general non-uniform subdivision of B-splines. Formulation of a class of geometric problems in more than one dimension, requiring the solution of nonlinear piecewise polynomial systems expressed in terms of B-splines, was made by Dokken[25], who suggested a solution approach based on [21]. Subdivision techniques have also been used in a wide variety of intersection problems for geometric modeling. Sederberg [72] developed an adaptive subdivision algorithm to intersect planar algebraic curves expressed in the barycentric Bernstein basis within triangles. Patrikalakis, Prakash, and Kriezis [64] [45] investigated the use of subdivision of algebraic curves in intersecting an implicit algebraic surface with a rational polynomial surface. Their method relies on the computation of real characteristic points of an algebraic curve represented in the tensor-product Bernstein basis within a rectangle, which typically involves intersecting two or three algebraic curves by repeated adaptive subdivision and minimization. Minimization is used to increase the precision of the root quadratically. Nishita et al. [58] developed an adaptive subdivision technique to intersect rays with trimmed rational polynomial surface patches, also recasting the problem as the

intersection of two algebraic curves expressed in the Bernstein basis. Sederberg and Nishita [73] extended this method to intersect parametric curves with parametric surfaces. Vafiadou and Patrikalakis [84] used a similar algorithm coupled with minimization to ray-trace offset surfaces.

Based on the work above, we have devised a general root-finding approach which can be used to locate all isolated roots of a system of simultaneous nonlinear polynomial equations. In Sherbrooke and Patrikalakis [77] and Sherbrooke [76], we devise two algorithms, the linearly convergent Projected-Polyhedron technique, and the quadratically convergent Linear Programming method, prove their convergence properties and analyze their computational complexity. The methods (as well as a hybrid technique which uses both methods) are used to solve distance computation problems and compute singular points of vector fields in Patrikalakis *et al.* [63] and Zhou *et al.* [90].

Since the subdivision methods above belong to the general class of interval-based methods, some review of other interval-based techniques is in order. These techniques are primarily interval Newton methods combined with bisection to ensure convergence, as in Kearfott [44] and Neumaier [57]. Bliek [4] studies such methods in the context of design automation and also reviews the state of the art of interval methods in detail. Most recently, Maekawa and Patrikalakis [51] [50] extend the Projected-Polyhedron method to operate in rounded interval arithmetic. Their examples, which use the method to decompose a parametric polynomial surface into regions with specific ranges of curvature [51], and compute singularities of normal offsets and intersections between two normal offsets [50], indicate that the extension is highly successful. Although the extension is slower, it is slower by only a constant factor, which can be expected to drop drastically on a machine with an efficient low-level interval arithmetic library or a machine capable of performing interval operations in hardware. Hu [43] extends the method further to handle unbalanced systems of equations (more equations than unknowns, and vice versa) and develops low-level operations to speed up the handling of interval arithmetic during the algorithm. An overview of the current state of the method may be found in Patrikalakis *et al.* [62] [60].

Chapter 3

Mathematical Properties of the MAT

3.1 Introduction and Definitions

In this chapter, we investigate the mathematics underlying the Medial Axis and its related sets. It turns out that many important results may be derived from well-understood principles of differential geometry and analysis. In particular, we will begin by concentrating on the effect of curvature on the shape of the Medial Axis. Then later on in the chapter we will address a deeper mathematical concern: the relationship between an object and its Medial Axis. We will show that for a broad class of objects, the Medial Axis is a *strong deformation retract* of its associated object and therefore, among other properties, the Medial Axis is path connected if the object is. The connectivity of the Medial Axis is extremely important since the algorithms we will introduce in later chapters crucially depend on it.

For the purposes of our discussion, it is useful to redefine the Medial Axis Transform more precisely:

Definition 3.1 Let D be a subset of $\Re^n$. Then the *Medial Axis Transform* of D, $MAT(D)$, is a subset of $\Re^{n+1}$ consisting of the closure of all points $(\mathbf{x}, r)$ satisfying the following properties:

1. The closed ball of radius r around $\mathbf{x}$, $\overline{B}_r(\mathbf{x})$, is entirely contained within D.

2. If $\overline{B}_r(\mathbf{x}) \subset \overline{B}_{r'}(\mathbf{y})$ for some $\mathbf{y} \in D$ and $r' > r$, then $\overline{B}_{r'}(\mathbf{y}) \not\subset D$.

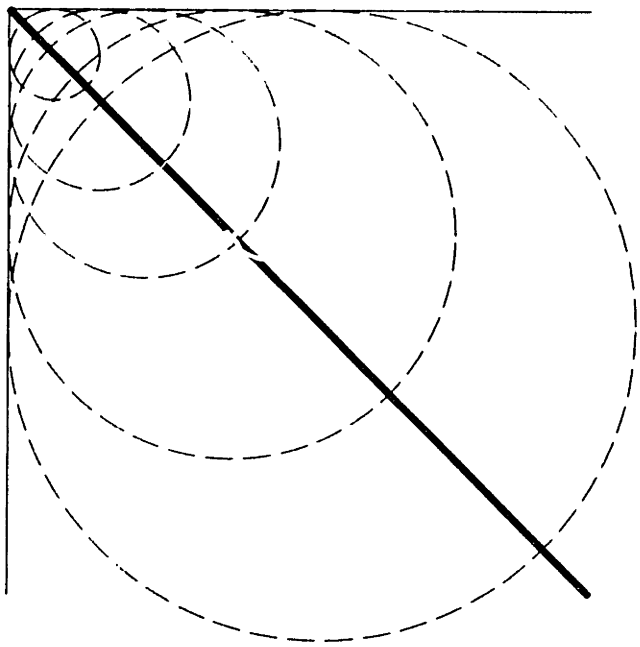

Figure 3-1: Radii of maximal spheres converge to 0 near the corner of a 2-D rectangle. Therefore the corner of the rectangle is a limit point of maximal disc centers but not itself a center of a maximal disc.

When the two conditions above hold, then $\overline{B}_r(\mathbf{x})$ is *maximal* with respect to D. The *Medial Axis* of D, denoted $M(D)$, is the projection of $MAT(D)$ to $\Re^n$ obtained by discarding the last coordinate r of each point in $MAT(D)$.

In this thesis, we will be concerned mainly with compact three-dimensional regions bounded by piecewise C^2 surfaces (such as may arise in a geometric modeling system which uses a boundary representation (B-Rep) based on rational B-spline curves and surface patches). Notice that we wish the Medial Axis to be a closed set, which is why we take the closure of all centers of maximal spheres. As shown in Wolter [88], it is possible, even when the boundary is Lipschitz continuous up to the first derivative, that the limit of a sequence of centers of maximal discs is not itself a center of a maximal disc. A more obvious example occurs at the corner of a rectangle. In this case, the corner is the limit of a sequence of centers of maximal discs (see figure 3-1) whose radii are decreasing to 0. We choose to include such points since closing the Medial Axis adds the valuable property of compactness to the set.

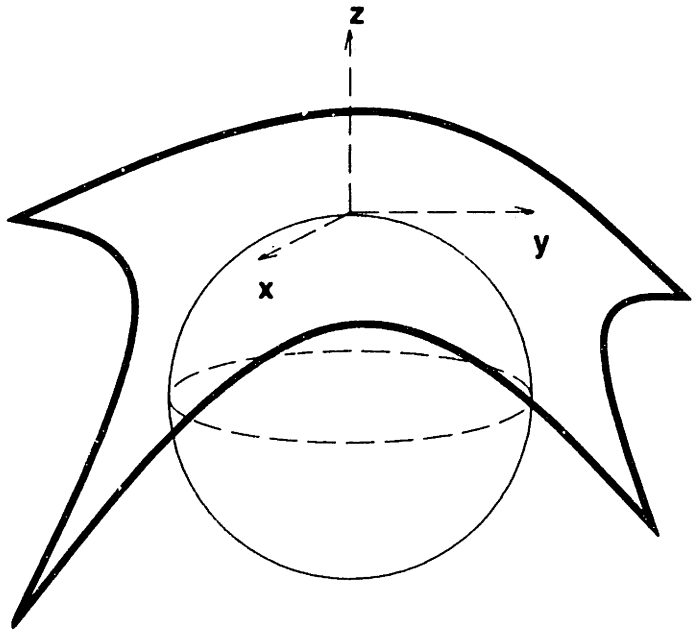

Figure 3-2: A C^2 surface neighborhood of a point is translated to a neighborhood of the origin and rotated so that the outward-directed normal to the surface points directly upward. The maximal sphere tangent to the surface at this point is also shown.

3.2 Curvature Theorems

We begin our mathematical investigation by proving some important relations between the maximum principal curvature of a surface in 3-D and its Medial Axis. In his thesis, Wolter [87] considers generalizations of many of the results in this section, in the context of the Cut Locus. By limiting ourselves to 3-D, we can use simple geometric arguments to obtain these results, which will be used in Chapter 7, when we design an algorithm for curved objects in $\Re^3$. In order to simplify our discussions, we make the following assumptions, without loss of generality (see figure 3-2):

1. We are looking at a small area on the boundary of our object which is C^2.
2. The area of the surface we are interested in has been translated to a neighborhood of the origin.
3. The object has been rotated so that the normal of the boundary surface at the origin is pointing directly upward, in the direction of increasing z.
4. The normal is outward-directed. Thus, locally, the outside of the object is "up" while the inside is "down."

We also need a few definitions to clarify the discussion.

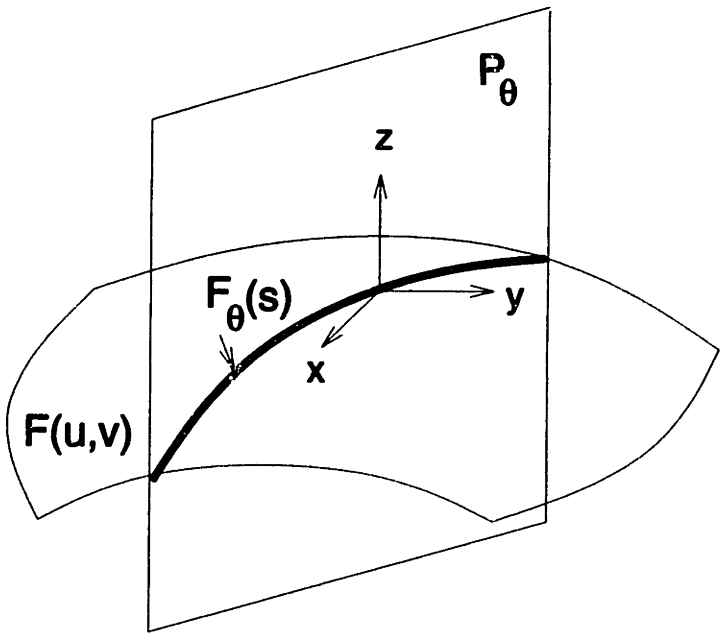

Figure 3-3: A normal section curve

Definition 3.2 Assume that the surface F is oriented according to the conditions enumerated above. Let P_θ denote the plane spanned by the z-axis and a vector in the (x, y)-plane at angle θ with respect to the x-axis. A *normal section curve of angle θ on F at the origin* $\mathbf{0}$, which we denote by F_θ, is the curve obtained by intersecting P_θ with F. We assume that F_θ is arc-length parameterized and that $F_\theta(0) = \mathbf{0}$.

Figure 3-3 illustrates the preceding definition. The purpose of introducing the normal section curve above is to enable us to work with the curvatures of the surface at the origin:

Definition 3.3 The *normal section curvature function* $\kappa_\theta(t)$ of the normal section curve F_θ is the real-valued function which represents the curvature of the curve at $F_\theta(t)$. Note that $\max_\theta\{\kappa_\theta(0)\}$ is the *maximum principal curvature* of F at $\mathbf{0}$ and that $\min_\theta\{\kappa_\theta(0)\}$ is the *minimum principal curvature*. For other θ, $\kappa_\theta(0)$ is equal to the normal curvature of F at $\mathbf{0}$ in the direction of θ.

We require that κ_θ be positive if F_θ is convex near the origin of P_θ (i.e. curving downward) and negative if it is concave (i.e. curving upward). The maximal sphere touching at the origin has the same curvature convention; since spheres are everywhere convex, the curvature will be positive in any normal section. In particular, the curvature of a sphere is equal to the reciprocal of the radius of the sphere and is independent of the normal section.

With these definitions in mind, we proceed to the first of our curvature theorems. We begin by stating and proving an important lemma which relates the curvature of F at a footpoint to the curvature of the maximal sphere at that footpoint.

Theorem 3.4 Let F be a C^2 surface making up the boundary ∂B of a submanifold B of $\Re^3$, and let $\mathbf{x}$ be a point in the interior of F (so that there is a C^2 neighborhood of $\mathbf{x}$ on F). Then any maximal ball S containing $\mathbf{x}$ must be tangent at $\mathbf{x}$; furthermore, its curvature (the reciprocal of the ball's radius) must be greater than or equal to the maximum principal curvature of the boundary surface at $\mathbf{x}$.

Proof: Establish coordinates on F so that it is situated as required in the conditions above. Thus now $\mathbf{x} = \mathbf{0}$, and the normal at $\mathbf{0}$ is directed upwards. First, we must demonstrate that S is tangent to F at $\mathbf{0}$. Note first that $\mathbf{0}$ must lie on the boundary of S; for if it did not, then $\mathbf{0}$ would lie in the interior of S and therefore in the interior of B since $S \subset B$. Now, the tangent plane to F at $\mathbf{0}$ is horizontal, since the normal to F is vertical there. Assume that the tangent plane to S at $\mathbf{0}$ is not horizontal. Then the two planes intersect transversely, and therefore in any neighborhood of $\mathbf{0}$ there are points on the tangent plane of the ball which lie above $\mathbf{0}$. But since the tangent planes represent first-order approximations to F and S respectively, for arbitrarily small neighborhoods of $\mathbf{0}$, there are points on S which have larger z-coordinates than points of F and therefore lie on the outside of F. Thus such a ball cannot be maximal, and we conclude that the tangent plane of S at $\mathbf{0}$ must coincide with the tangent plane of F at $\mathbf{0}$.

Now, to demonstrate the curvature relationship, consider an arbitrary *normal section* F_θ at $\mathbf{0}$ and a corresponding normal section of the sphere ∂S (denoted S_θ) at $\mathbf{0}$. Take a Taylor expansion of the z-coordinate of these curves around $\mathbf{0}$ up to second order (this is valid under the assumption that F is C^2). We obtain

$$z_{F_\theta}(t) = 0 + tz'_{F_\theta}(0) + \frac{t^2}{2} z''_{F_\theta}(\delta) \tag{3.1}$$

$$z_{S_\theta}(t) = 0 + tz'_{S_\theta}(0) + \frac{t^2}{2} z''_{S_\theta}(\epsilon) \tag{3.2}$$

where $|\delta| \leq |t|$ and $|\epsilon| \leq |t|$. Assume now that both normal sections are arc-length parameterized. Then the Frenet-Serret relations tell us that $z'_{F_\theta}(0)$ is equal to the z-component of the tangent vector to $\mathbf{0}$, which is simply 0 since the tangent plane is horizontal. Since

the tangent plane to S coincides with the tangent plane to F, $z'_{S_\theta}(0) = 0$ also. Another Frenet-Serret relation tells us that $\frac{dT}{ds} = -\kappa N$, where T is the unit tangent vector (equal to the first derivative) and N is the unit normal. (Here we have a negative sign in front of κ since our convention is that convex portions of a surface have positive curvature.)

Now, z_{F_θ} must be greater than or equal to z_{S_θ} in a neighborhood of $\mathbf{0}$, since we require that the ball be locally on the inside of F. Substituting in for the second derivatives in (3.1) and (3.2) we have

$$-\frac{t^2}{2}\kappa_\theta(\delta)N_F(\delta) \geq -\frac{t^2}{2}\kappa_S N_S(\epsilon) \tag{3.3}$$

where N_F is the z-component of the unit normal to F_θ and N_S is the z-component of the unit normal to S_θ. Notice that κ_S is independent of θ and t, since the curvature of a sphere is everywhere constant. Simplifying (3.3), we have

$$\kappa_S \geq \frac{N_F(\delta)}{N_S(\epsilon)}\kappa_\theta(\delta). \tag{3.4}$$

This relationship must hold for arbitrarily small neighborhoods of $\mathbf{0}$; in particular, it must hold in the limit as $t \to 0$. In this limit, δ and ϵ both approach 0 since they are both bounded by t, and thus

$$\kappa_S \geq \kappa_\theta(0) \tag{3.5}$$

since both N_F and N_S approach 1. Now this relationship must hold for all θ; in particular, it must hold for the maximum principal curvature at $\mathbf{x}$. Thus we conclude that the maximum principal curvature at $\mathbf{x}$ is always less than or equal to the curvature of the associated maximal ball.□

This theorem, although fairly simple conceptually, has far-reaching implications. In particular, it establishes an important link between maximum principal curvature of a surface and the radius function of the Medial Axis Transform. Notice that in the statement of the theorem we say there is a "less than or equal" relationship between the two curvatures, and most of the time we would expect the maximum principal curvature of the surface to be strictly less than the curvature of the maximal sphere. But what happens when they are equal? Intuitively, we might expect points with such a property to be points on the edges of the Medial Axis; points which, locally at least, have traveled as close as they possibly can get to the boundary. In fact, this generally turns out to be the case (see figure 3-4).

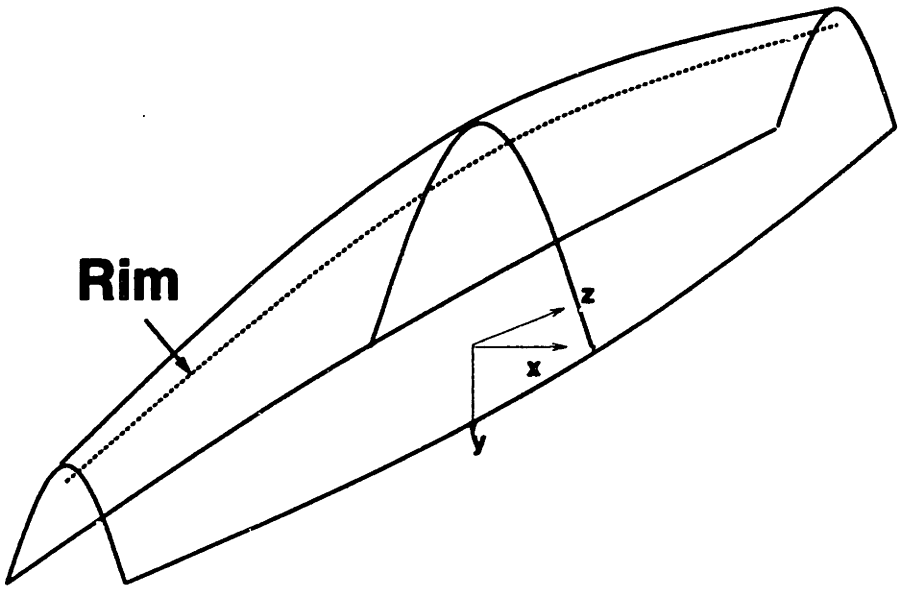

Figure 3-4: This figure shows a surface formed by taking a parabola and varying its maximum curvature as it is swept along the z-axis according to the rule $y = (z^2+1)x^2 - \frac{1}{z^2+1}, y \leq 0$; this surface is assumed to form part of the boundary of some 3-D object. The *rim* curve shown is an edge of the Medial Axis. It passes through the foci of the parabolas, where the radius of curvature of the parabola is equal to the distance from the focus to the parabola.

However, it is not necessarily obvious when such an equality would occur. For clarification, we turn to the following definition:

Definition 3.5 Let $\mathbf{x}$ be a point in a C^2 neighborhood on the surface F. Let $\kappa_{\max}$ be the maximum principal curvature of F at $\mathbf{x}$. Suppose there exists an $\varepsilon > 0$ such that for all θ, $\kappa_\theta(t) \leq \kappa_{\max}$ when $|t| < \varepsilon$. Then $\mathbf{x}$ is said to be a *local maximum of sectional curvature.*

It turns out that local maxima of sectional curvature give rise to a number of points on the edge of the Medial Axis, as the following theorem shows:

Theorem 3.6 Let F be as before, and let B be a submanifold of $\Re^3$ with $F \subset \partial B$ as part of its boundary. Let $\mathbf{x}$ be a local maximum of sectional curvature on F with maximum principal curvature strictly greater than zero, let S be the ball tangent to $\mathbf{x}$ with radius $r = \frac{1}{\kappa_{\max}(\mathbf{x})}$, and let $\mathbf{y}$ be its center. Then S is locally maximal in B near $\mathbf{x}$. If in addition $S \subset B$, then $\mathbf{y}$ is on the Medial Axis of B.

Proof: Again, establish coordinates on F so that $\mathbf{x} = \mathbf{0}$ and the normal at $\mathbf{x}$ points directly upward. To show that S is locally maximal, we need to show that it cannot lie in

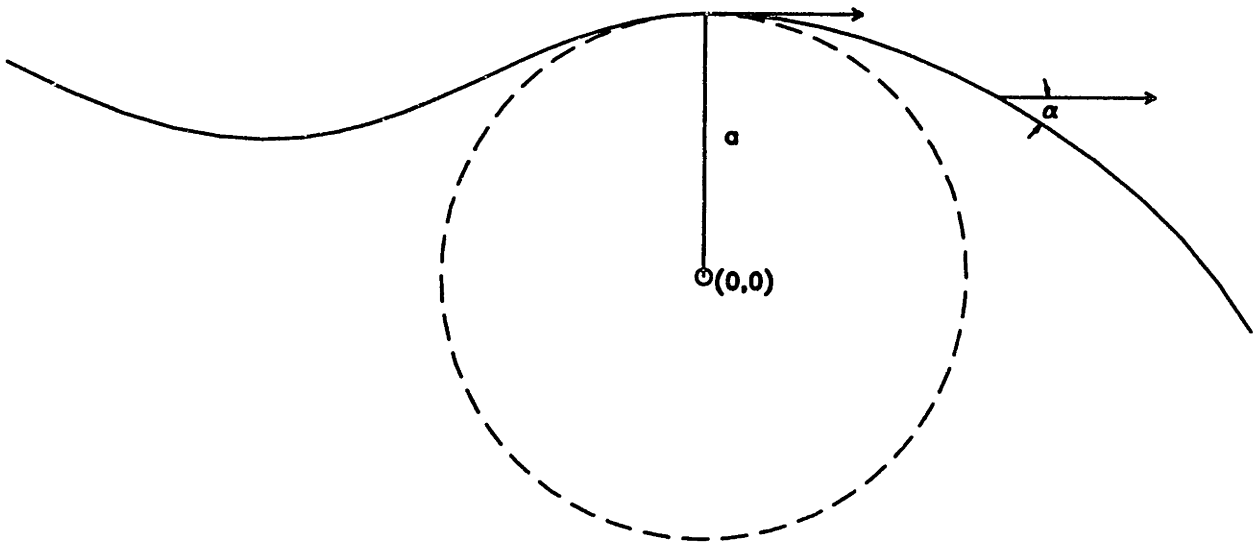

Figure 3-5: When the curvature of $\mathbf{r}$ is less than or equal to $\frac{1}{a}$, $\mathbf{r}$ stays outside the circle of radius a.

any larger ball in B but that it is contained in B in a neighborhood of $\mathbf{x}$.

Note that S is uniquely determined since we have specified its radius and center (the center $\mathbf{y}$ may be found by moving downward from $\mathbf{0}$ a distance equal to the specified radius). We claim that this ball is locally maximal in B. Note that since S contains $\mathbf{x}$, any larger ball must also contain $\mathbf{x}$. Therefore $\mathbf{x}$ will also be a footpoint on a larger ball. But by theorem 3.4 no maximal ball can have a larger radius than the reciprocal of the maximum curvature at a footpoint. Therefore, S cannot be contained in any larger ball which is itself contained in B.

It remains to show that S lies (locally) inside B. To show this, we use the following lemma:

Lemma 3.7 Consider a circle of radius a centered at the origin in the plane. Let $\mathbf{r}$ be an arc-length parameterized curve tangent to the point $(0, a)$ such that for $|t| < \varepsilon$, the curvature $\kappa(t)$ at $\mathbf{r}(t)$ is less than or equal to $\frac{1}{a}$. Then every point on $\mathbf{r}$ is at least a distance a from the origin.

Proof of Lemma:

This result is also considered in Wolter [88]. Here we provide a full proof. Assume without loss of generality that the tangent to $\mathbf{r}$ at $(0, a)$ is the vector $(1, 0)$, as shown in figure 3-5. (If it were pointing in the opposite direction, we could reparameterize the curve so that it was equal to $(1, 0)$.) We consider only the portion of the curve to the

right of $(0, a)$ since the proof for the other half of the curve is analogous. We use the well-known relationship between the curvature of a planar curve and the angle its tangent passes through:

$$\alpha(s_1) - \alpha(s_0) = \int_{s_0}^{s_1} \kappa(s)ds \tag{3.6}$$

where $\alpha(s)$ is the angle of the tangent with respect to the horizontal and s_0 and s_1 are parameter values on an arc-length parameterized curve. $\alpha(0) = 0$ since the tangent is horizontal to the right at $t = 0$. Therefore, $\alpha(t) = \int_0^t \kappa(s)ds$ is the angle of the unit normal at parameter t. Because of our curvature convention, positive curvature causes the curve to bend downward and therefore the tangent $\mathbf{r}'(t)$ is given by

$$\mathbf{r}'(t) = (\cos \int_0^t \kappa(s)ds, -\sin \int_0^t \kappa(s)ds). \tag{3.7}$$

And since $\mathbf{r}(t) = \mathbf{r}(0) + \int_0^t \mathbf{r}'(t)dt$,

$$\begin{aligned} x_r(t) &= \int_0^t \cos \int_0^t \kappa(s)dsdt \\ y_r(t) &= a - \int_0^t \sin \int_0^t \kappa(s)dsdt \end{aligned} \tag{3.8}$$

where x_r and y_r are the components of $\mathbf{r}$. But since $\int_0^t \kappa(s)ds \leq \frac{t}{a}$, and since when $p \leq q$ but both p and q are small, $\cos p \geq \cos q$ and $\sin p \leq \sin q$,

$$\begin{aligned} x_r(t) &\geq \int_0^t \cos \frac{t}{a}dt = a \sin \frac{t}{a} \\ y_r(t) &\geq a - \int_0^t \sin \frac{t}{a}dt = a \cos \frac{t}{a} \end{aligned} \tag{3.9}$$

And therefore the distance from $\mathbf{r}$ to the origin is $\sqrt{x_r^2 + y_r^2} \geq a$. □.

The use of this lemma is immediately apparent; if for $|t| \leq \varepsilon$ we know that $\kappa_\theta(t) \leq \kappa_{\max}$ for all θ, then each normal section of the ball S of radius $\frac{1}{\kappa_{\max}}$ lies within the corresponding normal section of F. Thus within a neighborhood of the origin, S is contained within B and hence is locally maximal within B. If in addition $S \subset B$, then $\mathbf{y} \in M(B)$ since we have already shown that S may not be contained in any larger ball in B.□

Let us pause for a moment and examine the meaning of this theorem. It states that if a certain curvature condition is satisfied at one footpoint, then the maximal sphere associated with that footpoint is uniquely determined as long as it is contained in B. Thus, frequently

a local maxima of sectional curvature will be the sole footpoint of its associated Medial Axis Point. Of course, this need not always be the case; although a ball of radius $\frac{1}{\kappa_{\max}}$ is guaranteed to be *locally* contained within B, it need not be completely contained. Therefore, this theorem becomes more powerful when we prove the following theorem, which shows that *every* point in $M(B)$ with only one footpoint in its projection has a radius function equal to the reciprocal of maximum curvature at the footpoint, if the footpoint lies in a C^2 neighborhood:

Theorem 3.8 Suppose $\mathbf{y}$ is a point on the Medial Axis of B, and let $\mathbf{x}$ on $F \subset \partial B$ be its sole footpoint. Let S be the maximal ball associated with $\mathbf{y}$, and suppose F is C^2 in a neighborhood of $\mathbf{x}$. Then the maximum principal curvature at $\mathbf{x}$ must be equal to the reciprocal of the radius of S.

Proof: A similar result is obtained in Wolter [88] for a completely C^2 boundary. Here we provide a different proof based on local arguments. Establish our usual coordinate system so that $\mathbf{x}$ is at $\mathbf{0}$ and $\mathbf{y}$ lies directly beneath $\mathbf{x}$ at some distance r. Now, it is clear by theorem 3.4 that r is less than or equal to the reciprocal of the maximum principal curvature of F at $\mathbf{0}$; or, equivalently, r is less than or equal to the minimum radius of curvature at $\mathbf{0}$. By continuity, for any $\varepsilon > 0$ there exists a $\delta > 0$ such that for $|t| < \delta$, $\kappa_\theta(t) < \kappa_{\max} + \varepsilon$ for all θ. Let $U \subset F$ be the neighborhood of $\mathbf{0}$ such that if $\mathbf{z} = F_\theta(t)$ for some θ and some t such that $|t| < \delta$, then $\mathbf{z} \in U$ (i.e., $U = \{F_\theta(t) : |t| < \delta, 0 \leq \theta < 2\pi\}$). Then $\partial B - U$ is a closed set and therefore compact. Thus the distance function from $\mathbf{y}$ to $\partial B - U$ has a minimum value r' which is strictly greater than r since there is no footpoint of $\mathbf{y}$ in $\partial B - U$. Thus if $\mathbf{y}'$ is a point such that $d(\mathbf{y}, \mathbf{y}') < \frac{r'-r}{2}$, then $d(\mathbf{y}', \partial B - U) > \frac{r'+r}{2}$ by the triangle inequality. In addition, by Lemma 3.7, as long as a ball tangent to F at $\mathbf{0}$ has a radius smaller than $\frac{1}{\kappa_{\max}+\varepsilon}$, it will be locally contained within B.

Now assume that the radius r of S is not equal to $\frac{1}{\kappa_{\max}}$. Then if we pick ε sufficiently small, there will be some real number a_1 such that $r < a_1 < \frac{1}{\kappa_{\max}+\varepsilon}$. There will also be a real number a_2 such that $r < a_2 < \frac{r'+r}{2}$. Then the ball of radius $a = \min\{a_1, a_2\}$ tangent to F at $\mathbf{0}$ is not only locally contained in B because $a_1 < \frac{1}{\kappa_{\max}+\varepsilon}$, but is also globally contained within B since its center $\mathbf{y}'$ is farther than $\frac{r'+r}{2}$ from $\partial B - U$. Furthermore, the ball is a proper superset of $B_r(\mathbf{y})$, which violates the assumption that $\mathbf{y}$ is on the Medial Axis. Therefore we have achieved a contradiction and thus $r = \frac{1}{\kappa_{\max}}$.□

3.3 The Connectivity of the Medial Axis

In this section we prove that if a compact manifold with a piecewise C^2, completely G^1 boundary is connected, then so is its Medial Axis. In fact, the theorem we will show is more general; we will prove that a *strong deformation retraction* exists between such a manifold and its Medial Axis and therefore the Medial Axis has a fundamental group isomorphic to the fundamental group of the manifold. A similar theorem was proven by Wolter [88]; it applies to submanifolds of $\Re^2$ with piecewise C^2 boundaries and submanifolds of $\Re^n$ with completely C^2 boundaries when $n > 2$. Thus our theorem, which relaxes the order of boundary continuity, is more general for $n > 2$. The theory of fundamental groups of spaces, which is a part of homotopy theory, is beyond the scope of this thesis, and we refer the reader to Munkres [54] for an introduction to the subject. In this thesis, we shall be mainly concerned with the fact that the strong deformation retraction preserves such properties as *path connectivity* and *simple connectivity.*

This general theorem implies that the Medial Axis of a polyhedral solid is connected. Although a polyhedral solid is not G^1 itself, it is piecewise C^2, and nearly G^1. Using a small offset to smooth most of the boundary of the solid, we will show that any two points on the Medial Axis of the solid may be joined by a path. The connectivity of the Medial Axis helps to justify the approach of our algorithm, which relies on connectivity to trace out the pieces of the Medial Axis. In Chapter 6, when we discuss the completeness of the algorithm for general polyhedra, we will use this result.

The notions of a *minimal join* and a *strong deformation retract* are of central importance to our proof. We have already defined minimal joins in definition 2.13, but we need to define strong deformation retracts:

Definition 3.9 Let I be the closed interval $[0, 1]$. Let A be a subspace of a topological space X. Then A is called a *strong deformation retract* of X if there exists a continuous map $H : X \times I \longrightarrow X$ such that

1. $H(x, 0) = x$ for $x \in X$,
2. $H(x, 1) \in A$ for $x \in X$,
3. $H(a, t) = a$ for $a \in A$ and $t \in I$.

H is called a *strong deformation retraction.*

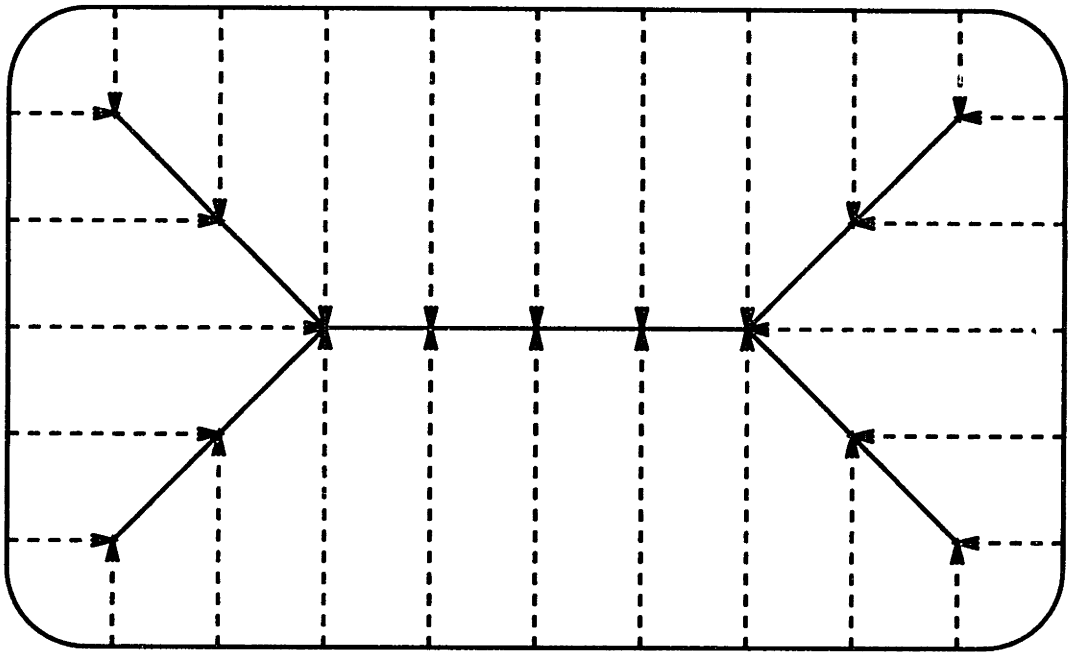

Figure 3-6: A strong deformation retraction of an area with a piecewise C^2, completely G^1 boundary. Points move along the arrows to the Medial Axis of the area.

A strong deformation retraction of a region in $\Re^2$ with a G^1 boundary is shown in figure 3-6. Definition 3.9 is taken from Munkres [54] and, loosely speaking, means that the space X may be shrunk into a subspace A in a continuous fashion. For example, the unit circle S^1 of radius 1 centered at the origin is a strong deformation retract of the punctured plane $\Re^2 - \mathbf{0}$ but not of $\Re^2$. We can see why this is so if we express every point in the punctured plane in terms of its polar coordinates r and θ; we move each point (r, θ) to the point $(1, \theta)$ in a continuous fashion. However, in $\Re^2$ this mapping does not work; any θ is a valid coordinate for the origin, so it is not clear where on S^1 to move the origin. If we arbitrarily pick a θ, a discontinuity will be introduced in the retract map at the origin.

Now we prove an important lemma, which establishes a map from an open subset of the boundary of a compact n-dimensional submanifold of $\Re^n$ to its Medial Axis and proves the continuity of the map. This subset is assumed to be G^1 and piecewise C^2.

Lemma 3.10 Let B be an n-dimensional compact submanifold of $\Re^n$ and let $M(B)$ be its Medial Axis. Let A be an open subset of ∂B which is G^1 and piecewise C^2 (recall that G^1 means that the tangent plane to A is continuous and piecewise C^2 means that A is an open subset of the union of a finite number of compact C^2 $n-1$-dimensional manifolds joined together in a G^1 manner). Then for every point $\mathbf{x} \in A$ there is one and only one maximal ball which touches $\mathbf{x}$. Furthermore, the function $\mathbf{s} : A \longrightarrow M(B)$, which maps each point

$\mathbf{x} \in A$ to the center of its maximal ball, is continuous.

Proof: This proof needs to be split into several steps.

Step 1: In this first step, we prove that every point $\mathbf{x} \in A$ has one and only one maximal ball touching it.

It is proven in Wolter [88] that the Medial Axis Transform is invertible; given the Medial Axis and the radius function, it is possible to reconstruct the original object as the union of all the balls centered on the Medial Axis with radii equal to the value of the radius function at the center. Therefore any point $\mathbf{x} \in \partial B$ must lie within some maximal ball. In particular, $\mathbf{x}$ may not lie in the interior of any maximal ball or else $\mathbf{x}$ would lie in the interior of B and not on the boundary. Therefore $\mathbf{x}$ must lie on the boundary of some maximal ball.

Now, we claim that if A is G^1, then any maximal ball touching $\mathbf{x}$ must have the same tangent space as ∂B at $\mathbf{x}$. For suppose some maximal ball touches $\mathbf{x}$ but does not have the same tangent space as A at $\mathbf{x}$. Then the tangent space of the sphere at $\mathbf{x}$ intersects the tangent space of A at $\mathbf{x}$ transversally. But since tangent spaces represent first-order approximations to functions, a transverse intersection of tangent spaces signifies that the ball crosses the boundary, by an application of Taylor's theorem. Since the boundary partitions space into two disjoint regions, one inside the manifold and one outside (see [88] for a proof), this crossing of the boundary violates the requirement that the ball lie entirely within B, and we therefore conclude that any maximal ball touching $\mathbf{x}$ must have the same tangent space as A at $\mathbf{x}$. (Note that this argument is a generalization of theorem 3.4 to n-dimensions.)

Suppose there are two maximal balls which have the same tangent space as A at $\mathbf{x}$. Both balls must lie on the same side of the tangent space, since they must be wholly inside of B. Elementary geometry tells us that a ball is determined uniquely by a point, the tangent space at that point, and a radius; since the tangent spaces are the same for both balls, only the radii may differ. But if one radius is smaller than the other, then one ball will be entirely contained within the other, and therefore will not be maximal. So we conclude that only one maximal ball, at a radius $r(\mathbf{x})$, may touch $\mathbf{x}$.

Step 2: Since only one ball is associated with each $\mathbf{x} \in A$, the function $s : A \to M(B)$ which maps each point to the center of its maximal ball is well-defined. We now prove the continuity of this function.

Let $\mathbf{n}(\mathbf{x})$ be the inward directed unit normal of A at $\mathbf{x}$ (this normal is well defined and continuous since the tangent space exists and is continuous at $\mathbf{x}$). Then clearly we can define the function $\mathbf{s}$ by

$$\mathbf{s}(\mathbf{x}) = \mathbf{x} + r(\mathbf{x})\mathbf{n}(\mathbf{x}) \tag{3.10}$$

where $r(\mathbf{x})$ is the radius of the maximal ball touching $\mathbf{x}$. The function $\mathbf{s}$ may also be considered as a *variable distance offset* of the boundary. Obviously, if $r(\mathbf{x})$ is continuous on A then $\mathbf{s}$ will be continuous as well. Now, it is easy to see that the radius function *defined on* $M(B)$ is continuous; the function defined on B which expresses the minimum distance to the boundary ∂B is continuous, as Wolter shows in [87], and the radius function is merely the restriction of this function to the domain $M(B)$ which must be continuous as well since the restriction of a continuous function is continuous.

However, this result does not necessarily imply that the function $r : A \longrightarrow \Re$ is continuous, since it is defined on A rather than on $M(B)$. The function r is not a restriction of a distance function, so we will need to prove its continuity differently. So, pick a point $\mathbf{x}$ on A, and pick a small $\varepsilon > 0$. To prove continuity, we will need to show that given these two conditions, there exists a δ such that for all $\mathbf{y} \in A$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$, $|r(\mathbf{y}) - r(\mathbf{x})| < \varepsilon$. (Here $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$.)

There are two criteria which define a maximal ball inside a domain. The first is that the ball must be contained in the domain, and the second is that it cannot be contained in any other ball contained in the domain. As a result, our proof will have two parts. The first criterion will result in a proof that for all $\mathbf{y} \in A$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$, $r(\mathbf{y}) < r(\mathbf{x}) + \varepsilon$. The second criterion will result in a proof that for all $\mathbf{y} \in A$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$, $r(\mathbf{y}) > r(\mathbf{x}) - \varepsilon$. Clearly, these two results taken together satisfy the conditions for continuity.

Step 3: In this step, we show that $r(\mathbf{y}) < r(\mathbf{x}) + \varepsilon$ for all $\mathbf{y} \in A$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$. There are two possibilities for the point $\mathbf{s}(\mathbf{x})$; either it has two or more minimal joins to the boundary (case A), or it is a limit point of a sequence of points with two or more minimal joins to the boundary (case B) (see Wolter [88]).

Case A: Let us for the moment assume that it has at least two minimal joins to the boundary. Then one of the minimal joins is the line segment $l_{\mathbf{x}}$ joining $\mathbf{s}(\mathbf{x})$ to $\mathbf{x}$, and there is another minimal join $l_{\mathbf{x}'}$ joining $s(\mathbf{x})$ to some other point $\mathbf{x}' \in \partial B$ (see Figure 3-7). (Note that $\mathbf{x}'$ is not required to be in A; however, this does not affect the proof, as we shall see.)

Now, the three points $\mathbf{x}$, $\mathbf{s}(\mathbf{x})$, and $\mathbf{x}'$ form an angle of size $\alpha > 0$. Let $\mathbf{p}_\mathbf{x}$ be the nearest point to $\mathbf{x}'$ on $l_\mathbf{x}$. Using elementary trigonometry, we see that the distance from $\mathbf{p}_\mathbf{x}$ to $\mathbf{x}'$ is given by $r(\mathbf{x})\sin\alpha$, and the distance from $\mathbf{p}_\mathbf{x}$ to $\mathbf{x}$ is given by $r(\mathbf{x})(1-\cos\alpha)$ (see Figure 3-7). Pick some point $\mathbf{y} \in A$ such that $d(\mathbf{x},\mathbf{y}) < \delta$ (we will see how δ is determined in a moment). Construct the segment $l_\mathbf{y}$ as we did with $l_\mathbf{x}$. We now establish the property that $r(\mathbf{y})$ cannot be much larger than $r(\mathbf{x})$, i.e. that $r(\mathbf{y}) < r(\mathbf{x}) + \varepsilon$.

For any $\mathbf{z} \in A$ we can define a line $L_\mathbf{z}$ which is the line passing through $\mathbf{z}$ in the direction of $\mathbf{n}(\mathbf{z})$ as

$$L_\mathbf{z}(t) = \mathbf{z} + \mathbf{n}(\mathbf{z})t. \tag{3.11}$$

Then the distance from $L_\mathbf{z}$ to $\mathbf{x}'$ is simply

$$d_{\mathbf{x}'}(\mathbf{z}) = \min_t |\mathbf{z} + \mathbf{n}(\mathbf{z})t - \mathbf{x}'|. \tag{3.12}$$

The t at which (3.12) achieves its minimum is determined by setting the derivative of $|\mathbf{z} + \mathbf{n}(\mathbf{z})t - \mathbf{x}'|^2$ with respect to t equal to zero:

$$\mathbf{n}(\mathbf{z}) \cdot (\mathbf{z} + \mathbf{n}(\mathbf{z})t_{\min} - \mathbf{x}') = 0 \tag{3.13}$$

Since $\mathbf{n}(\mathbf{z})$ is a unit vector, we obtain

$$t_{\min} = \mathbf{n}(\mathbf{z}) \cdot (\mathbf{x}' - \mathbf{z}) \tag{3.14}$$

Clearly, $t_{\min}$ itself may be considered as a continuous function of $\mathbf{z}$, since $\mathbf{x}'$ is fixed.

Let $\varepsilon' = \frac{\varepsilon}{k}$, where k is a constant depending only on $\mathbf{x}$ and α (the precise definition of k is given below). Define the point $\mathbf{p}_\mathbf{z}$ on $L_\mathbf{z}$ as

$$\mathbf{p}_\mathbf{z} = \mathbf{z} + \mathbf{n}(\mathbf{z})t_{\min}(\mathbf{z}). \tag{3.15}$$

$\mathbf{p}_\mathbf{z}$ is the nearest point to $\mathbf{x}'$ on $L_\mathbf{z}$; obviously it too can be considered as a continuous function of $\mathbf{z}$ since it is the result of addition and multiplication of continuous functions. Therefore there exists a δ such that for any $\mathbf{y} \in \partial B$ such that $d(\mathbf{x},\mathbf{y}) < \delta$, $d(\mathbf{p}_\mathbf{x}, \mathbf{p}_\mathbf{y}) < \varepsilon'$

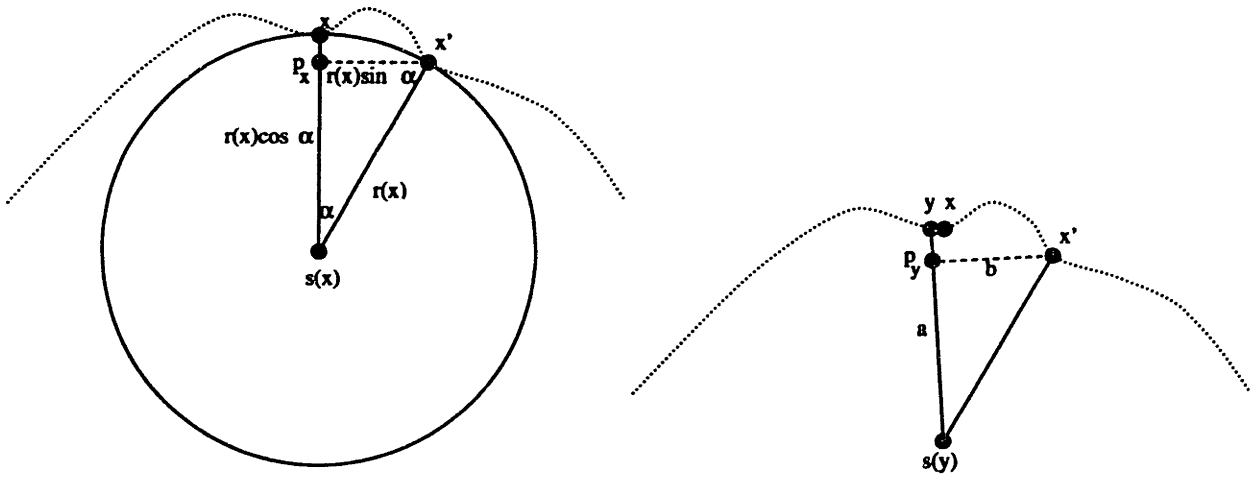

Figure 3-7: The maximal ball at $\mathbf{s(x)}$ touches the boundary at $\mathbf{x}$ and $\mathbf{x}'$. Since $\mathbf{x}$ and $\mathbf{x}'$ are separated by some angle α, we can create a triangle of $\mathbf{x}'$, $s(\mathbf{x})$, and $\mathbf{p_X}$. By moving away from $\mathbf{x}$ on A a small amount to the point $\mathbf{y}$, we can create a new right triangle of $\mathbf{s(y)}$, $\mathbf{x}'$, and $\mathbf{p_y}$. The lengths a and b marked in the new triangle are no more than ε' greater than $r(\mathbf{x})\sin\alpha$ and $r(\mathbf{y}) - r(\mathbf{x})(1-\cos\alpha)$ respectively, by continuity.

and $|t_{\min}(\mathbf{x}) - t_{\min}(\mathbf{y})| < \varepsilon'$. Therefore, by the triangle inequality,

$$d(\mathbf{x}', \mathbf{p_y}) \leq d(\mathbf{x}', \mathbf{p_X}) + d(\mathbf{p_X}, \mathbf{p_y}) < r(\mathbf{x})\sin\alpha + \varepsilon'. \tag{3.16}$$

In addition, the distance from $\mathbf{y}$ to $\mathbf{p_y}$ is equal to $t_{\min}(\mathbf{y}) > t_{\min}(\mathbf{x}) - \varepsilon'$. But

$$t_{\min}(\mathbf{x}) = r(\mathbf{x}) - r(\mathbf{x})\cos\alpha. \tag{3.17}$$

Thus we can get an upper bound for the distance between $\mathbf{s(y)}$ and $\mathbf{p_y}$ as (see Figure 3-7)

$$\begin{aligned} d(\mathbf{s(y)}, \mathbf{p_y}) &= r(\mathbf{y}) - t_{\min}(\mathbf{y}) \\ &< r(\mathbf{y}) - r(\mathbf{x})(1-\cos\alpha) + \varepsilon'. \end{aligned} \tag{3.18}$$

Now, the distance from $\mathbf{s(y)}$ to $\mathbf{x}'$ must be greater than or equal to $r(\mathbf{y})$:

$$r(\mathbf{y}) \leq d(\mathbf{s(y)}, \mathbf{x}') \tag{3.19}$$

or else the maximal ball at $\mathbf{s(y)}$ will no longer be contained in B.

But if we construct the triangle formed by $\mathbf{s(y)}, \mathbf{p_y}$, and $\mathbf{x}'$ (see Figure 3-7), the

Pythagorean theorem combined with equations (3.16) and (3.18) shows that

$$d(\mathbf{s}(\mathbf{y}),\mathbf{x}') \leq \sqrt{[r(\mathbf{y}) - r(\mathbf{x})(1-\cos\alpha) + \varepsilon']^2 + [r(\mathbf{x})\sin\alpha + \varepsilon']^2}. \tag{3.20}$$

Using (3.19), the square of the right hand side of (3.20) must be greater than or equal to $r^2(\mathbf{y})$, so after simplifying we obtain

$$\begin{aligned} r^2(\mathbf{y}) + r^2(\mathbf{x})(\cos^2\alpha - 2\cos\alpha + 1) - 2r(\mathbf{y})r(\mathbf{x})(1-\cos\alpha) + r^2(\mathbf{x})\sin^2(\alpha) \quad &+ \\ 2\varepsilon'[r(\mathbf{y}) - r(\mathbf{x})(1-\cos\alpha) + \varepsilon' + r(\mathbf{x})\sin\alpha] \quad &\geq \quad r^2(\mathbf{y}). \end{aligned} \tag{3.21}$$

The function r is clearly bounded above by some number R over the set ∂B since B is compact and therefore ∂B is of finite extent. Assume that $\varepsilon' < 1$; then we can define the constant k by

$$kr(\mathbf{x})(1-\cos\alpha) = R - r(\mathbf{x})(1-\cos\alpha) + r(\mathbf{x})\sin\alpha + 1 \tag{3.22}$$

With this definition in mind, $2kr(\mathbf{x})(1-\cos\alpha)\varepsilon'$ is a strict upper bound on the ε' term in equation (3.21) dependent only on $\mathbf{x}$ and α. Subtracting $r^2(\mathbf{y})$ from both sides of (3.21), substituting in the k term, and further simplifying gives

$$2r^2(\mathbf{x})(1-\cos\alpha) - 2r(\mathbf{y})r(\mathbf{x})(1-\cos\alpha) + 2kr(\mathbf{x})(1-\cos\alpha)\varepsilon' \geq 0. \tag{3.23}$$

Note that $2r(\mathbf{x})(1-\cos\alpha)$ is strictly greater than zero; $\mathbf{x}$ has at least two minimal joins, so both α and $r(\mathbf{x})$ are strictly greater than zero. Therefore, after dividing (3.23) by $2r(\mathbf{x})(1-\cos\alpha)$, we conclude that when $d(\mathbf{x},\mathbf{y}) < \delta$,

$$r(\mathbf{y}) \leq r(\mathbf{x}) + \varepsilon \tag{3.24}$$

since $\varepsilon = k\varepsilon'$. Thus far, equation (3.24) only holds for $\mathbf{x}$ which have an $\mathbf{s}(\mathbf{x})$ with two or more minimal joins to the boundary.

Case B: Suppose, on the other hand, that $\mathbf{s}(\mathbf{x})$ has only one minimal join to the boundary. Then $\mathbf{s}(\mathbf{x})$ is the limit point of a sequence of points with two or more minimal joins [88] and it is therefore possible to find a point $\mathbf{z} \in A$ such that $d(\mathbf{s}(\mathbf{z}),\mathbf{s}(\mathbf{x})) < \frac{\varepsilon}{4}$ and $\mathbf{s}(\mathbf{z})$ has minimal joins to the boundary at the points $\mathbf{z}$ and $\mathbf{z}'$. (Note that we can force $\mathbf{z}$ to be

in A since A is open in ∂B, and therefore all sufficiently small neighborhoods of $\mathbf{x}$ will be contained in A). By the triangle inequality,

$$d(\mathbf{s}(\mathbf{x}), \mathbf{z}') \leq d(\mathbf{s}(\mathbf{x}), \mathbf{s}(\mathbf{z})) + d(\mathbf{s}(\mathbf{z}), \mathbf{z}') < r(\mathbf{z}) + \frac{\varepsilon}{4} \tag{3.25}$$

But because $d(\mathbf{s}(\mathbf{z}), \mathbf{s}(\mathbf{x})) < \frac{\varepsilon}{4}$, $|r(\mathbf{z}) - r(\mathbf{x})| \leq \frac{\varepsilon}{4}$. This result holds by the definition of r; $r(\mathbf{z})$ is the distance of $\mathbf{s}(\mathbf{z})$ to ∂B, and thus the distance of $\mathbf{s}(\mathbf{x})$ to ∂B cannot differ by more than $\frac{\varepsilon}{4}$. Therefore

$$d(\mathbf{s}(\mathbf{x}), \mathbf{z}') < r(\mathbf{x}) + \frac{\varepsilon}{2}. \tag{3.26}$$

Now we can use the same reasoning we have already used. The only difference is that instead of starting with the triangle of $\mathbf{p_x}, \mathbf{s}(\mathbf{x})$, and $\mathbf{x}'$, and moving to the triangle $\mathbf{p_y}, \mathbf{s}(\mathbf{y})$, and $\mathbf{x}'$ within ε' of the original triangle, we start with the triangle $\mathbf{p_x}, \mathbf{s}(\mathbf{x})$, and $\mathbf{z}'$ and move to the triangle $\mathbf{p_y}, \mathbf{s}(\mathbf{y})$, and $\mathbf{z}'$ within $\frac{\varepsilon'}{2}$. The result is again equation (3.24).

Step 4: We are only halfway to a proof of continuity at this point. We have proven that in a small neighborhood of every point $\mathbf{x} \in A$, r cannot get much greater than $r(\mathbf{x})$; however, we have not yet shown that it can't get much less than $r(\mathbf{x})$ either. In order to prove this, we will need the piecewise C^2 continuity condition for the first time in this lemma.

The key to this part of the proof is that there are two possibilities for the point $\mathbf{x}$: either there is a neighborhood of $\mathbf{x}$ where the boundary is completely C^2, or there is no such neighborhood. In the latter case, $\mathbf{x}$ will be a common point of a finite number of adjoining C^2 pieces of the boundary. Imagine, for either case, taking any plane containing $\mathbf{x}$ and $\mathbf{s}(\mathbf{x})$, and intersecting this plane with the boundary (this is a generalization of the normal section idea we used in three dimensions). The cross-section of boundary we obtain may not be curvature continuous at $\mathbf{x}$ in the second case; however, in either case there will certainly be a left limit and a right limit to the curvature function as we approach $\mathbf{x}$ from the left or right respectively. In any event, whether or not there is a neighborhood of $\mathbf{x}$ where the boundary is C^2, it will still be true that for all neighborhoods smaller than a certain size, $\mathbf{x}$ can be joined to any other point $\mathbf{y} \in A$ in the neighborhood by a C^2 path in the plane of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{s}(\mathbf{x})$.

So pick an arbitrary point $\mathbf{y} \in A$ within a small neighborhood of $\mathbf{x}$. We will see how small to make this neighborhood in a moment. Let $\mathbf{s}(\mathbf{y})$ be the center of the associated

maximal ball at $\mathbf{y}$, and let $r(\mathbf{y})$ be the radius function at $\mathbf{y}$. We want to show that $r(\mathbf{y})$ is close to $r(\mathbf{x})$; in order to do this, we need to show that all balls tangent at $\mathbf{y}$ are contained in B as long as the radii of such balls are a little smaller than $r(\mathbf{x})$. In order to show that the ball is contained in B, we only need to show that it is *locally* contained within B. This is because the majority of the sphere tangent at $\mathbf{y}$ will be contained within the sphere tangent at $\mathbf{x}$. This is fairly obvious, but an analogy will help to clarify the situation. Suppose the ball is tangent at $\mathbf{x}$ and we take another ball of a smaller radius also tangent at $\mathbf{x}$. Initially the smaller ball is contained in the larger ball, and as we roll the smaller ball along the boundary (which we can do because the tangent plane changes continuously) a small amount of the smaller ball emerges from the larger. The size of this small amount increases as we proceed with rolling, but clearly for small rolling distances, the only part of the small ball outside the larger ball is a small area near the point of tangency.

Now we only need to prove that given an $\varepsilon > 0$, there exists δ such that for any $\mathbf{y}$ in $\{\mathbf{y} \in A : d(\mathbf{x}, \mathbf{y}) < \delta\}$, the ball of radius $r(\mathbf{x}) - \varepsilon$ is locally contained in B. Accordingly, consider any plane P containing $\mathbf{x}$ and $\mathbf{s}(\mathbf{x})$. The plane P intersects the boundary ∂B and the ball tangent to $\mathbf{x}$ as shown in figure 3-8. The intersection of P with the ball tangent to $\mathbf{x}$ must be C_P, a circle of radius $r(\mathbf{x})$ since the plane passes through the center of the ball. A curve R_P results from the intersection of the plane with ∂B. The curve must lie outside the circle since the ball is contained in B.

In order for C_P to lie inside R_P, the radius of curvature (or each of the two limits of the radius of curvature, approaching from the left or right respectively) must be greater than or equal to $r(\mathbf{x})$ by lemma 3.7. Consider a point $\mathbf{y}$ in the neighborhood of $\mathbf{x}$ which also lies in P. If $\mathbf{y}$ is sufficiently close to $\mathbf{x}$, then the path from $\mathbf{x}$ to $\mathbf{y}$ will be C^2; therefore, the radius of curvature of R_P for all $\mathbf{y}$ within some distance δ_P to $\mathbf{x}$ will be strictly greater than $r(\mathbf{x}) - \varepsilon$. Let δ_P be the the greatest lower bound on the set of values for which this bound on the radius of curvature holds. We claim that the greatest lower bound of δ_P over all possible P is some $\delta > 0$. To see this, note that $\mathbf{x}$ lies at the intersection of a finite number of C^2 manifolds $M_1, \ldots, M_n$ (possibly only one such manifold, in which case the neighborhood of $\mathbf{x}$ is completely C^2). In each manifold M_i, the curvature along any planar section of M_i through $\mathbf{x}$ is continuous. Furthermore, the manifold may be sectioned so that the planar sections are related in a continuous manner (in 3-D, for example, we can rotate a normal sectioning plane 360 degrees around the normal, and the normal section curve

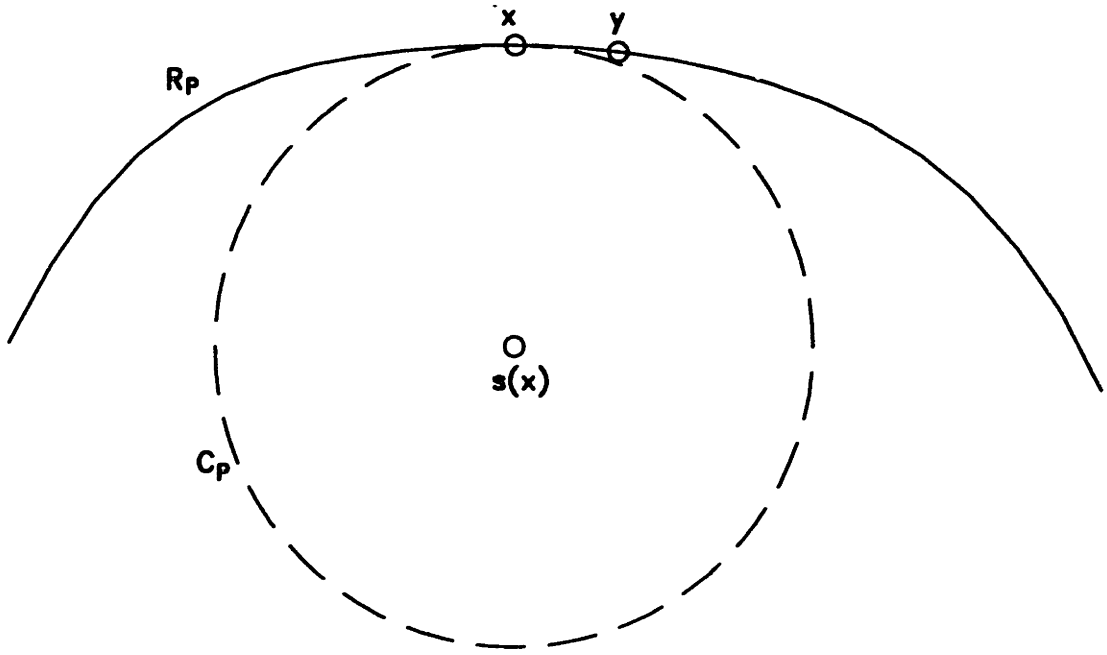

Figure 3-8: The intersection of a plane P with B near $\mathbf{x}$. R_P is the intersection of P and ∂B and C_P is the intersection of P and the maximal ball centered at $\mathbf{s}(\mathbf{x})$. Nearby, at $\mathbf{y}$, a circle of slightly smaller radius is also inside B. Also, most of the second circle is contained in the first.

will deform continuously as we rotate). Thus for each manifold there is some δ_i such that in any sectioning plane P the radius of curvature changes less than ε. Of course, taking $\delta = \min_i\{\delta_i\}$ will give a δ completely independent of P.

Thus for any $\varepsilon > 0$, there exists a δ such that for any plane P and any point $\mathbf{y} \in P \cap A$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$, the radius of curvature at $\mathbf{y}$ is greater than $r(\mathbf{x}) - \varepsilon$. Thus at $\mathbf{y}$ we can construct a circle of radius $r(\mathbf{x}) - \varepsilon$ which fits within R_P in a neighborhood of size δ around $\mathbf{x}$. Now in order for such a circle to be globally inside R_P, the majority of the circle must lie inside C_P. From the local containment argument above, we know that there is a δ' such that picking a $\mathbf{y}$ inside a $\delta'-$neighborhood of $\mathbf{x}$ will make the circle at $\mathbf{y}$ contained within C_P except inside the $\delta-$neighborhood of $\mathbf{x}$. Take the minimum of δ and δ' and assign it to δ as an allowable neighborhood size.

Now we know that in any sectioning plane, a circle of radius $r(\mathbf{x}) - \varepsilon$ will be contained in B as long as we stay within a neighborhood of size δ around $\mathbf{x}$. The cross-section of any ball of radius $r(\mathbf{x}) - \varepsilon$ can have a radius no greater than $r(\mathbf{x}) - \varepsilon$. Therefore it must be true that a ball of radius $r(\mathbf{x}) - \varepsilon$ will be contained in B within a neighborhood of size δ around $\mathbf{x}$. To see this, note that every point of such a ball is contained in some P, and we have already shown that within P containment holds.

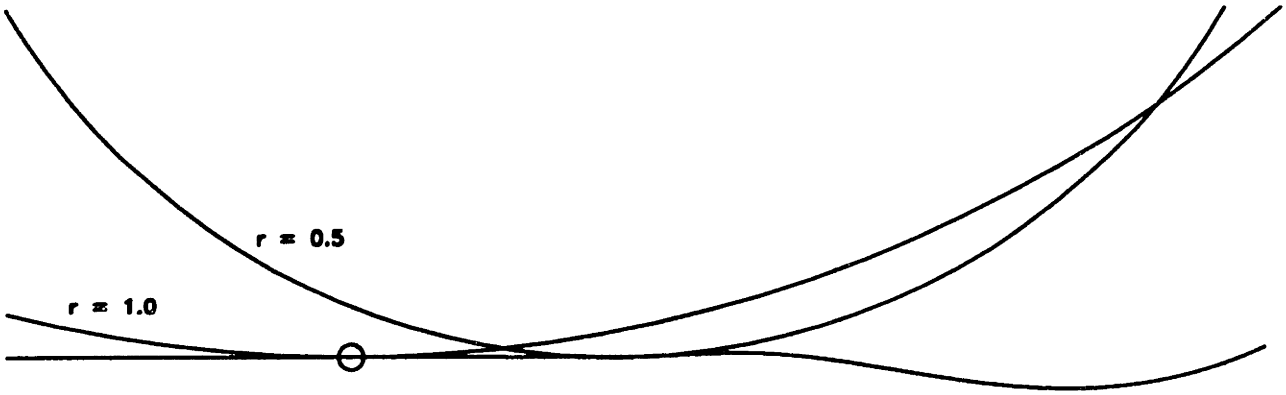

Figure 3-9: The bottom boundary of the two-dimensional region B. The radius function is approaching 0.5 as we approach the origin from the right. However, at the origin, a circle of radius 1 fits inside B, so there is a discontinuity in the radius function.

Since the ball of radius $r(\mathbf{x}) - \varepsilon$ tangent to some point $\mathbf{y}$ within a distance δ of $\mathbf{x}$ is contained within B, the maximal ball at y must have a radius of at least this value. Thus we have shown that $r(\mathbf{y}) \geq r(\mathbf{x}) - \varepsilon$ for all $\mathbf{y}$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$.

Step 5: Let δ equal the minimum of the δs from steps 3 and 4. We can now conclude that for any $\varepsilon > 0$ and any $\mathbf{x} \in A$ there exists a δ such that for all $\mathbf{y}$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$, $|r(\mathbf{y}) - r(\mathbf{x})| \leq \varepsilon$ and therefore r is continuous. Since r is continuous, so is $\mathbf{s}$ by (3.10). □

Note: We need piecewise C^2 continuity for lemma 3.10 to be correct. We present an example of a 2-D region whose boundary is C^1 everywhere and C^2 everywhere except at the origin (and not piecewise C^2, as we shall see). Because it is not completely C^2, we will see that r is not continuous at the origin and therefore neither is $\mathbf{s}$. This example is modified from Wolter [88]. Consider a region B with a bottom defined by a line segment from $(0,0)$ to $(-10,0)$ and for $x > 0$ up to $x = 10$, by the curve $(x, f(x))$, where $f(x) = \frac{1}{2}x^4 \sin\frac{2}{x}$ (see figures 3-9 and 3-10). f' and f'' may be computed as $f'(x) = 2x^3 \sin\frac{2}{x} - x^2 \cos\frac{2}{x}$ and $f''(x) = 6x^2 \sin\frac{2}{x} - 6x\cos\frac{2}{x} - 2\sin\frac{2}{x}$. We assume that beyond $x = 10$ and $x = -10$ the boundary is completely C^2 and stays at least 10 units away from the origin. (One could construct such a boundary with a cubic spline that maintains continuity up to second order over its length and is furthermore second order continuous at $x = 10$ and $x = -10$.) Notice that ∂B is completely C^2 everywhere except at the origin, where it is only C^1. Furthermore, since no limit exists to $f''(x)$ as $x \to 0_+$, the boundary is not even piecewise C^2.

The curvature of $(x, f(x))$ is given by $\kappa(x) = \frac{f''(x)}{(1+[f'(x)]^2)^{\frac{3}{2}}}$. Now, as we approach the origin from the right, the dominant term of $f''(x)$ will be $-2\sin\frac{2}{x}$ since all other terms drop to 0. Therefore the curvature will have an infinite number of local maxima near the origin.

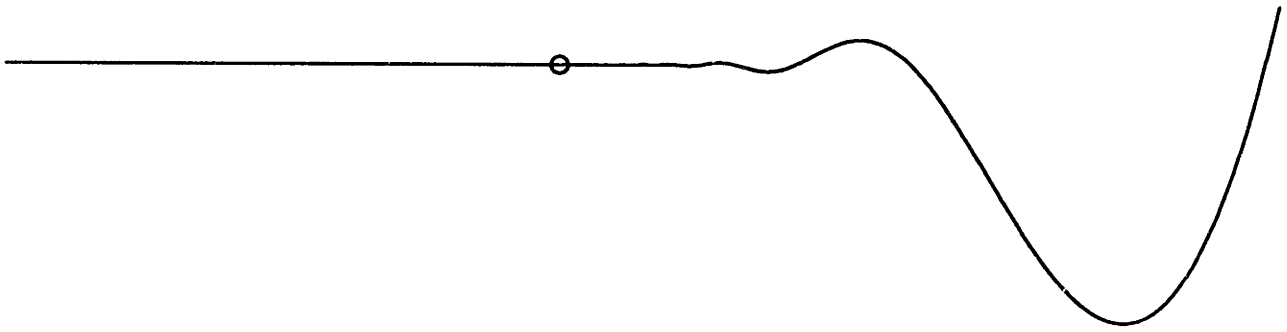

Figure 3-10: The bottom boundary of B again, scaled 10 times in the y-direction.

In addition, the denominator of $\kappa(x)$ approaches 1 since $f'(x) \to 0$, so these local maxima will approach 2 as $x \to 0$. Each local maximum of curvature gives rise to a maximal disc centered at the center of curvature (this is a special case of theorem 3.6 for two dimensions), so there are an infinite number of maximal discs whose centers are converging on the point $(0, \frac{1}{2})$ as $x \to 0_+$. Therefore, the radius function of $M(B)$ is approaching $\frac{1}{2}$ as $x \to 0_+$. But we claim that as we approach 0 from the left, the radius function must be at least 1. To see this, consider a maximal disc centered at $(0,1)$ with radius 1. A Taylor expansion of the function $\sqrt{1-x}$ around $x = 0$ gives $\sqrt{1-x} \leq 1 - \frac{1}{2}x$ in $[0,1]$, so $\sqrt{1-x^2} \leq 1 - \frac{1}{2}x^2$ in $[0,1]$. Thus the following inequalities must hold in $[0,1]$:

$$1 - \sqrt{1-x^2} \geq \frac{1}{2}x^2 \geq \frac{1}{2}x^4 \sin\frac{2}{x}. \tag{3.27}$$

Therefore, the disc of radius 1 centered at $(0,1)$ is completely contained in B and hence the radius function is at least 1 when we approach from the left. So there is a discontinuity in r at the origin and as a result $\mathbf{s}$ is not continuous on ∂B.

Theorem 3.11 Let B be a compact submanifold of $\Re^n$ with a G^1 boundary ∂B which is also piecewise C^2. Then the Medial Axis of B, $M(B)$, is a strong deformation retract of B. In particular, if B is path connected, then $M(B)$ is path connected.

Proof: Define the *projection function* $\pi : B - M(B) \longrightarrow \partial B$ as

$$\pi(\mathbf{x}) = \mathbf{x} - d(\mathbf{x}, \partial B)\nabla d(\mathbf{x}, \partial B) \tag{3.28}$$

π is well-defined and continuous since both d and ∇d are continuous as long as $\mathbf{x}$ is not on the Medial Axis (see Wolter [88]). It assigns to each point of $B - M(B)$ the nearest point

on ∂B (note that there can only be one nearest point outside the Medial Axis). Consider the function $\mathbf{m} : B \longrightarrow M(B)$ defined by

$$\mathbf{m}(\mathbf{x}) = \begin{cases} \mathbf{s}(\pi(\mathbf{x})) & \text{if } \mathbf{x} \in B - M(B) \\ \mathbf{x} & \text{if } \mathbf{x} \in M(B) \end{cases} \tag{3.29}$$

where $\mathbf{s}$ is defined by equation 3.10. Obviously $\mathbf{m}$ is continuous on $B - M(B)$; any point in $B - M(B)$ has a neighborhood in $B - M(B)$ since $B - M(B)$ is open in B, and $\mathbf{m}$ is the composition of two continuous functions in that neighborhood by lemma 3.10. It remains only to be shown that $\mathbf{m}$ is continuous on $M(B)$.

That $\mathbf{m}$ is in fact continuous is easy to see. Let $\mathbf{y} \in M(B)$, and let $Y \subset \partial B$ be the set of footpoints of $\mathbf{y}$. Let r be the radius of the maximal ball at $\mathbf{y}$. Take a ball of radius δ around each footpoint, and let U_δ be the union of all of these balls intersected with the boundary ∂B. U_δ is open in ∂B, so the complement $\partial B - U_\delta$ is closed and therefore compact. Let $r' = d(\mathbf{y}, \partial B - U_\delta)$. Since r' is the minimum of the continuous distance function on the compact set $\partial B - U_\delta$, the minimum is achieved at some point $\mathbf{z}$ on $\partial B - U_\delta$. Since $\mathbf{z}$ is not a footpoint of $\mathbf{y}$, $r' > r$. Let $r' - r = 2\delta'$. Let $\mathbf{x} \in B_{\delta'}(\mathbf{y})$; then $d(\mathbf{x}, \partial B) < r + \delta'$. Furthermore, the nearest point(s) to $\mathbf{x}$ must all lie within U_δ since the distance from $\mathbf{x}$ to $\partial B - U_\delta$ must be greater than or equal to $r + \delta'$ by the triangle inequality. Now, if $\mathbf{x} \in M(B)$, then $d(\mathbf{m}(\mathbf{x}), \mathbf{m}(\mathbf{y})) < \delta'$. On the other hand, if $\mathbf{x} \notin M(B)$, then there is a unique closest point $\pi(\mathbf{x}) = \mathbf{x}'$ on ∂B which lies in U_δ. There is at least one footpoint $\mathbf{y}' \in Y$ of $\mathbf{y}$ within δ of $\mathbf{x}'$, by the definition of U_δ.

Now recall that $\mathbf{s}$ is continuous on ∂B by Lemma 3.10. Since ∂B is compact, $\mathbf{s}$ must also be *uniformly continuous* (see Munkres [54]). That is, for any $\varepsilon > 0$, there exists a δ such that for *any* pair of points $\mathbf{x}, \mathbf{y} \in \partial B$ such that $d(\mathbf{x}, \mathbf{y}) < \delta$, $d(\mathbf{s}(\mathbf{x}), \mathbf{s}(\mathbf{y})) < \varepsilon$. Therefore, pick an $\varepsilon > 0$ and use the δ specified in the uniform continuity condition to make the set U_δ. Since $\mathbf{x} \in B_{\delta'}(\mathbf{y})$ is not in $M(B)$, $\mathbf{m}(\mathbf{x}) = \mathbf{s}(\mathbf{x}')$. But $d(\mathbf{s}(\mathbf{x}'), \mathbf{s}(\mathbf{y}')) < \varepsilon$ since $d(\mathbf{x}', \mathbf{y}') < \delta$. Since $\mathbf{m}(\mathbf{x}) = \mathbf{s}(\mathbf{x}')$ and $\mathbf{m}(\mathbf{y}) = \mathbf{y} = \mathbf{s}(\mathbf{y}')$, we conclude that $\mathbf{m}$ is continuous at $\mathbf{y}$ and therefore continuous over all of B.

Consider the function $H : B \times I \longrightarrow B$ defined by

$$H(\mathbf{x}, t) = \mathbf{x} + t(\mathbf{m}(\mathbf{x}) - \mathbf{x}) \tag{3.30}$$

H is composed of arithmetic operators of continuous functions and is therefore continuous. Furthermore, as may be easily verified, it satisfies the three conditions of a strong deformation retraction from B to $M(B)$. Therefore we conclude that $M(B)$ is a strong deformation retract of B.

As shown by Munkres [54], a strong deformation retract preserves path connectivity. Therefore, if B is path connected, then $M(B)$ is as well. □

We will now use our preceding results to prove that the Medial Axis $M(P)$ of a polyhedral solid P without cavities has a connected Medial Axis. To do this, we use the notion of interior offset defined in definition 2.5. From lemma 3.12, we will conclude that the Medial Axis of an interior offset of P is contained in the Medial Axis of P. Furthermore, by the same lemma, any points on the Medial Axis of P with a radius function greater than ϵ will lie on the Medial Axis of the interior offset of P at distance ϵ (which we call P_ϵ). Therefore, for any pair of points on the Medial Axis with a nonzero radius function, there is some interior offset that contains those two points on its Medial Axis. We will show that there is a continuous path on the Medial Axis of the offset joining those two points, and thus the path also lies in $M(P)$. Because every point in $M(P)$ with a zero radius function is a limit point of a sequence of points with nonzero radius functions, the existence of a path between any two points with nonzero radius functions implies that $M(P)$ is path connected (using lemma 3.15).

In order to show that there is a continuous path on the Medial Axis of P_ϵ, we recognize that the boundary of P_ϵ is nearly G^1; in fact, only at convex edges and within ϵ of endpoints of convex edges may the boundary fail to be G^1, and even in these areas the boundary is at least piecewise G^1. We classify points on $M(P_\epsilon)$ into two categories defined below, linkage points and non-linkage points. The set of non-linkage points is proven to be path connected, and finally the set of non-linkage points is shown to be a dense subset of $M(P_\epsilon)$, implying that $M(P_\epsilon)$ is path connected (again using lemma 3.15).

We need three lemmata in our proof; the first, mentioned above, demonstrates that the Medial Axis of an interior offset is contained in the Medial Axis of the progenitor, and is applicable to a wider class of objects than simply polyhedra. The second lemma is more specific to polyhedra and concerns the shape of the locus of points inside P_ϵ which are closest to some given point on the boundary of P_ϵ. The lemma proves that this locus is a line segment (possibly of zero length, for example in the case of a convex vertex). If the

length is nonzero, we can therefore define a normal (and hence a tangent plane) at a point on the boundary of the offset, whether or not this normal is continuous. This result is critical in proving that a linkage point is a limit point of a sequence of non-linkage points. The third lemma shows that when a dense, path-connected subset D of a set S has a certain property which we call the ***limited path property***, then S is path connected. Since we are only showing that a dense subset of the MA is path connected, we shall need this lemma.

Lemma 3.12 Let B be a compact submanifold of $\Re^n$ and let ∂B be its boundary. Let B_ε be the interior offset of B at distance ε. Then $M(B_\varepsilon) \subset M(B)$. Furthermore, if $\mathbf{x} \in M(B)$ has a maximal sphere of radius greater than ε, then $\mathbf{x} \in M(B_\varepsilon)$.

Proof of Lemma 3.12: Let $\mathbf{x} \in M(B_\varepsilon)$ have a maximal ball of radius r associated with it. Then the distance to ∂B_ε is equal to r and thus the distance to ∂B is equal to $r + \varepsilon$. Furthermore, the ball of radius $r+\varepsilon$ is maximal in B; for, if it were contained in some larger ball of radius $r' + \varepsilon$ contained in B, then the ball of radius r' would be contained in B_ε by the definition of the interior offset. In addition, the ball of radius r' would contain the ball of radius r inside B_ε, since reducing the radii of the two balls by ε maintains containment. But we assumed that the ball of radius r was maximal in B_ε, so there is a contradiction. Since the $\mathbf{x}$ chosen above was an arbitrary point of $M(B_\varepsilon)$ (except for some limit points in $M(B_\varepsilon)$), we conclude that $M(B_\varepsilon) \subset M(B)$ since both sets are closed.

Now suppose that $\mathbf{x} \in M(B)$ has a maximal ball of radius r greater than ε. Then by the definition of the interior offset, the distance from $\mathbf{x}$ to ∂B_ε is equal to $r - \varepsilon$. Furthermore, the ball of radius $r - \varepsilon$ around $\mathbf{x}$ is maximal in B_ε; again, if it were contained in some larger ball, then the ball of radius r would have to be contained in a corresponding larger ball, which would be a contradiction. Thus we conclude that $\mathbf{x} \in M(B_\varepsilon)$. □

Lemma 3.13 Let $\mathbf{x}$ be a point on ∂P_ε, where ε is less than half the minimum distance between any two vertices of P. Suppose $\mathbf{y} \in P_\varepsilon$ and $\mathbf{z} \in P_\varepsilon$ are two points lying on minimal joins to $\mathbf{x}$. Then $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are collinear.

Proof of Lemma 3.13: We prove by contradiction. Draw the line segment L_1 from $\mathbf{x}$ to $\mathbf{y}$ and the line segment L_2 from $\mathbf{x}$ to $\mathbf{z}$. Assume that there is an angle $\alpha \neq 0$ between L_1 and L_2. Extend L_1 and L_2 to the boundary ∂P beyond $\mathbf{x}$, so that L_1 intersects ∂P in some point $\mathbf{y}'$ and L_2 intersects in some point $\mathbf{z}'$.

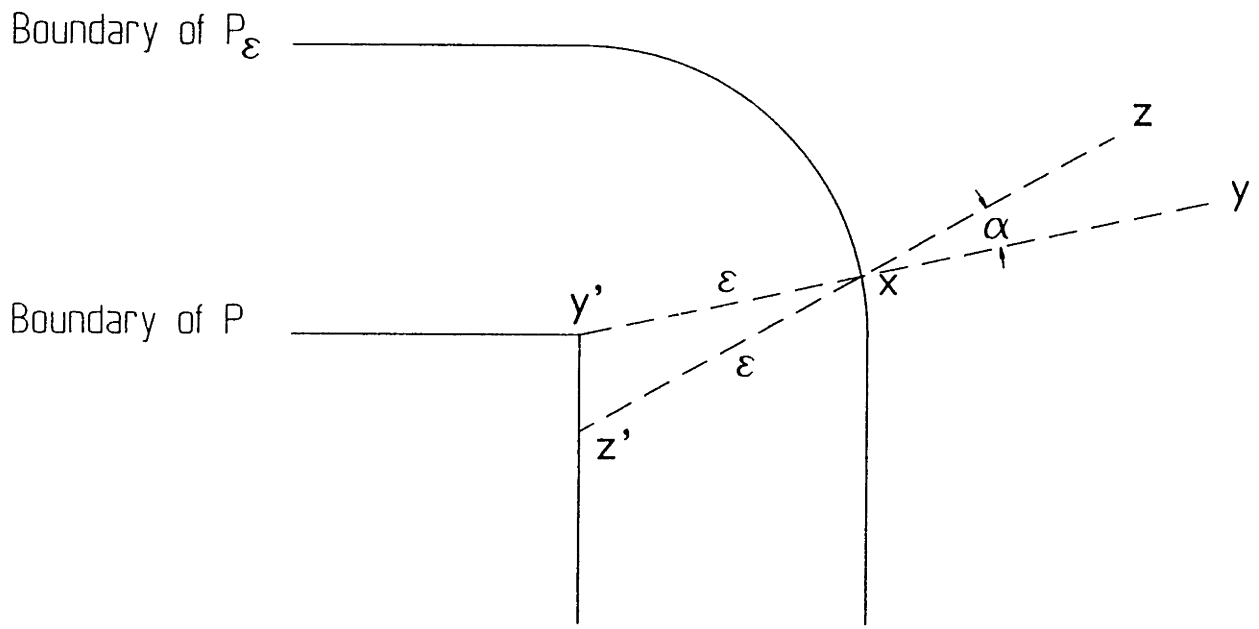

Figure 3-11: Extending the minimal joins from $\mathbf{y}$ and $\mathbf{z}$ shows that $\mathbf{y}$, $\mathbf{z}$, and $\mathbf{x}$ must be collinear, or $\mathbf{y}'$ and $\mathbf{z}'$ will not be closest points to $\mathbf{y}$ and $\mathbf{z}$ respectively.

The distance from $\mathbf{y}$ to $\mathbf{y}'$ must be $d(\mathbf{x},\mathbf{y})+\varepsilon$. If this were not the case (i.e. $d(\mathbf{x},\mathbf{y}') > \varepsilon$), there would be a point $\mathbf{y}'' \neq \mathbf{y}'$ on ∂P such that $d(\mathbf{x},\mathbf{y}'') = \varepsilon$ by the definition of the offset. By the triangle inequality, $d(\mathbf{y},\mathbf{y}'') \leq d(\mathbf{y},\mathbf{x}) + \varepsilon$. Let $\mathbf{x}'$ be a point on ∂P_ε which also lies on the line segment joining $\mathbf{y}$ and $\mathbf{y}'$. By the definition of the offset, $d(\mathbf{x}',\mathbf{y}'') \geq \varepsilon$. But since $d(\mathbf{y},\mathbf{x}') \geq d(\mathbf{y},\mathbf{x})$, we must have $d(\mathbf{y},\mathbf{y}'') \geq d(\mathbf{y},\mathbf{x}) + \varepsilon$. Therefore $d(\mathbf{y},\mathbf{y}'') = d(\mathbf{y},\mathbf{x}) + \varepsilon$. Since equality can only happen when $\mathbf{x},\mathbf{y}$, and $\mathbf{y}''$ are collinear, we conclude that $\mathbf{y}' = \mathbf{y}''$. Thus the distances from $\mathbf{y}$ and $\mathbf{z}$ to ∂P must be given respectively by $d(\mathbf{x},\mathbf{y}) + \varepsilon$ and $d(\mathbf{x},\mathbf{z})+\varepsilon$, and $\mathbf{y}'$ and $\mathbf{z}'$ must be closest points to $\mathbf{y}$ and $\mathbf{z}$ respectively. But an examination of figure 3-11 shows that $\mathbf{y}'$ is closer to $\mathbf{z}$ than $\mathbf{z}'$ and $\mathbf{z}'$ is closer to $\mathbf{y}$ than $\mathbf{y}'$. For example, the triangle inequality forces $d(\mathbf{z},\mathbf{y}') < d(\mathbf{z},\mathbf{x}) + \varepsilon = d(\mathbf{z},\mathbf{z}')$ when $\alpha \neq 0$. The only way that this contradiction will not occur is if $\alpha = 0$. □

Definition 3.14 Let D be a path connected subset of $\Re^n$. We say that D has the ***limited path property*** if for any $\varepsilon > 0$, there exists a $\delta > 0$ such that for any $\mathbf{x},\mathbf{y} \in D$ such that $d(\mathbf{x},\mathbf{y}) < \delta$, there exists a continuous path $\phi : [0,1] \longrightarrow D$ of total length less than ε such that $\phi(0) = \mathbf{x}$ and $\phi(1) = \mathbf{y}$.

Lemma 3.15 Let D be a path connected dense subset of a compact set $S \subset \Re^n$, and suppose D has the limited path property. Then S is path connected.

Proof of Lemma 3.15: Let $\mathbf{x} \in S - D$. Since D is dense in S, every neighborhood of $\mathbf{x}$ must contain a point of D. Let $\delta(\varepsilon)$ be the δ specified by the limited path property for any $\varepsilon > 0$. Inductively create a sequence $\{\mathbf{x}_i\}$ converging to $\mathbf{x}$ as follows:

1. For each positive integer i in succession, construct the open ball of radius $\delta(\frac{1}{2^{i+1}})$ around $\mathbf{x}$.

2. Remove all the points $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}$ from the open ball.

3. Since the resulting set is a neighborhood of $\mathbf{x}$, there exists a point in D inside the set. Let this point be $\mathbf{x}_i$.

Between each $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ there exists a path of length no larger than $\frac{1}{2^i}$ between the two points, since both $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ lie in the ball of radius $\delta(\frac{1}{2^{i+1}})$. Since $[0,1]$ is homeomorphic to any closed interval $[a,b]$, there exists a continuous function $\phi_i : [\sum_{j=1}^{i-1} \frac{1}{2^j}, \sum_{j=1}^{i} \frac{1}{2^j}] \longrightarrow D$ connecting $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$.

Now consider the function $\phi : [0,1] \longrightarrow D$ which is defined as $\phi(\mathbf{x}) = \phi_i(\mathbf{x})$ when $\mathbf{x}$ lies in the domain of definition of ϕ_i. Clearly, ϕ is well-defined everywhere except at 1, and is continuous. Defining $\phi(1) = \mathbf{x}$ makes ϕ continuous because the sequence $\phi(0), \phi(\frac{1}{2}), \phi(\frac{1}{2} + \frac{1}{4}), \ldots$ converges to $\phi(1)$. Furthermore, $\phi([0,1]) \subset S$. Since $\mathbf{x}$ was arbitrary, S must be path connected. □

Theorem 3.16 The Medial Axis of a polyhedral solid in $\Re^3$ (possibly concave, possibly with polyhedral holes but no cavities) with a path connected interior is connected.

Proof: Consider a polyhedron P with a finite number of polyhedral holes. The boundary of P is composed of a connected shell of planar facets with a finite number of convex or non-convex edges. The edges intersect at one of three different types of vertices:

1. *convex vertices*, where only convex edges meet,

2. *reflex vertices*, where only non-convex edges meet, and

3. *mixed vertices*, where a mixture of convex and non-convex edges meet.

Suppose we form a new solid $P_\varepsilon \subset P$ by taking an inward offset of P's boundary at a distance ε and making this offset the boundary of P_ε. The offsetting action smooths

non-convex edges, making them G^1; however it leaves convex edges non-smooth and may result in fairly complicated behavior in the neighborhood of vertices.

To examine what happens around all types of vertices, consider a small sphere of radius ϵ centered on a vertex of P. Trim away any portion of the sphere which lies outside the polyhedron. Next, trim away any portions of the sphere which are closer to the incident faces than to the vertex. Finally, trim away remaining portions of the sphere which are closer to non-convex edges than to the vertex. If any part of the sphere remains, it will be the interior offset of the boundary at distance ϵ due solely to the vertex; furthermore, any points in the interior of the polyhedron which are closer to the vertex than any other point on the boundary must lie in one of the directions remaining on the sphere. Call the remaining portion of the sphere the *normal space at distance* ϵ of the vertex.

Notice first that any convex vertex will have an empty normal space, while reflex vertices always have a non-empty normal space. This can be seen by noting that there are no points in the interior of an object which have a convex vertex as their closest point, whereas at a reflex vertex, there are an infinite number of such points. Figure 3-12 shows the construction of a normal space at a reflex vertex with three non-convex edges incident to it. The sphere in the center of the figure has been trimmed away by the offsets of adjacent faces and edges. Notice the completely G^1 character of the offset for a reflex vertex. Mixed vertices may have either an empty or non-empty normal space; however, note that if any vertex has a non-empty normal space, it is joined in a G^1 fashion to the offset of any non-convex edge adjacent to the vertex (there is always at least one non-convex edge, of course, since the vertex is non-convex). To see this, note that the cylinder of radius ϵ around any non-convex edge, which was part of the set cut out from the sphere around the vertex, is tangent to the sphere. Since the cylinder represents the offset of the non-convex edge, any point in the interior of the normal space may be joined to a point on the offset of the non-convex edge by a G^1 path; furthermore, there is a neighborhood of the path which is G^1 and piecewise C^2.

Now pick two points $\mathbf{p}$ and $\mathbf{q}$ on the Medial Axis of P. Assume that each of these points has at least two footpoints on the boundary (for, if this were not the case for one of the points, it would be a limit point of a sequence of points with two minimal joins, as shown in Wolter [88]). Then both points have maximal balls of non-zero radius greater than some number $\epsilon > 0$. We force ϵ to be significantly smaller than the minimum distance between

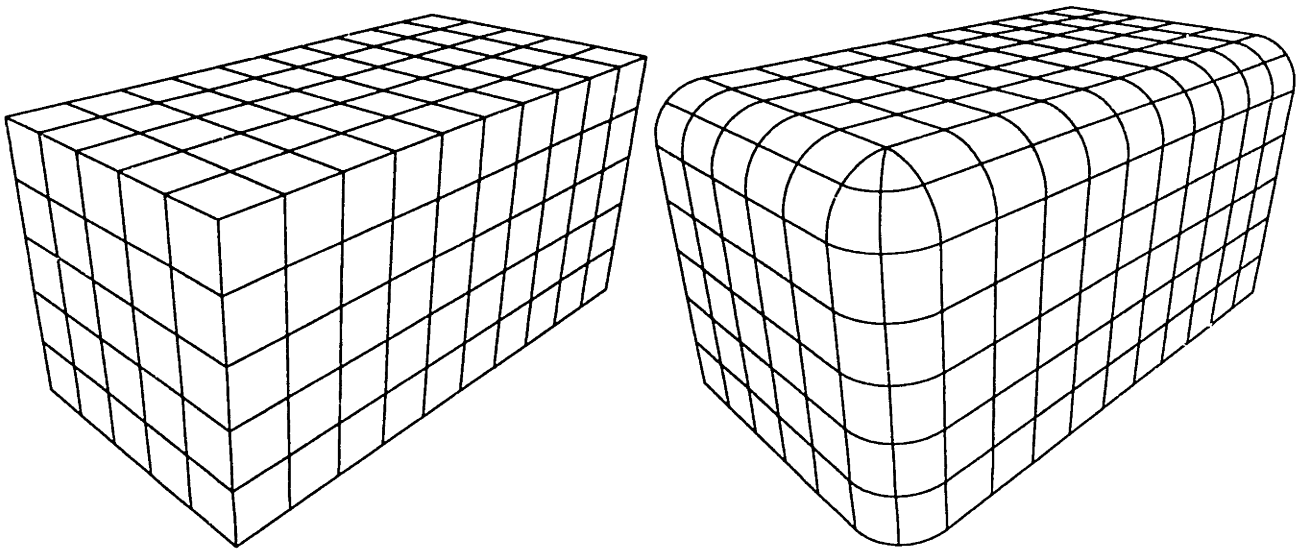

Figure 3-12: Constructing the normal space of a vertex with three non-convex edges incident. A sphere is drawn around the vertex, which has pieces trimmed away by offsets of adjacent edges and faces.

any two vertices, so that the vertices do not interfere with one another when we take an interior offset of P.

By lemma 3.12, we conclude that both $\mathbf{p}$ and $\mathbf{q}$ are on $M(P_\varepsilon)$ as well as $M(P)$. Let $\mathbf{p}'$ be a footpoint of $\mathbf{p}$ and $\mathbf{q}'$ be a footpoint of $\mathbf{q}$ on ∂P_ε. There are two possibilities for the point $\mathbf{p}$: either at least one of its footpoints $\mathbf{p}'$ is in a G^1 neighborhood of ∂P_ε, or all of the footpoints lie on tangent plane discontinuities of ∂P_ε. In the latter case, we call $\mathbf{p}$ a *linkage point.* Let us assume for the moment that neither $\mathbf{p}$ nor $\mathbf{q}$ is a linkage point and that $\mathbf{p}'$ and $\mathbf{q}'$ are footpoints lying in G^1 neighborhoods of ∂P_ε; we will see below that linkage points may be considered as limit points of sequences of non-linkage points. In addition, it is obvious that linkage points are extremely rare, since we require *all* footpoints to lie on tangent discontinuities of a piecewise G^1 boundary.

We can construct a continuous path ϕ on ∂P_ε by moving along a shortest path from $\mathbf{p}'$ to $\mathbf{q}'$, so that we cross a finite number of convex edges. Because $\mathbf{p}'$ and $\mathbf{q}'$ lie in G^1 neighborhoods of ∂P_ε, we can tweak the path slightly so that it moves around all vertices by either staying on a G^1 area of the boundary or crossing a convex edge. Pick a small $\delta > 0$ and split the path into a finite number of sub-paths $\phi_1, \ldots, \phi_n$ which lie in completely G^1 neighborhoods by discarding any areas of the path which lie within δ of a convex edge. Note that each of the ϕ_i are also continuous. By lemma 3.10, along any subpath ϕ_i, the function $\mathbf{s} \circ \phi_i$ is continuous and therefore represents a continuous path on $M(P_\varepsilon)$. Here

$\mathbf{s} : \partial P_\varepsilon \longrightarrow M(P_\varepsilon)$ is the same continuous function we used earlier to map points on the boundary of a manifold to the Medial Axis of the manifold.

Now notice that each path $\mathbf{s} \circ \phi_i$ ends within δ of a convex edge, and the succeeding path $\mathbf{s} \circ \phi_{i+1}$ begins within δ of the same edge; furthermore the ending and starting points of the two paths are within 2δ of one another. Clearly, as $\delta \to 0$, the points converge on the same point on the convex edge, and therefore we conclude that $\mathbf{p}$ and $\mathbf{q}$ may be joined by a continuous path on $M(P_\varepsilon)$.

At this point, all we need to show is that if either $\mathbf{p}$ or $\mathbf{q}$ is a linkage point, it may be expressed as a limit point of non-linkage points. We show this now.

Suppose that $\mathbf{p}$ is a linkage point and that $\mathbf{p}'$ is one of its footpoints on ∂P_ε. Assume without loss of generality that $\mathbf{p}$ has at least two footpoints (for if it did not, it would be a limit point of a sequence of points on $M(P_\varepsilon)$ with at least two footpoints). Then there is no other point on $M(P_\varepsilon)$ with two or more footpoints that has $\mathbf{p}'$ as a footpoint. For, by lemma 3.13, any other such point would have to be collinear with $\mathbf{p}'$ and $\mathbf{p}$ which would imply that one maximal ball would be contained in another. Let $\mathbf{p}''$ be another footpoint of $\mathbf{p}$.

Note that ∂P_ε is piecewise G^1, as is ∂P. Hence $\mathbf{p}'$ lies at the boundary of a finite number of G^1 regions and therefore a finite number of tangent planes may be defined at $\mathbf{p}$ by a limiting argument. But by lemma 3.13, since the line segment from $\mathbf{p}$ to $\mathbf{p}'$ is a minimal join, a unique tangent plane may be defined at $\mathbf{p}'$. Therefore only one of the finite number of tangent planes is the correct one. All but one of the tangent planes are invalid because along most of the directions specified by the tangent planes, there are no minimal joins because of the existence of a convex edge. In the case of a convex vertex, for example, there is no point in the interior of P_ε which has the vertex as its closest point. However, it is possible that one tangent plane generates a set of minimal joins, and lemma 3.13 stipulates that there can be at most one. The existence of $\mathbf{p}$ means that there is exactly one, which also implies that one of the G^1 regions incident to $\mathbf{p}'$ has a tangent plane approaching the tangent plane at $\mathbf{p}'$ in small neighborhoods of $\mathbf{p}'$.

Therefore, there is some G^1 set on ∂P_ε associated with the tangent plane at $\mathbf{p}'$, which we will call A'. Consider a sequence of points $\{\mathbf{p}'_i\}$ in A' converging to $\mathbf{p}'$. Map this sequence to a sequence of points $\{\mathbf{p}_i\}$ on $M(P_\varepsilon)$. As $i \to \infty$, the tangent planes at $\mathbf{p}'_i$ become closer and closer to the tangent plane at $\mathbf{p}'$, so the minimal join from $\mathbf{p}_i$ to $\mathbf{p}'_i$ approaches the

minimal join from $\mathbf{p}$ to $\mathbf{p}'$. Note that the length of these minimal joins cannot get much larger than the distance from $\mathbf{p}$ to $\mathbf{p}'$ or else $\mathbf{p}''$ will lie inside the maximal ball at $\mathbf{p}_i$. In addition, the length of the minimal joins cannot get much smaller than the distance from $\mathbf{p}$ to $\mathbf{p}'$. Imagine shrinking the maximal ball at $\mathbf{p}$ slightly and rolling it to some $\mathbf{p}_i'$ nearby. As in the proof of lemma 3.10, the only portion of the ball outside the maximal ball at $\mathbf{p}$ is a small portion close to $\mathbf{p}'$. But because the boundary is made up of a finite number of pieces of cylinders, spheres, and planes joined in a G^1 fashion (and thus of bounded, if not continuous, curvature), the rolling action will not cause the ball to go outside the boundary. Therefore it will still be contained in P_ε. Thus $\{\mathbf{p}_i\}$ will converge to $\mathbf{p}$.

We have shown that any two non-linkage points may be joined by a path in $M(P_\varepsilon) \subset M(P)$. Furthermore, any linkage point is a limit point of a sequence of non-linkage points. Since $\mathbf{p}$ and $\mathbf{q}$ were arbitrary points on $M(P)$ with nonzero radius values, and since the subset of points with nonzero radius values is dense in $M(P)$, we have shown that a dense subset D of $M(P)$ (the set of non-linkage points) is path connected. Now, D may be expressed as the union of a finite number of sets D_{ij}, where D_{ij} contains every non-linkage point equidistant from boundary elements i and j. Each D_{ij} has the limited path property, being a trimmed plane or trimmed quadric surface, so $\overline{D_{ij}}$ is path connected. But $M(P) = \bigcup \overline{D_{ij}}$, so every point in $M(P)$ lies in at least one of the $\overline{D_{ij}}$. Consequently, every point in $M(P)$ may be joined by a path to any point in $D \subset M(P)$, so $M(P)$ is path connected. □

We conjecture that the result may be extended to polyhedra with cavities, and even that a similar result holds for arbitrary polytopes in $\Re^n$.

Chapter 4

Classification of Medial Axis Points

4.1 Introduction

From the definition of the Medial Axis given in Chapter 2, it is not immediately obvious how we might classify different types of points on the Medial Axis. For example, we might make a distinction between those points which are centers of maximal balls and those which are merely limit points of centers, but this distinction turns out not to be terribly useful.

A considerably more useful scheme for classification is based on the local shape of the Medial Axis around each point. As Brandt demonstrates [11], the local shape of the Medial Axis is intimately related to the number of points on the associated maximal ball which also touch the boundary of the object. The classification scheme we discuss below is an extension of Brandt's scheme which allows us to handle degeneracies in a natural way. With the aid of classification, we will design an algorithm which will trace out the Medial Axis of an object piece by piece.

We consider first the case of convex polyhedra, since most of the commonly occurring types of Medial Axis points appear in the MA of convex polyhedral solids.

4.2 Classification for Convex Polyhedra

Our algorithm for convex polyhedral solids uses a classification of the different types of Medial Axis points which arise in order to generate an adjacency graph of polygons, line

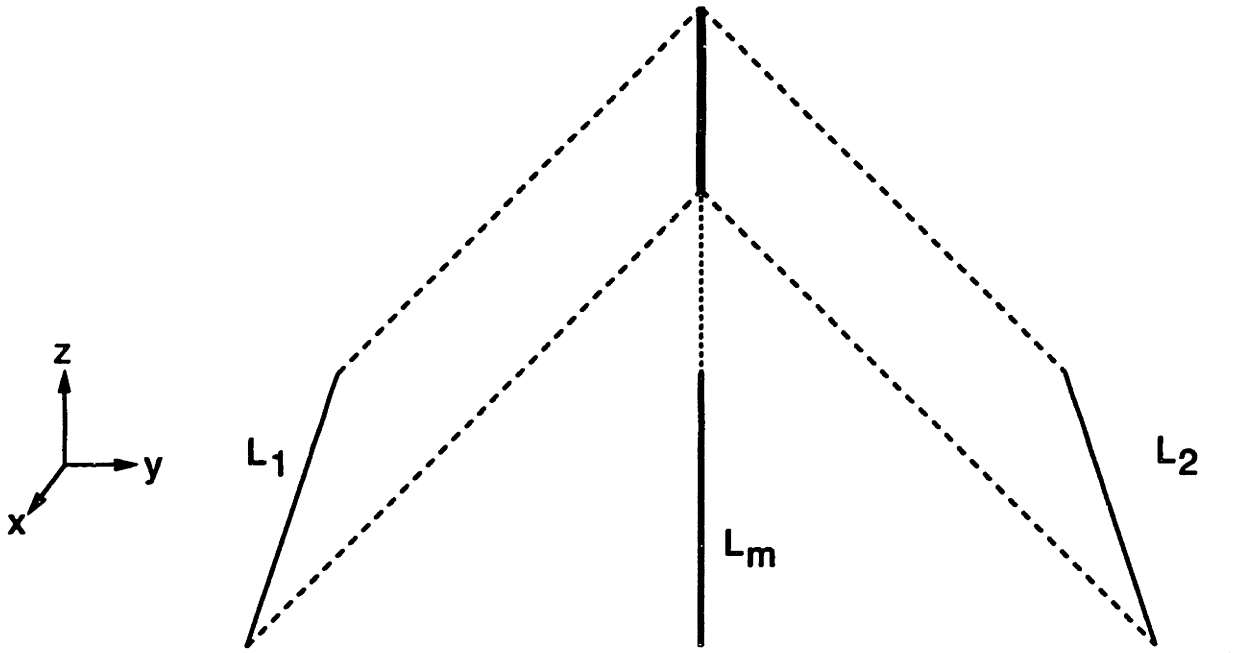

Figure 4-1: The offset sweeps of two line segments in the (x, y)-plane are planes at a 45° angle to the z-axis which intersect in a line segment. The projection of the line segment back onto the (x, y) plane gives the bisector of the two line segments.

segments, and vertices which represents the Medial Axis of the solid. Instead of focusing on points of closest approach, as in Hoffmann [42], we concentrate on *junction points* and *seam-end points*, where the different edges and faces of the MA intersect.

Before describing the algorithm in Chapter 5, we first need to enumerate the possible types of MA points and discuss their relationships with one another. In the following discussion, we shall be considering a convex polyhedral solid P in 3-D with N faces, each face P_i indexed from 1 to N. We disallow polyhedra which have coplanar, adjacent faces, since in this case the two faces may be combined into one.[1] It is instructive in Medial Axis investigations to consider a hypersurface in one dimension higher which represents the union of all possible offsets of a progenitor (see work by Hoffmann [41] and Chiang [19]):

Definition 4.1 Let S be a set in $\Re^k$ with a normal function $\mathbf{N}_S$ defined on it. The *offset sweep* of S, $O(S)$, is a set in $\Re^{k+1}$ defined as follows: A point $(\mathbf{x} + r\mathbf{N}_S(\mathbf{x}), r)$ is a point in $O(S)$ for all real r if and only if $\mathbf{x}$ is in S.

Figure 4-1 shows the offset sweeps of two line segments, and how the intersection of the

[1]Note that in a floating point implementation it may be necessary to specify an angular tolerance on adjacent faces of a polyhedron; if the angle between two adjacent faces is sufficiently close to π, then the two faces are combined into one.

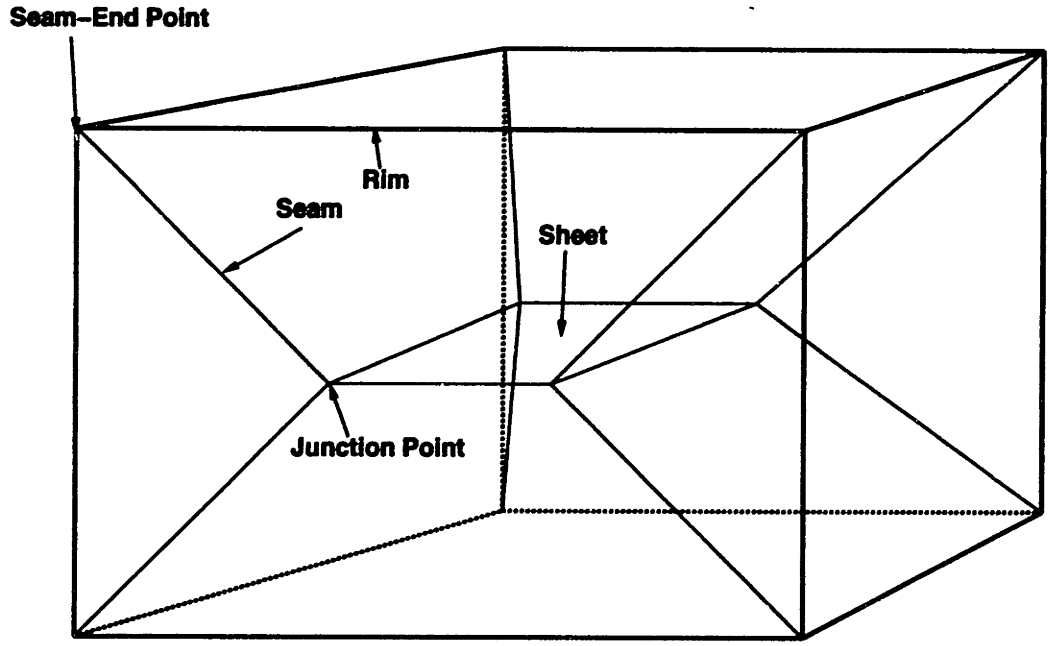

Figure 4-2: Classification of MA points

sweeps determines their bisector. In our discussion, we shall be concerned with the offset sweeps of polygonal faces on the polyhedral boundary, where the normal is constant and directed inside the solid. We shall also be concerned with the correspondence between MA points and these polygonal faces of the boundary, as the following definition demonstrates:

Definition 4.2 Let $\mathbf{p}$ be a point on the MA of a convex polyhedral solid P. Let $B_r(\mathbf{p})$ be the associated maximal ball around p, of radius r. Let $\mathbf{q}_1, \ldots, \mathbf{q}_n$ be the points on the boundary of the solid where $B_r(\mathbf{p})$ is tangent. The $\mathbf{q}_i$ are said to be *footpoints* of $\mathbf{p}$. Let j_i represent the index of the face on which $\mathbf{q}_i$ lies. Then each face P_{j_i} is called a *governor* of $\mathbf{p}$.

Notice that since every face in a polyhedron is planar, there is a one-to-one correspondence between governors and footpoints for each MA point. Armed with these definitions, we can build a classification scheme, closely related to the one used by Brandt [11], to enumerate all the possibilities that may appear in the MA of convex polyhedral solids (see Figure 4-2):

Junction Point: The offset sweeps of all the governors of this type of point must intersect in only one point (the junction point itself). Note that although typically only four surfaces are required, degeneracies may occur; for example in the case of a cube, the point at the center of the cube is a junction point with six governors (all six faces

of the cube) (see Figure 5-4). Notice also that four governors may not be enough to determine a junction point, as a degenerate seam may also have four governors (see Figure 5-3).

Seam Point: For these points, the offset sweeps of their governors must intersect in a line segment. A connected component of seam points is called a *seam*. Note that typically three governors are required, but more may occur under degenerate conditions (Figure 5-3).

Sheet Point: A sheet point is generated when the offset sweeps of its governors intersect in a planar polygon. A connected component of sheet points is called a *sheet*. Two governors are required.

Seam-End Point: These points do not arise from the maximal ball condition of the Medial Axis, but are actually among the limit points of the Medial Axis. A seam-end point results when a seam runs into the boundary of the solid. Since a seam is determined by at least three governors, these points always coincide with the intersection of the governors themselves and therefore are vertices of the convex polyhedron.

End Point: In the case of convex polyhedral solids, these points form the convex edges of the polyhedron, and they result from the intersection of sheets with the boundary. Connected components of end points are called *rims*.

This classification scheme becomes even more useful when we observe relationships between various types of MA points. There are, in fact, obvious relationships between the different classes, which arise due to the configuration of distance vectors to governors in the vicinity of the MA point under consideration. As shown by Brandt [11], the structure of the MA near an MA point can be determined based on the configuration of the point's footpoints. We expand on this idea with the following theorem, which shows us how to glue different pieces of the MA together:

Theorem 4.3 (Classification Theorem) Let $\mathbf{p}$ be an MA point. Suppose $\mathbf{p}$ has n footpoints $\mathbf{q}_1, \ldots, \mathbf{q}_n$ on the polyhedron P and let $P_g = \{P_1, \ldots, P_n\}$ be the set of the corresponding governors. Let $X = \{P_{j_1}, \ldots, P_{j_m}\}$ be a proper subset of P_g where $m \geq 2$. Finally, let $\mathbf{d}_i = \frac{\mathbf{q}_i - \mathbf{p}}{|\mathbf{q}_i - \mathbf{p}|}$ be the unit vector which points from $\mathbf{p}$ to $\mathbf{q}_i$. Then the following conditions are equivalent:

1. For any neighborhood U of $\mathbf{p}$ there exists an $\mathbf{x} \in U$ which lies on the MA and has X as its set of governors.

2. There exists a unit vector $\mathbf{v}$ which is equiangular to the vectors $\mathbf{d}_{j_1}, \ldots, \mathbf{d}_{j_m}$ such that $\mathbf{v} \cdot \mathbf{d}_i < \mathbf{v} \cdot \mathbf{d}_{j_1}$ if $P_i \notin X$ and $\mathbf{v} \cdot \mathbf{d}_i = \mathbf{v} \cdot \mathbf{d}_{j_1}$ if $P_i \in X$.

We will prove this theorem in a moment, but first let us consider its meaning. This theorem forms the core of our algorithm since it tells us precisely what classes of skeleton points are close to a given point. For example, examine figure 4-2. Consider the junction point marked in the figure which has the Front, Bottom, Left, and Top faces as its governors. Using the theorem, we realize that four seams emanate from this point, with governor sets {Front, Bottom, Left}, {Front, Bottom, Top}, {Front, Left, Top}, and {Bottom, Left, Top}. For any one of these four sets, we can find an equiangular vector which satisfies the hypotheses of the theorem. Similarly, all possible $\binom{4}{2} = 6$ subsets of two elements generate sheets adjacent to the junction point since equiangular vectors can be found for each of these cases too. (In fact, an infinite number of such vectors can be found for each of the six sheets since their bisector is a plane.)

In the typical case of a junction point with four governors, in fact, it is true that all $\binom{4}{3}$ three-element subsets generate adjacent seams and all $\binom{4}{2}$ two-element subsets generate adjacent sheets. This is easy to see; in the case of a seam, there is one line that is at the same angle to three planes. One direction on this line makes the fourth surface closer than the other three, and the other direction does the opposite and is therefore the vector that we seek in the theorem. However, in the presence of degeneracies, more care is required. In figure 5-4 we notice that the junction point at the center of the cube has six governors, but instead of there being $\binom{6}{3} = 20$ seams, there are only 8. This is because although equiangular lines exist for all sets of three faces, some of these lines are infeasible since moving in either direction along the line makes some other face closer than the three faces in our subset.

Using our main theorem, it is straightforward to construct an efficient algorithm which moves from piece to piece of the MA, tracing out all possible branches and only visiting each branch once. In Chapter 5, we develop such an algorithm.

Proof of Theorem 4.3: $1 \longrightarrow 2$. Let $B_r(\mathbf{p})$ be the maximal sphere associated with $\mathbf{p}$. Now clearly any face not in P_g is at a distance strictly greater than r; therefore by picking

U sufficiently small we can assure that faces not in P_g will not be a concern to us since they will be too far away for any maximal ball to touch them.

So let $\mathbf{x} \in U$ be on the MA with governor set X, with U being very small. By definition of the MA, $\mathbf{x}$ is equidistant from every face in X. We also know that $\mathbf{p}$, which lies close by, is at a distance r from every face in P_g, and thus equidistant from every face in X.

Consider the vector $\mathbf{v}'$ which starts at $\mathbf{p}$ and ends at $\mathbf{x}$. Each endpoint of this vector is equidistant from every face in X. But the distance from a moving point to a plane varies linearly if the point is moving along a line, so we conclude that the distance to each face in X varies at the same rate as we move from $\mathbf{p}$ to $\mathbf{x}$; therefore the equidistantiality property holds everywhere along the path. But this property precisely characterizes an equiangular vector, so we conclude that $\mathbf{v}'$ is equiangular with respect to the set of faces in X.

Let $\mathbf{v}$ be the unit vector $\frac{\mathbf{v}'}{|\mathbf{v}'|}$. We note that the inner product of $\mathbf{v}$ with any of the $\mathbf{d}_{j_i}$ is the same for any i; this is because since both vectors are unit, the inner product is equal to the cosine of the angle between the vectors, and of course the angles must be the same if $\mathbf{v}$ is equiangular. For any other $\mathbf{d}_i$, where $P_i \notin X$, we note that the angle between $\mathbf{v}$ and $\mathbf{d}_i$ must be strictly greater; for if it were equal, P_i would remain equidistant and if it were less, P_i would become closer than any face in X. Thus we conclude that $\mathbf{v} \cdot \mathbf{d}_{j_1} > \mathbf{v} \cdot \mathbf{d}_i$ in this case.

$2 \longrightarrow 1$. This half of the proof is trivial. We note merely that if we have a $\mathbf{v}$ with the properties listed above, then equidistantiality to faces in X is maintained from $\mathbf{p}$ in the direction of $\mathbf{v}$ for at least a short distance. Also, for faces in P_g but not in X the inner product inequality implies that faces in X are strictly closer than faces not in X as we move from P. Therefore any point $\mathbf{x}$ that is close to $\mathbf{p}$ in the direction of $\mathbf{v}$ is an MA point with X as its set of governors. □

4.3 Classification for Solids with Curved Boundaries

In the more general cases, where the solid may have curved boundaries or nonzero genus, classification may become considerably more complex. In fact, for any closed subset of a maximal sphere, it is possible to construct a solid which has a point on its Medial Axis whose footpoints are precisely that closed subset (see figure 4-3). Fortunately, the most common types of footpoints are the same ones we see in the convex polyhedral case. Because the

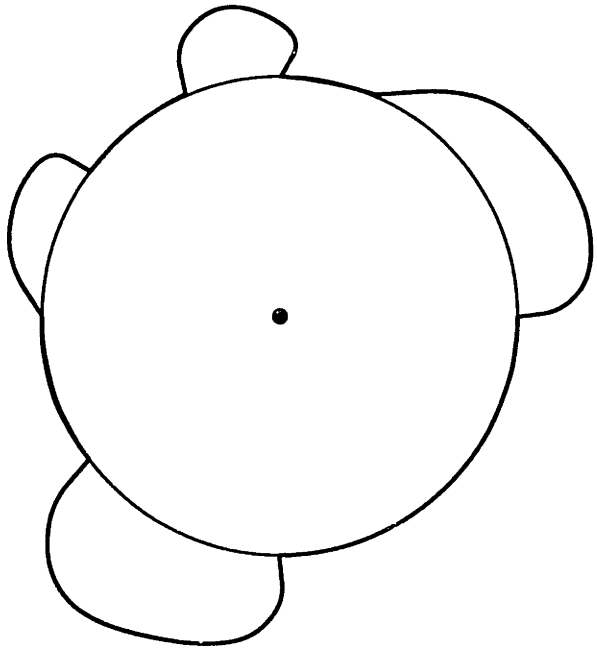

Figure 4-3: Any closed subset of a maximal sphere may constitute the set of footpoints for a given Medial Axis point. This particular maximal disc has an infinite number of footpoints, arranged in a complicated fashion around the circumference of the disc.

surfaces of the solid may not be planar, however, it is possible for boundary elements to intersect in more complicated ways. It may not be immediately obvious how the boundary may be divided up into governors, as it was for convex polyhedra. Therefore, a redefinition of governor is necessary:

Definition 4.4 Let $\mathbf{x}$ be a point on the Medial Axis of B, an n-dimensional submanifold of $\Re^n$. Suppose ∂B is divided into boundary elements $\partial B_1, \ldots, \partial B_m$, each of which is C^2 and compact. Let $\mathbf{x}'$ be any footpoint of $\mathbf{x}$. Then if U is a neighborhood of $\mathbf{x}'$ and $\mathbf{x}' \in \partial B_i$, then $U \cap \partial B_i$ is called a *governor* of $\mathbf{x}$. If U is a ball of radius ε around $\mathbf{x}'$, then $U \cap \partial B_i$ is called a *governor of size* ε of $\mathbf{x}$.

This redefinition of governor reflects the fact that we are concerned with the behavior of the boundary near the footpoints of any point on the Medial Axis. As one might expect, the intersection of the offset sweeps of these governors near a point on the Medial Axis will tell us the behavior of the MA in a neighborhood of the point.

Definition 4.5 Let B be a compact submanifold of $\Re^3$ with a piecewise C^2 boundary divided into boundary elements as before. Let $\mathbf{x} \in M(B)$, and let G_ε be the set of governors of size ε of $\mathbf{x}$. For each element $g \in G_\varepsilon$, let $O(g)$ be the offset sweep of g. Then $\mathbf{x}$ is a

Junction Point if for all sufficiently small ε, $\bigcap_{g \in G_\varepsilon} O(g) = \{\mathbf{x}\}$.

Seam Point if for all sufficiently small ε, $\bigcap_{g \in G_\varepsilon} O(g)$ is a 1-manifold (curve) passing through $\mathbf{x}$ but not ending at $\mathbf{x}$.

Sheet Point if for all sufficiently small ε, $\bigcap_{g \in G_\varepsilon} O(g)$ is a 2-manifold (surface) containing $\mathbf{x}$ in its interior.

Seam-End Point if $\mathbf{x}$ is a limit of a sequence of seam points but not a junction or seam point itself.

End Point if $\mathbf{x}$ is a limit of sheet points but not a junction, seam, sheet, or seam-end point itself.

These definitions are more abstract than the simpler definitions for convex polyhedral solids, but they are a generalization of the polyhedral scheme. From the definitions, we can see that Medial Axis sheets correspond to surfaces; seams correspond to curves, usually where surfaces meet; junction points correspond to isolated points usually occurring at the meeting of seams; and seam-end points and end points are limits of seams and sheets respectively, usually where footpoints on the boundary draw closer and closer to each other until they coincide. Of course, the relationships between different classes of points need not always hold as they did in the polyhedral case (for instance, the axis of a right circular cylinder, which forms part of its Medial Axis, does not result from the intersection of two sheets). But the relationships do hold for the vast majority of points, and it is possible to treat other types of points independently and connect them up to the Medial Axis. We consider such problems later, in chapter 7.

4.4 Governing Equations for Different MA Points

When the boundary ∂S of the solid S may be described parametrically, it is possible to generate systems of equations which govern sheets, seams, and junction points. Suppose, therefore, we have a point $\mathbf{p}$ on the Medial Axis of S with footpoints $\mathbf{f}_1, \ldots, \mathbf{f}_n$. Suppose that within disjoint neighborhoods of each $\mathbf{f}_i$, the boundary ∂S is described by the vector

equation $[x\ y\ z]^T = \mathbf{F}^i(u_i, v_i)$. Then the offset sweep of each $\mathbf{F}^i$ is given by

$$\mathbf{O}^i(u_i, v_i, r) = (\mathbf{F}^i(u_i, v_i) + r\mathbf{N}^i(u_i, v_i), r), \tag{4.1}$$

where $\mathbf{N}^i$ is the unit normal at (u_i, v_i). It is possible to express the locus of points equidistant from neighborhoods of the footpoints as the simultaneous solution of the system

$$\mathbf{O}^1(u_1, v_1, r) = \mathbf{O}^2(u_2, v_2, r) = \ldots = \mathbf{O}^n(u_n, v_n, r), \tag{4.2}$$

which is a system of $3(n-1)$ equations in the $2n+1$ unknowns $u_1, v_1, u_2, v_2, \ldots, u_n, v_n, r$. (The fourth coordinate of each O^i is always r, so we discard the useless condition $r = r$ that appears in each equation.)

When the $\mathbf{F}^i$ are polynomial (e.g. because the boundary is expressed with Bézier or B-spline patches) then the $\mathbf{O}^i$ need not be polynomial, since the $\mathbf{N}^i$ generally contain square roots. It may therefore be desirable, for example in the case of determining junction points, to reformulate the problem in a system containing only polynomial equations. Then the system may be solved with a nonlinear polynomial equation solver (see for example Sherbrooke and Patrikalakis [77] and Maekawa and Patrikalakis [51]).

In order to generate polynomial equations, we make use of the *envelope equation* as Gursoy did [35] for the 2-D MAT problem. The envelope of a set A at radius r is the union of spheres of radius r with centers at every point of A. For a C^1 region of a surface described parametrically by $[x\ y\ z]^T = \mathbf{F}^i(u_i, v_i)$, the envelope equation describes the boundary of this union and is given by

$$\begin{aligned} D(u_i, v_i, x, y, z) = ([x\ y\ z]^T - \mathbf{F}(u_i, v_i))^2 &= r^2 \\ \frac{\partial D}{\partial u_i} &= 0 \\ \frac{\partial D}{\partial v_i} &= 0. \end{aligned} \tag{4.3}$$

The first of the three equations specifies that each point on the envelope lies at a distance r from the center of its associated sphere. The second and third equations state that the tangent plane to the envelope is parallel to the tangent plane to F^i at the center of the

associated sphere (u_i, v_i). To see this, differentiate D with respect to u_i to obtain

$$\frac{\partial D}{\partial u_i} = ([x\ y\ z]^T - \mathbf{F}^i(u_i, v_i)) \cdot \frac{\partial F^i}{\partial u_i} = 0. \tag{4.4}$$

A similar equation results from differentiating D with respect to v_i. Together, the two equations state that the line joining $(x\ y\ z)^T$ to $\mathbf{F}^i(u_i, v_i)$ is perpendicular to both partial derivatives of $\mathbf{F}^i$ and therefore parallel to the normal. Therefore, the points on the envelope are a normal offset of F^i at a distance r and hence the tangent plane to the envelope is parallel to the tangent plane of $\mathbf{F}$. If we allow r to vary, we sweep out all possible envelopes of $\mathbf{F}$.

Therefore, the locus of points equidistant from neighborhoods of the footpoints $\mathbf{f}_1, \ldots, \mathbf{f}_n$ may also be described as the intersections of envelopes of these neighborhoods, with r varying. Under this formulation, we have a system of $3n$ polynomial equations in $2n + 4$ unknowns.

The offset sweep of simple boundary elements, such as planes, non-convex edges, non-convex vertices, natural quadrics, torii, and cyclides may be easily expressed by implicit equations in four variables $f^i(x, y, z, r) = 0$. In this case, the locus of points equidistant from neighborhoods of the footpoints is described by the n implicit equations of the governors in 4 unknowns. When these equations are of low degree, this is usually the most compact and easily solved representation of the locus. See Appendix A for the implicit equations of some simple offset sweeps.

4.4.1 Finding Junction Points

Suppose we have four or more governors and we suspect that one or more junction points results from these governors. If the offset sweeps are in implicit polynomial form, or if the envelope formulation (4.3) is a system of polynomials, then the equations may be solved by using the simultaneous equation solution techniques discussed in Chapter 2. If the governors are planes, then simple Gaussian elimination is sufficient to solve the system, as we shall see in Chapter 5.

Problems may arise when the system of equations does not result in isolated roots but rather in a continuous solution set, which may occur if the governors define a degenerate seam rather than one or more junction points. This case is easy to detect if the system

is linear (in which case the matrix of coefficients has a nontrivial nullspace) but less easy in the general polynomial case. One method of dealing with this problem involves running the Projected-Polyhedron solver discussed in Sherbrooke and Patrikalakis [77] at a coarse tolerance and examining the bounding boxes which remain after a few iterations. If they appear to be clustered together, then the solution set may consist of non-isolated roots. If we suspect the solution set to consist of non-isolated roots, there are other tests that may be performed to help confirm our suspicions. See Sherbrooke [76] for more details on this issue.

4.4.2 Tracing Seams

If we have a system of equations governing a seam, we can form a system of differential equations to trace out the seam. Assume for simplicity that we have a seam defined by the three equations $f^i(x,y,z,r) = 0$ for $i = 1,2,3$. This is an underconstrained system of three equations in four unknowns. Let us further suppose that we have a starting point and an initial tangent to the seam defined by this system of equations. As we shall see in the next three chapters, this information is acquired automatically in the course of running our algorithm.

Taking the differential of the system of equations gives

$$f^i_x dx + f^i_y dy + f^i_z dz + f^i_r dr = 0 \tag{4.5}$$

for $i = 1,2,3$. In matrix form, this may be expressed as

$$[F][dx\ dy\ dz\ dr]^T = [0\ 0\ 0]^T \tag{4.6}$$

where F is the 3×4 matrix of partial derivatives. In general, if F may be evaluated at a point (x,y,z,r), then there is a family of vectors $[dx\ dy\ dz\ dr]^T$ for which (4.6) will hold. These vectors will all be tangents to the seam at (x,y,z,r) since (4.6) is the differential of the system of equations describing the seam. Therefore, if we can find any one of these vectors and make it unit length, then we will obtain a unit tangent to the seam at each point, which is effectively an ordinary differential equation describing an arc-length parameterized seam curve. Since we have the initial tangent to the seam, we can ensure that the unit tangent has the correct direction and is not pointing backwards.

Finding this unit tangent is fortunately quite easy. We use the Singular Value Decomposition (SVD) on the matrix F, as described in Golub and Van Loan [33]. The SVD gives us, among other things, a basis for the nullspace of the matrix F, i.e. a basis for all vectors $[dx\ dy\ dz\ dr]^T$ which when premultiplied by F result in the zero vector. Since a seam is a curve, the basis will typically consist of only one vector. Simply scaling this vector by its magnitude will result in a unit tangent to the seam curve.

A similar approach holds for more complicated formulations of the seam curve (e.g. the envelope or offset sweep formulations discussed above). All that is needed is to take differentials of each governing equation and then use the SVD on the matrix of partial derivatives at each point in order to find the unit tangent.

Occasionally, the matrix F may be degenerate or near-degenerate (i.e. the nullspace may be two- or three-dimensional). Degeneracy may occur when governors are tangent to one another and therefore their offset sweeps have the same tangent spaces at the point of tangency. In this situation, we can obtain a unit tangent by using second order derivatives. We can either use adjacency information to determine if a tangency exists, or examine the singular values of F which are output by the SVD. If more than one singular value is close to zero (we use a tolerance of 10^{-4} to check this), then we have to use a second-order expansion. Alternatively, if the matrix is only near-degenerate but not actually degenerate, we can use a rational arithmetic implementation of the SVD to determine the tangent vector, assuming that our boundary elements are represented with rational numbers.

Assume for simplicity again that the seam is defined by three equations. The second order check proceeds by setting up the following system of equations, by Taylor's theorem:

$$0 = \frac{1}{\varepsilon}[f^i(\mathbf{x} + \varepsilon\mathbf{h}) - f^i(\mathbf{x})] = [\nabla\mathbf{f}^i(\mathbf{x})]\mathbf{h} + \frac{1}{2}\varepsilon\mathbf{h}^T[H^i]\mathbf{h} \tag{4.7}$$

for $i = 1, 2, 3$. Here $\nabla\mathbf{f}^i$ is the gradient of f^i, H^i is the Hessian of f^i, $\mathbf{x} = [x\ y\ z\ r]$ is the starting junction point, and $\mathbf{h} = [h_x\ h_y\ h_z\ h_r]$ is the unknown unit tangent vector to the seam. The fourth equation is $|\mathbf{h}| = 1$, the normalization condition. In the limit as $\varepsilon \to 0$, this system of equations should converge to one of the two unit tangents to the seam. In order to solve this system of four equations for the unknown four-element vector $\mathbf{h}$, we need to choose an ε sufficiently small so that the tangent is close but not so small that numerical error due to the degeneracy of $[F]$ is serious. Using double-precision floating

point arithmetic, we have observed that $\varepsilon = 10^{-4}$ is a good compromise.

Similar systems of equations may be set up for seams described by more complicated systems. Unfortunately, the system cannot be solved with simple Newton-Raphson because the Jacobian is singular; however, we have used a simple quasi-Newton method which does not use derivatives and have observed excellent results.

4.4.3 Tracing Sheets

The tracing of sheets is analogous to the tracing of seams except that the differential equation we obtain is no longer an ordinary differential equation but a partial differential equation. As before, the differentials of the governing equations are taken, and then the SVD may be applied to the matrix of partial derivatives at any given point in order to determine the nullspace. The nullspace of the matrix should generally be two-dimensional, since a sheet is a surface, so the SVD will provide two basis vectors.

The actual tracing of the sheets is somewhat difficult due to the fact that each sheet may be bounded by a complicated network of seams and arms. Also, partial differential equations are usually much more difficult to deal with than ordinary differential equations. For the purpose of visualization, we recommend finding polyline approximations to the bounding curves of the sheet (the governing equations of these bounding curves emerge automatically during the course of our algorithm). The polyline approximation of a seam curve may be obtained by applying the tracing technique described above. For each point in each bounding polyline, apply the SVD to the matrix of partial derivatives to find the tangent plane to the sheet at that point. This will give a normal for each point in the bounding polyline.

Once these normals are obtained, pick a starting point on the boundary of the sheet (for example, a junction point) and march along the tangent plane a short distance to four new points ninety degrees apart. If any of the points lie outside the network of bounding curves, discard them, otherwise proceed recursively, storing the normal at each step. This simple technique should be sufficient for visualization purposes. For more stringent accuracy requirements, a PDE solver should give better performance.

Chapter 5

An Algorithm for Determining the MAT of Convex 3-D Polyhedral Solids

5.1 Introduction

In this chapter, we introduce a connectivity-based algorithm to determine the Medial Axis of a convex polyhedral solid, and analyze its complexity and stability. This technique builds naturally on the classification scheme discussed in the previous chapter, and exhibits attractive properties of efficiency and stability. Most of the algorithm is recursive in nature, so we will consider the recursive portion now. Later on in this section, we will discuss starting and terminating the algorithm.

The recursive part of the algorithm starts at a junction point and locates all adjacent seams to the junction point. It then finds the endpoints of all of these seams, whether they are seam-endpoints or new junction points. Then, after finding the convex edges adjacent to each seam-endpoint, it continues the recursion at each of the new junction points. During the recursion a graph of vertices and edges is built up, where each vertex is either a junction point or a seam-endpoint and each edge is a seam or a convex edge. Once the recursion is done, each closed loop in the graph is traversed to elicit the sheet bounded by the loop.

5.2 The Recursion

Suppose we have found a junction point $\mathbf{p}$ in the interior of our polyhedral solid P, and suppose this junction point has governor set $P_g = \{P_1, \ldots, P_n\}$, where $n \geq 4$.

1. Using theorem 4.3, find all seams which emanate from $\mathbf{p}$. In order to generate all possible seams, first generate all subsets of three elements from the set P_g. Then for each subset,

   (a) Let P_a, P_b, and P_c be the three faces and let $\mathbf{d}_a, \mathbf{d}_b$, and $\mathbf{d}_c$ be unit vectors which point in the direction of the shortest paths from $\mathbf{p}$ to each of these three faces. We need to find an equiangular vector (a vector which forms the same angle with all three of these unit vectors). To do this, note that the vectors $\mathbf{d}_a + \mathbf{d}_b$ and $\mathbf{d}_a \times \mathbf{d}_b$ are both equiangular to $\mathbf{d}_a$ and $\mathbf{d}_b$, so their cross product $\mathbf{n}_1$ represents a normal to the plane which bisects $\mathbf{d}_a$ and $\mathbf{d}_b$. Construct the normal $\mathbf{n}_2$ to the plane bisecting $\mathbf{d}_b$ and $\mathbf{d}_c$ in the same way. Then the vector $\mathbf{n}_1 \times \mathbf{n}_2$ is mutually perpendicular to $\mathbf{n}_1$ and $\mathbf{n}_2$ and therefore lies in both bisecting planes. Since it lies in both bisecting planes, the angle it makes with each unit vector is the same and therefore it is an equiangular vector.

   (b) The preceding step gives us two equiangular unit vectors of $\mathbf{d}_a, \mathbf{d}_b$ and $\mathbf{d}_c$: the unit vector of $\mathbf{n}_1 \times \mathbf{n}_2$ and the unit vector which points in the opposite direction. Our next step is to discover whether either vector represents a tangent to an actual seam or not. Theorem 4.3 tells us that we can make this decision taking the dot products of both vectors with the unit distance vectors to all of the other faces in P_g. If for one of the vectors the dot product is always less than the dot product with $\mathbf{d}_a$ (or $\mathbf{d}_b$ or $\mathbf{d}_c$ since the angles are the same) then a seam exists with governors P_a, P_b and P_c. Otherwise, no such seam exists, and we return to step 1 to consider a new subset of three elements.

2. At this point, we know a seam exists with governors P_a, P_b and P_c adjacent to $\mathbf{p}$. Check to see if a seam with these governors already exists by examining a global lookup table. If it does, add the seam as an adjacent edge to $\mathbf{p}$ and return to step 1 to consider a new subset of three elements.

3. On the other hand, if the seam was not found, create it and add it as an adjacent edge to **p**. Also add it to the global lookup table for future reference.

4. The point **p** is one endpoint of this seam. Now we must find the other. There are only two possibilities: either the seam ends in another junction point, or it hits the boundary (and thus ends in a seam-endpoint). First we check to see if the seam ends in another junction point. This junction point must have P_a, P_b, and P_c as three of its governors since it is adjacent to the seam, and it must have at least one other governor which is not a member of P_g. (Otherwise, we would just end up with **p** again rather than a new junction point.) So, for each face in P not in P_g,

   (a) Form the offset sweeps of P_a, P_b, P_c and the new face. This leaves us with four equations in four variables (x, y, z and r) (see appendix A). Furthermore, definition 4.1 shows that these equations are linear since the normal to a plane is constant.

   (b) Solve these four equations in four unknowns for x, y, z and r. If the system does not have a unique solution, discard it and go on to the next face. If it does, record it as a candidate junction point.

   (c) The candidate junction point must be distance minimal with respect to other faces on the boundary. So, find the distance from the candidate point to all the boundary faces. If the distance to any other face is less, discard the point and go on to the next face. If the distance is equal, add the associated face to the set of governors.

5. If we have found a junction point, add it into the data structure if it is new. Make a recursive call back to step 1 with this new junction point. If the point is not new, connect it to the current seam and return without taking another recursive step.

6. If no junction point was found, the only possibility is that the seam ends in a seam-endpoint. Therefore, faces P_a, P_b, and P_c intersect in a convex vertex which is the same as the seam-endpoint.

7. Go to step 1 for each of the other three-element subsets of P_g.

5.3 Starting and Ending the Recursion

The preceding section demonstrates how the method proceeds once a junction point has been found. It turns out that starting and ending the algorithm is considerably easier. To start the process, we merely need to locate a convex vertex on the boundary. Since this vertex lies at the meeting of three or more faces, simply select three of these faces. These three faces must govern a seam (with a seam-endpoint at the convex vertex); we have seen in the previous section how a bisecting vector of three faces tangent to a seam may be found, and this seam obviously satisfies distance minimality in at least a small neighborhood of their intersection since the radius function is near zero. Once we have found a seam, we proceed as we did in the recursion to find a junction point at the other end. Once we have the junction point, we begin the recursion.

The algorithm will eventually stop when all seams have been traversed. At this point we have a graph connecting junction points, seams, and seam-endpoints, which must be completed by adding sheets and convex edges.

In order to add convex edges, we merely need to connect all the seam-endpoints together. This is a trivial matter; simply consider each edge of the boundary and connect two seam-endpoints together if the edge has them as its endpoints.

Finding sheets is slightly more difficult, but proceeds similarly. The first step is to merge seams together which have the same endpoints but different governors. This situation occurs when a degenerate seam (a seam with more than three governors) appears in the MA and is mistakenly flagged by our algorithm as more than one seam. However, it is easily dealt with by discarding all seams except one and making its set of governors the union of its governors and the governors of the discarded seams. Once the merging is performed, we select a seam and create the sheets which are adjacent to it (all possible subsets of two elements). Then for each of these sheets, we start at the seam and traverse the graph, following adjacent junction points, seams, convex edges, and seam-endpoints only if the two elements governing the sheet are in the governor sets of the vertices and edges in the graph. Every junction point or seam-endpoint we reach is added as a vertex of the sheet polygon, and each seam or convex edge is added as an edge of the sheet.

Clearly, if we move through the list of seams and start the sheet traversal for all pairs of elements which are subsets of the seam's governors and have not already been considered,

we will find all sheets in the MA, and thus find the complete MA of P.

5.4 Radius Function Computation

The algorithm discussed above will find the complete MA of a convex polyhedral solid. However, what if we want to determine the radius function as well, as part of the full MAT? We would want to do this, for example, if we were concerned about being able to invert the MAT and find the original polyhedron, or in order to assist the direct application of the MAT in the problems identified in the introduction. As it happens, it is easy to include radius function information as the algorithm proceeds. All that is needed is to record the radius function at each junction point. The offset sweeps of the faces of P are linear, so not only are pieces of the MA linear, but also the radius function itself varies linearly along each branch of the MA. Therefore, a record of the radius at each junction point is sufficient to determine the radius function at any point on the MA. For example, consider a point on a seam. This point may be expressed as a convex combination of the two end points of the seam (which are either junction points or seam-endpoints) and is therefore of the form $t\mathbf{x}_0 + (1-t)\mathbf{x}_1$ where $\mathbf{x}_0$ and $\mathbf{x}_1$ are the endpoints of the seam and t is a parameter characterizing the point. Then the radius function at the point is given simply by $tr(\mathbf{x}_0) + (1-t)r(\mathbf{x}_1)$. Similarly, if the point lies on a sheet, it may be expressed as a convex combination of the vertices of the sheet; the same convex combination of the radius values at these vertices will give the radius function at the point in question. Note that although there may be more than one convex combination of vertices of the sheet for the point, the result of the radius function will be the same for any valid convex combination.

5.5 Data Structure

The data structure used to represent the MAT of a convex polyhedral solid must be non-manifold since areas of the MA may well be non-manifold. Fortunately, a simple adjacency graph may be built as the algorithm proceeds. Unlike more complex non-manifold data structures (see Bardis and Patrikalakis [3] for a review) this data structure does not carry ordering information. Each face (sheet) contains a list of adjacent edges (seams or convex edges) and a list of vertices (junction points or seam-endpoints). Similarly, each edge has a list of adjacent faces and vertices, and each vertex has a list of adjacent edges and faces.

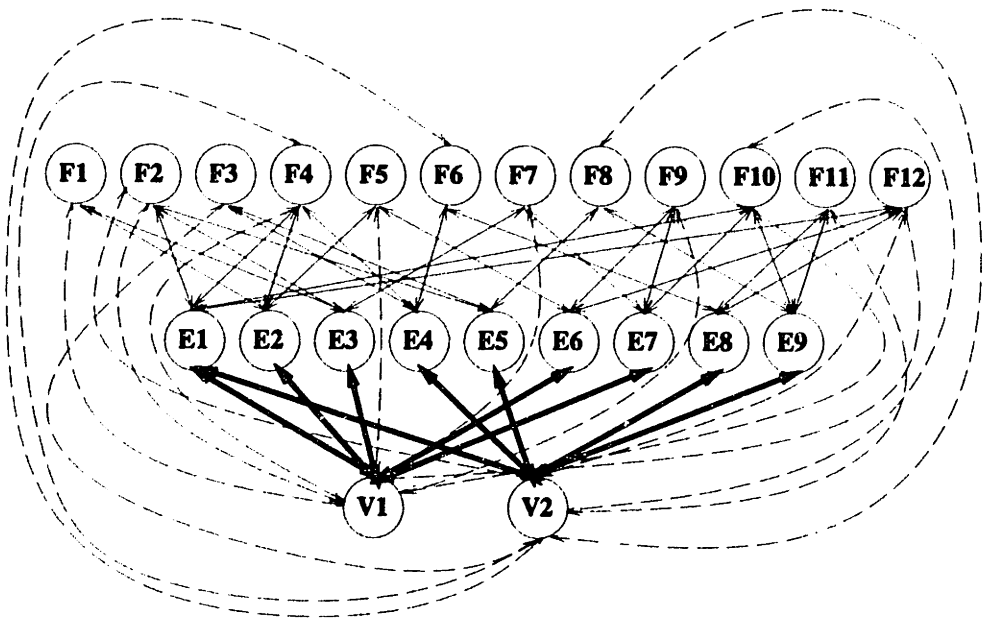

Figure 5-1: The adjacency graph of the interior portion of the MAT of a $2 \times 2 \times 1$ rectangular box. Convex edges and vertices are left out for simplicity.

The adjacency graph is built up naturally during the algorithm as we proceed from piece to adjacent piece of the MA. Figure 5-1 shows the adjacency graph of the interior portion of the MA (not including convex edges and vertices) for a $2 \times 1 \times 1$ rectangular box. Section 5.7 contains figures of this box and its MA.

5.6 Algorithm Analysis

5.6.1 Complexity

It is difficult to analyze the complexity of this algorithm in terms of n, the number of faces of P, since the number of seams and sheets may vary from polyhedron to polyhedron. However, we can determine a reasonable complexity bound based on the number of seams, n_S, in the MA as well as n.

An examination of the algorithm in section 5.2 will show that the major cost of the algorithm lies in finding a new junction point to end a seam, since this phase must be performed for every feasible seam. The number of feasible seams is typically on the same order as the number of subsets of three elements, unless the junction point in question is extremely degenerate, so we are not concerned about the cost of the feasibility check except to observe that it is an extremely fast portion of the algorithm. So it only adds a

small amount of work to the algorithm, which is overshadowed by the determination of new junction points as n gets large.

In a naive approach to finding a new junction point, we would consider every face in P not in P_g, the set of governors of the old junction point. This number is $\mathcal{O}(n)$. Then for every face which gives a candidate, we have to do $\mathcal{O}(n)$ distance checks to make sure the point is distance minimal. So in theory it is possible that $\mathcal{O}(n^2)$ work is required to find a new junction point. However, it is typically true that the fourth face that we seek is adjacent to one of the three faces governing the seam, so by utilizing adjacency information in our input graph we can expect to reduce the cost to $\mathcal{O}(n)$ since the number of faces adjacent to three faces is usually a small factor independent of n. And of course, distance checks to polygons are themselves virtually instantaneous. Some time is spent maintaining global lookup tables for seams and junction points to ensure that work is not repeated; however the amount of time to perform a lookup in the table is logarithmic in the size of the table [22], and therefore does not affect the overall complexity of the algorithm.

We conclude that the cost of the recursive portion of the algorithm is $\mathcal{O}(n_S n)$ in the average case. The addition of convex edges to the graph is linear in the number of edges of the polyhedron, which is $\mathcal{O}(n)$. This result is shown in Brisson [16] and the proof proceeds as follows: Let v, e, and f be the number of vertices, edges, and faces in P. Triangulate all the faces, so that v', e', and f' are similarly defined for the triangulation P'. By Euler's formula, $v' - e' + f' = 2$, and since P' has only triangular faces, $2e' = 3f'$ (since each edge has two common faces, and each face has three edges). Thus $e' = 3v' - 6$. Since $v' = v$ and $e \leq e'$, the number of edges in P is no greater than $3v - 6$ and therefore the number of edges is $O(n)$. The addition of sheets is linear in the number of seams, since each seam is traversed once for each subset of two elements of the seam's set of governors. So the total algorithm can be expected to be $\mathcal{O}(n_S n)$ in the average case.

Storage in this algorithm is proportional to the final MA graph. Additionally, no internal computation requires much storage. In fact, the most difficult computation in the algorithm is the solution of four linear equations in four unknowns.

5.6.2 Stability

A more serious question is the question of stability. The Medial Axis is extremely sensitive to certain types of perturbation (for instance, the addition of a small kink in the boundary).

However, if we assume that the number of faces in the polyhedron is fixed and then deform the polyhedron slightly, maintaining convexity, we observe some pleasing properties of the MA. Because the equidistantial point sets which make up the MA (junction points, seams, sheets) really represent an averaging process of boundary faces, they tend to be stable under perturbation. The MA becomes much less stable in degenerate or near-degenerate situations. For example, a degenerate junction point may become two or more nondegenerate junction points which lie extremely close to each other under some small perturbations of the boundary. Obviously, in this situation a small difference in the boundary may result in drastically different MA graphs.

We have two options for dealing with this situation. The first option is to put a tolerance on all distance checks. Under this option, junction points which are close to one another will be merged into a degenerate junction point. Although the MA is no longer "correct" if we utilize tolerances, it may actually be more useful. For instance, if we are using the MA for finite element meshing of the region, it is in our best interest *not* to have small branches as they will interfere with the meshing process [80].

The second option is to use exact, arbitrary precision rational arithmetic or rounded interval arithmetic to help make decisions for us. Rational arithmetic is an extremely appealing prospect since the numerical computation involved in the algorithm is minimal. Typically when using rational arithmetic, the growth of numerators and denominators in the representation of rational numbers is a concern. However, in this algorithm, the only possible place where appreciable growth can occur is in the junction point computation involving the solution of four linear equations in four unknowns. Since this computation depends only on the equations of the boundary faces, we can expect the sizes of the rationals to remain well under control.

5.7 Examples

Let us now examine a few examples of the algorithm in action. All of these examples were run on a Silicon Graphics Onyx workstation running at 150 MHz, and the code was written in C++ and linked with the IRIS Inventor toolkit.

Figure 5-2 shows our first example, the MA of a $3 \times 2 \times 1$ rectangular box. Notice that all of the junction points and seams are nondegenerate. In figure 5-3, we have induced a

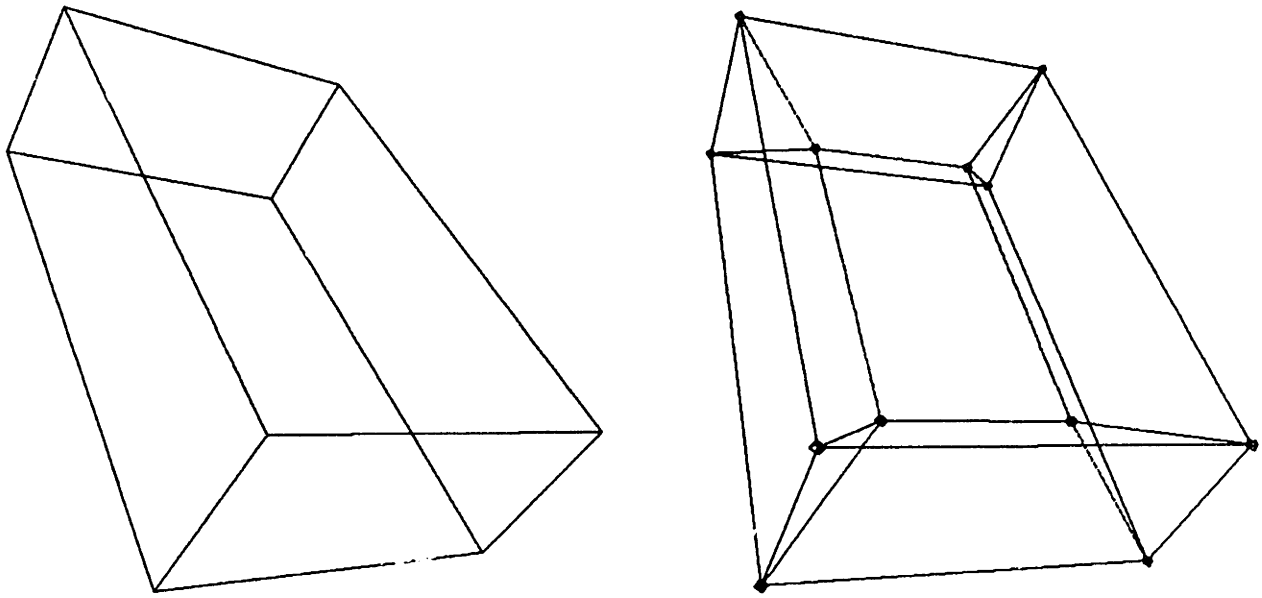

Figure 5-2: A $3 \times 2 \times 1$ rectangular box and its MA.

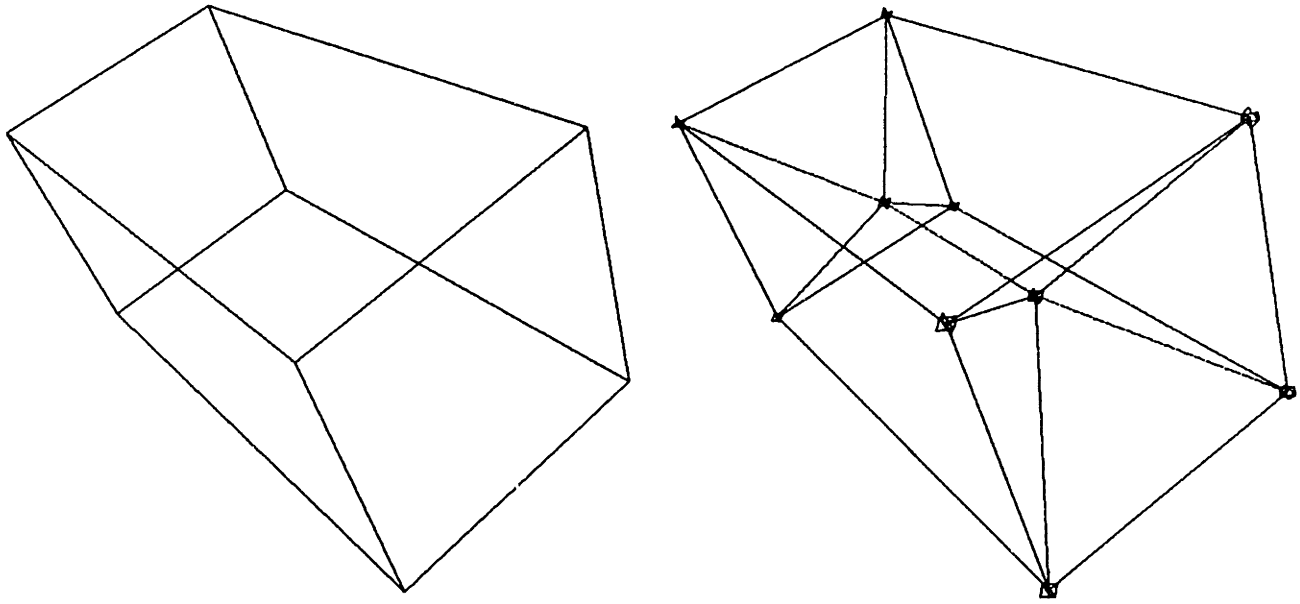

Figure 5-3: A $2 \times 1 \times 1$ rectangular box and its MA. Notice the two degenerate junction points and the degenerate seam joining them

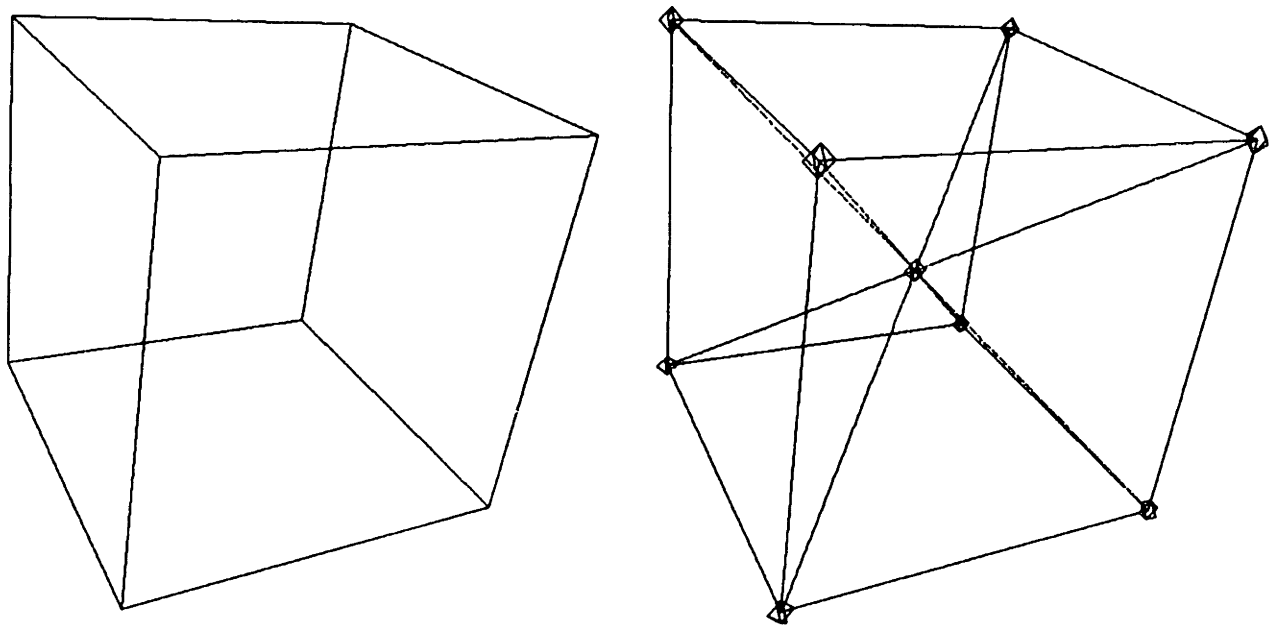

Figure 5-4: A cube and its MA.

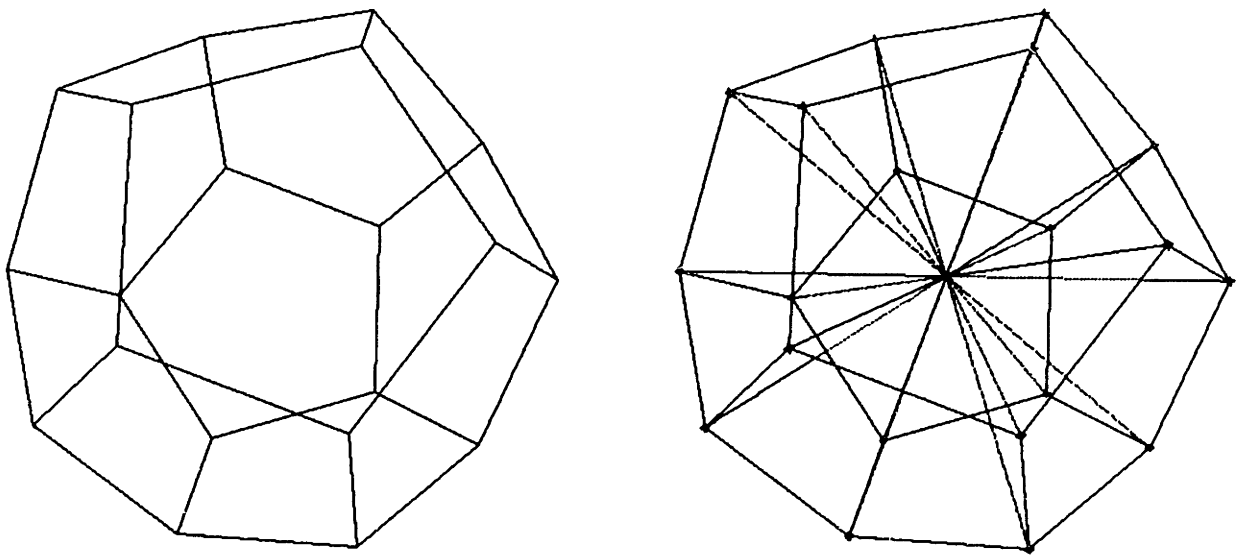

Figure 5-5: A regular dodecahedron and its MA.

degeneracy by reducing the length of the longest side to 1 so that the box is now $2 \times 1 \times 1$. The four junction points of the previous example have now merged into two degenerate junction points, each with five governors. The two junction points are connected by one degenerate seam with four governors. Going one step further, figure 5-4 induces a higher-order degeneracy by making all three sides equal. The degenerate seam has vanished as the two degenerate junction points become one degenerate junction point with six governors. Figure 5-5 shows a regular dodecahedron and its MA, which has one junction point. This junction point is extremely degenerate, with twelve governors.

Chapter 6

Extending the Algorithm to General 3-D Polyhedral Solids

6.1 Introduction

In the previous chapter, we introduced an algorithm for determining the Medial Axis Transform of a convex polyhedral solid. In this section, we examine the modifications needed to make the algorithm handle general polyhedral solids, with holes (arbitrary genus) or non-convex edges and vertices. Clearly, the major problem with extending the algorithm is that non-convex edges and non-convex vertices (which have to be added to the set of faces to generate the complete set of boundary elements) may give rise to nonplanar sheets and nonlinear seams. Such entities therefore have to be computed explicitly and represented or traced with a differential equation method.

However, there are other, less obvious problems that arise in the extension of the algorithm. The first major issue is the inability to apply the classification theorem 4.3 as stated in chapter 4, since the governors need no longer be planes. Figure 6-1 shows the problem with applying the classification theorem naively. The figure shows a junction point with three planar governors and one non-convex edge governor. The vector $\mathbf{v}$ which points straight down is equiangular with respect to all four unit vectors pointing to the footpoints of $\mathbf{p}$. Therefore, by the classification theorem for convex polyhedra, we might expect other Medial Axis points in a neighborhood of $\mathbf{p}$ with footpoints in neighborhoods of all four of $\mathbf{p}$'s footpoints. However, no such points exist, because of the fact that the offset of the

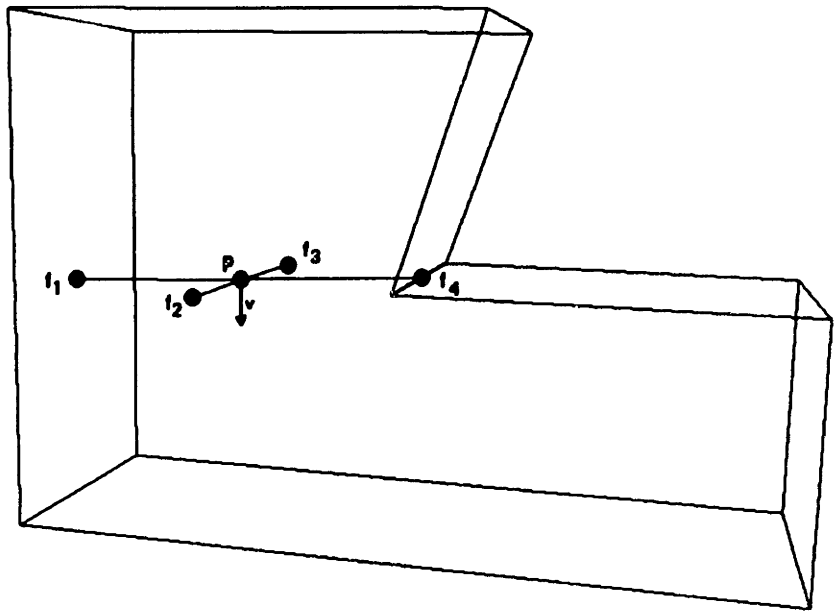

Figure 6-1: An example of the failure of the classification theorem 4.3. Because **v** is equiangular to all labeled footpoints, we might expect other MA points in the direction of **v** to be equidistant from points near all labeled footpoints. However, this is not the case, due to the non-linearity of the offset sweep of the non-convex edge. Fortunately, there are other approaches to answer classification ambiguities of this nature.

non-convex edge is curved instead of planar. Thus, in the direction of **v**, the distances to the three planar faces remain constant, but the distance to the edge increases. Therefore, if we are attempting to determine the local structure of the Medial Axis around **p**, the dot product test is insufficient, and therefore other methods must be utilized, as we show later on in this chapter.

Another major issue is one revolving around non-uniqueness of governor sets. For example, in the case of a convex polyhedral solid, it is impossible to have two different seam curves with the same three governors. It is easy to show this simply by intersecting the offset sweeps of the planes; since they are offset sweeps of convex planar faces, the offset sweeps themselves are convex and therefore their intersection set is connected. However, in the general polyhedral case, the offset sweeps no longer need be convex, and therefore the intersection sets may be disconnected. Hence, it is possible to have two different seam curves with the same three governors, or even two different sheets with the same two governors.

In addition, in the case of convex polyhedral solids, we did not worry about our ability to reach all junction points and seam-endpoints by starting at a single seam-endpoint on the boundary and traversing seams and rims. Unfortunately, there are certain classes of poly-

hedral solids for which this property does not hold; for example, the presence of polyhedral cavities (empty volumes within the polyhedral solid whose boundaries are not connected to the outer boundary of the solid) may cause such difficulty. We show in section 6.5 that when we require all faces of the polyhedron to be *simply connected*, and exclude polyhedral cavities, the problem does not arise. Since any multiply connected face may be subdivided into a number of simply connected faces, our algorithm will be able to handle polyhedral solids of arbitrary genus as long as polyhedral cavities are not present.

The final major difference is that the governing equations of junction points need no longer be linear, since the offset sweeps of non-convex edges and corners have quadratic terms. The obvious result of this problem is that a linear system solver is inapplicable to the problem, and we must use a solver tailored to nonlinear polynomial systems if we want to use the same approach. Unfortunately, there are a few inherent problems with blindly applying a nonlinear polynomial solver. The most serious is one due to tangency; the offset of a non-convex edge, for example, is a cylinder tangent to both of its adjacent faces. Therefore, a system of equations containing the offset sweeps of the edge and either of its adjacent faces will take a long time to solve due to the tangency.

These problems range in scope from annoyances to major hindrances. As a result, some sizeable modifications are needed in various parts of the algorithm, as we shall see below. The resulting algorithm is somewhat simpler conceptually than the algorithm for convex polyhedra, although there is some hidden implementation complexity which is mentioned later. The same basic idea of moving from junction point to junction point along seams, recursively, is kept, although the method of loop traversal used to determine sheets is somewhat more complicated. To aid the reader in implementing the algorithm, we give our examples in pseudocode, with explanations following each routine. As in the previous chapter, we begin with the recursion around a junction point.

The algorithm in this chapter is a refinement and expansion of Sherbrooke, Patrikalakis, and Brisson [78]. We note that it can be easily enhanced to compute the Voronoi Diagram rather than the Medial Axis, since it is based on the traversal of equidistantial sets between boundary elements. As we shall see in section 6.2.1, in determining the Medial Axis, certain candidate seams which are part of the Voronoi Diagram are marked infeasible because they do not lie on the Medial Axis. By removing this restriction and allowing sheets to exist which are equidistant from two adjacent tangent boundary elements (like a face and an

adjacent non-convex edge), the algorithm can be made to determine the Voronoi Diagram instead of the Medial Axis. The Voronoi Diagram itself decomposes a shape into simply connected regions, which is useful in finite element meshing.

6.2 Traversing from a Junction Point

Once at a junction point, the method of traversal is essentially the same as before. We begin by taking all three-element subsets of the governor set of the junction point in order to trace out all adjacent seams. In this case the possible governors consist of all faces, all non-convex edges, and all non-convex vertices. Each seam is checked for feasibility and degeneracy, as described below. Once checked, each feasible seam is itself traversed, and the point ending the seam is returned to us. Based on whether the endpoint is another junction point or a seam-endpoint, we continue the recursion. A lookup table of junction points is maintained during the procedure; newly discovered junction points are checked against this table using a small tolerance, and added if not found in the table.

One difference from the traversal mentioned in the previous chapter is that we make two copies of each possible three-element seam governor set, to signify that there are two possible tangents for each subset. There are rare cases where both tangent directions may generate valid seams, due to degeneracy. The tangents themselves are computed as in the previous chapter, by finding an equiangular vector between the three distance vectors. If two or more of the distance vectors are collinear, then the second-order check described in section 4.4.2 may be used to determine the tangent directions.

As we proceed through the recursion, we build up an adjacency graph containing the geometry of the Medial Axis. We connect each feasible seam to the starting junction point, and then for each seam we connect its ending point. Once we are done with the junction point, we mark it done so that we do not repeat any work. At the end of the recursion, we will have a network of vertices and edges forming the boundary of the Medial Axis.

```
TRAVERSE-JUNCTIONPOINT(j)
 1  S ← SUBSETS(Governor-Set[j], 3)
 2  V ← EMPTY-SET
 3  COMPUTE-TANGENTS(S)
 4  for i ← 1 to Number-of-Elements[S]
 5       do if FEASIBLE-SEAM(j, S[i]) = FALSE
 6            then DELETE-ELEMENT(S, S[i])
 7  CONSOLIDATE-SET(S)
 8  for i ← 1 to Number-of-Elements[S]
 9       do s_i ← SEAMCURVE(j, S[i])
10          v_i ← TRAVERSE-SEAMCURVE(s_i)
11          if LOOKUP-JUNCTION(v_i) = TRUE
12            then v_i ← CLOSEST-JUNCTION(v_i)
13            else ADD-TO-JUNCTION-LOOKUP-TABLE(v_i)
14          if Done[v_i] = FALSE
15            then CONNECT(j, s_i)
16                 CONNECT(s_i, v_i)
17                 ADD-ELEMENT(V, v_i)
18  S' ← SUBSETS(Governor-Set[j], 3)
19  Repeat steps 3–17 for S', but with the tangents pointing in the opposite direction
20  Done[j] = TRUE
21  for i ← 1 to Number-of-Elements[V]
22       do if V[i] is a junction point
23            then TRAVERSE-JUNCTIONPOINT(V[i])
24            else TRAVERSE-SEAMENDPOINT(V[i])
```

$1 \rightarrow 7$. Get all three-element subsets of j's governors and store them in S. Initialize V, the set of vertices connected to j through seam curves, to the empty set. Compute tangents to each possible seam curve in S. Go through all the subsets and delete the invalid ones. The valid ones are the ones which govern feasible seams in the neighborhood of j. During the course of the feasibility check, perform a degeneracy check as well to see if other governors of j also govern the seam (see below). If so, add these governors to the subset

to signify that the seam is degenerate. If the seam is infeasible, delete the set of governors from S. Finally, call CONSOLIDATE-SET on S. This routine makes sure that there is no duplication of sets in S. For example, consider a degenerate seam with governors $1, 2, 3, 4$. Originally two of the three-element subsets of governors would have elements $1, 2, 3$ and $1, 2, 4$. However, after the degeneracy check, element 4 would be added to the first subset and element 3 to the second, since there is degenerate seam with governors $1, 2, 3, 4$. The duplicate set must be removed in order to avoid duplicating work.

$8 \rightarrow 17$. Loop over those subsets which generate valid seams. For each subset, create the seam curve which starts at j and has the governors listed in the subset. Call TRAVERSE – SEAMCURVE in order to find the point v_i which ends the seam curve. Check first to see if v_i is sufficiently close to an already determined junction point; if so, replace v_i by that closest point. Otherwise, add the new junction point to the lookup table. If v_i has already been completely visited, then ignore it. Otherwise, connect the seam with its starting vertex j and its ending vertex v_i, and add the vertex to V, the list of vertices to be traversed on the next step.

$18 \rightarrow 19$. Do steps 3-17 again, but this time for an identical list of subsets with tangents pointing in the opposite directions as before.

$20 \rightarrow 24$. Set the *Done* flag on j to signify that it has been completely traversed. Now traverse all the vertices which end seams emanating from j, calling TRAVERSE – JUNCTIONPOINT or TRAVERSE – SEAMENDPOINT depending on the type of vertex.

6.2.1 The Feasibility Check

As stated before, the original dot product test is insufficient to determine feasibility and degeneracy for a seam. The problem is due to the fact that the dot product test is essentially a first-order test and therefore a non-convex edge and an adjacent planar face are treated the same way. One possible approach is to use higher order derivatives to make definite decisions. However, a simpler method is to simply trace a short distance along the seam and see if the seam has become infeasible. If it is still feasible, check the distances to the other governors in the junction point to determine degeneracy.

In the case of non-convex polyhedra, some additional checking for infeasibility is needed. Because we consider an edge and an adjacent vertex to be separate boundary elements, even

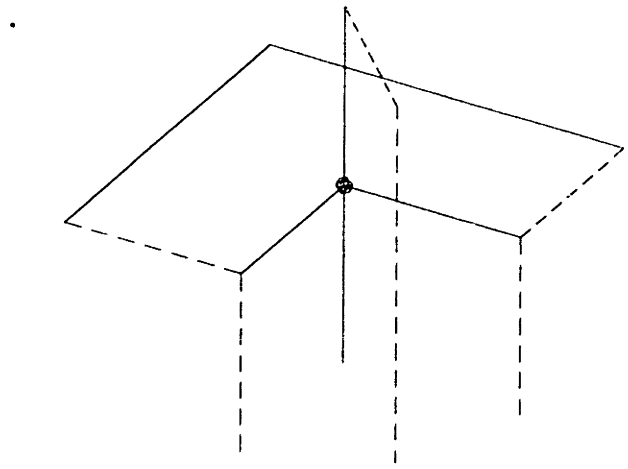

Figure 6-2: A rim curve on the interior of a polyhedral solid runs vertically up from a mixed vertex. The inside of the polyhedron is above the cut horizontal plane; the rim curve is induced by the sheet running through the middle of the cut.

though geometrically the vertex is contained in the edge, there will be a few seams which, although equidistantial to more than one boundary element, do not belong to the Medial Axis. Therefore, during the feasibility check, we examine the set of footpoints of the seam at the starting point. If the footpoints of the seam are all the same point (but on different boundary elements) then we usually should throw the seam away.

The only exception to this rule is if throwing away a seam will remove one of the bounding curves of a sheet. On occasion, there may be rim curves in the *interior* of the solid (see figure 6-2). Consider a horizontal plane which forms part of the boundary of P, so that if we are above the plane we are inside P, and cut 90 degrees out of the plane at the origin. In the cut, insert two planes perpendicular to the first plane, connected along the two non-convex edges resulting from the cut. Then there is a sheet which extends both below and above the horizontal plane; below the plane it is equidistant from the two planes, and above it is equidistant from the two non-convex edges. The sheet ends in a vertical ray extending upward from the origin; since any point on this ray has a maximal ball tangent only at the origin, the ray represents a rim rather than a seam.

In our algorithm, this ray can be detected as a seam since it is equidistant from the two non-convex edges and the non-convex vertex. Although it should be reclassified as a rim curve instead of discarded (since removing it will create an unbounded sheet), we can still

trace it using our seam traversal routine. We can detect such cases by checking that a sheet is connected to the rim; this will be true if some two-element subset of the rim's governors actually generates a sheet (i.e. the two elements are not tangent to one another).

How far can we trace along the seam in order to determine degeneracy? Fortunately, this is a simple question to answer. At the junction point where we start, compute the distances to every other element in the solid. Suppose the distance from the junction point to its governors is r, and the shortest distance to any other boundary element is $r' > r$. Then we can trace any distance up to $\frac{r'-r}{2}$ before any other boundary element has a chance of becoming active (this worst-case bound assumes the seam moves away from its governors as quickly as possible and approaches another boundary element as quickly as possible). In the pseudocode below, this distance is called SAFE-DIST. In a floating-point implementation, we will want SAFE-DIST to be safely smaller than $\frac{r'-r}{2}$ in order to avoid possible tolerancing difficulties.

```
FEASIBLE-SEAM(j, G)
 1  if EQUAL-FOOTPOINTS(j, G)
 2     then if not ADJACENT-SHEET(j, G)
 3             then return FALSE
 4  N ← Governor-Set[j] − G
 5  s ← TRACE-SEAM-LENGTH(j, G, SAFE-DIST)
 6  r ← Radius-Function[s]
 7  for i ← 1 to 3
 8       do if FOOTPOINT-IS-OFF(s, G[i])
 9             then return FALSE
10  for i ← 1 to Number-of-Elements[N]
11       do if DISTANCE(s, N[i]) < r
12             then return FALSE
13             else if DISTANCE(s, N[i]) = r
14             then ADD-ELEMENT(G, N[i])
15  return TRUE
```

$1 \rightarrow 3$. Check first to see if all the footpoints of the seam are equal at the starting point. If so, either the seam is infeasible, or it is an internal rim curve. Call ADJACENT-SHEET to

see if there exists a sheet adjacent to the seam at the starting point such that the sheet has a subset of the governors of the seam. ADJACENT-SHEET checks all three subsets of two elements to see if a sheet exists. The sheet must not be governed by two adjacent, tangent boundary elements (like a face and an adjacent non-convex edge), and of course it must be distance minimal. Our routine finds a vector perpendicular to the candidate seam and tangent to the seam and checks that one of the two unit directions to the sheet maintains distance minimality (as described later in section 6.7). If neither direction works, it returns FALSE.

$4 \rightarrow 6$. Initialize N to be the set of governors of the starting junction point which are not also governors of the candidate seam. Trace along the seam a small distance equal to SAFE-DIST. Assign the ending point of the trace to s. Let r be the radius function at that point as computed during the trace.

$7 \rightarrow 9$. During the trace, the footpoints of our seam points may have stepped off of the governors and onto the analytic extension of the governors instead. This could happen, for instance, if a degenerate junction point has a polygon and a non-convex edge of the polygon as two of its governors, so that the footpoint on the edge and the footpoint on the polygon coincide. Clearly, tracing in one direction should make the polygon inactive and the edge active. However, the differential equation trace works by using the mathematical equation of the plane containing the polygon (the analytic extension of the polygon) and therefore after tracing we must check that the footpoints are still on the governors and not simply on one of the analytic extensions.

$10 \rightarrow 14$. Since the three initial governors of the seam have passed their initial test, check the other governors of the starting junction point. If any of the governors are closer than the three seam governors, the seam is infeasible. If a governor is at the same distance as the three seam governors, it gets added to the governor set of the seam, signifying that the seam is degenerate.

15. If all tests are passed, TRUE is returned to signify feasibility.

6.3 Traversing from a Seam-Endpoint

Traversing from a seam-endpoint is the simplest part of the algorithm. The actual recursive aspect of the algorithm is handled at the junction points; the only thing needed at the

seam-endpoints is to find all the convex edges adjacent to the seam-endpoints and connect them in the data structure. The list of convex edges in the polyhedron is available to us from the beginning; it is simply a matter of searching through the list to find those edges which have the seam-endpoint as one of their vertices.

TRAVERSE-SEAMENDPOINT(s)

1 $C \leftarrow$ SUBSETS(*Governor-Set*[s],2)

2 **for** $i \leftarrow 1$ **to** *Number-of-Elements*[C]

3 **do if** there is a convex edge e_i starting at s with governor set $C[i]$

4 **then** CONNECT(s, e_i)

6.4 Traversing a Seam

Once we have a starting point, the set of governors, and an initial tangent, it is a simple matter to traverse the seam until the next junction or seam-endpoint. We simply integrate the system of differential equations describing the seam (see section 4.4.2) using, for example, an adaptive-step fourth-order Runge-Kutta method, and check at each step to see if another boundary element is about to become active. Since during the tracing process we update the value of the radius function, we merely have to check that the radius value is smaller than the distance to any boundary element which is not a governor of the seam. If the distance to another boundary element drops below the radius function, we stop tracing and signal that we have found a junction point. Alternatively, if the distance drops below a small positive tolerance, we conclude that the seam ends in a seam-endpoint on the boundary.

The initial approximations that we get by stopping the tracing may not be very accurate. Therefore, some refinement of our approximations may be necessary. In the pseudocode below there are two routines called REFINED-SEAMENDPOINT and REFINED-JUNCTIONPOINT. The first of these routines takes an approximation to a seam-endpoint and returns the actual seam-endpoint by looking through the list of convex vertices of the polyhedron and selecting the closest one. REFINED-JUNCTIONPOINT performs a similar task, taking an approximation to a junction point and refining it to a much closer approximation. There are several ways to perform this refinement. One possibility is to use minimization. For example, suppose there are four equations $f_1 = f_2 = f_3 = f_4 = 0$, each

one defining the offset sweeps of a governor of the junction point. Using our rough approximation as an initial guess, minimize the squared error $f_1^2 + f_2^2 + f_3^2 + f_4^2$; the error has a local minimum of zero at the junction point. Alternatively, we can use an equation solver like the Projected-Polyhedron method discussed in Chapter 2 to find a root of the system of equations within a small box around our approximate root. Once REFINED-JUNCTIONPOINT has refined the junction point, it returns the refinement, along with any extra governors the point may have due to degeneracy. The extra governors may be easily determined by checking distances to boundary elements at the refined point.

```
TRAVERSE-SEAM(s)
 1  G ← BOUNDARY-GEOMETRY − Governor-Set[s]
 2  last ← Starting-Point[s]
 3  while TRUE
 4        do next ← INTEGRATE-ODE(last, s, STEPSIZE)
 5           if Radius-Function[next] < SMALL
 6             then return REFINED-SEAMENDPOINT(s)
 7           for i ← 1 to Number-of-Elements[G]
 8                do if DISTANCE(G[i], next) ≤ Radius-Function[next]
 9             then return REFINED-JUNCTIONPOINT(next, s, G[i])
10           last ← next
```

$1 \rightarrow 2$. Let G be the set of all boundary elements not contained in the governor set of the seam s. Initialize *last* to be the last point visited, which at the beginning of the routine is the starting point of the seam.

$3 \rightarrow 9$. Integrate the system of differential equations a small distance on each iteration through the loop. If the radius function drops to a small value, we are approaching a seam-endpoint, so end the tracing and call REFINED-SEAMENDPOINT to compute the convex vertex lying at the intersection of the seam governors and return it. Otherwise, if the distance to any other element becomes less than or equal to our current radius function distance, we are done tracing also. In this case, we call REFINED-JUNCTIONPOINT to correct the value of *next* so that it is equal to the actual junction point resulting from the seam governors and the newly close element $G[i]$. Inside of REFINED-JUNCTIONPOINT we also check for degeneracy by checking distances to all boundary elements and adding those

elements at the same distance from the new junction point.

10. Otherwise continue by setting *last* equal to *next* and integrating again.

6.4.1 Implementation of Seam Traversal

First, it is obvious that checking the distance to each boundary element at every step is extremely wasteful. Fortunately, it is actually rare that we have to do such an exhaustive check. We have already performed a distance check against all boundary elements at the junction point which starts the seam s. Based on this check, we know exactly how far it is safe to trace before we should check again. For example, suppose the radius function equals r at the starting junction point but the distance to some other boundary element is equal to $r' > r$. We can trace a distance of $\frac{r'-r}{2}$ safely along the seam before the other element even has a chance of becoming active. (This is easy to see since along any curve, the distance to a boundary element may never increase or decrease at faster than unit speed. See Wolter [87] [88] for more details.) Therefore, we can use the following device:

1. Keep track of the distance traversed on the curve so far. Since our differential equation is arc-length parameterized, this simply means adding STEPSIZE to a counter at each step.

2. Keep a sorted list of "distance events." This list is sorted from smallest distance to largest, and each element contains a distance at which a boundary element must be re-checked. Thus in our example above, the first distance event resulting from the boundary element at a distance r' will occur at $\frac{r'-r}{2}$. As we trace along the seam, check the smallest element on the distance event list to see if a boundary element needs to be re-checked.

3. Once a distance event is executed, delete it from the list. If the distance from that boundary element is still far away, insert another distance event at the appropriate point in the sorted list.

The second implementation issue is the determination of STEPSIZE. We recommend a small enough stepsize to achieve good accuracy in the tracing of the seam curve and to avoid stepping over too many distance events at once, but not so small that the process is overly slowed. Since the differential equations for polyhedra tend to be quite stable, a

large stepsize of around 10^{-2} or 10^{-3} may be perfectly sufficient to achieve an accuracy of around 10^{-8}, assuming that the polyhedron has a length scale of about 1. However, we recommend decreasing the stepsize once the distance events are no longer far away. This will generate increased accuracy as we approach a new junction point. It is also wise to decrease the distance slightly when approaching the boundary, to make sure that we are actually approaching a seam-endpoint and not a junction point with a small radius function.

A third implementation detail is a useful tool for interrogation. It is necessary to record points along the seam curve every so often so that a polygonal approximation to the seam curve may be kept for interrogation and easy visualization.

Several optimizations may be made to the algorithm beyond the ones outlined above. For example, many of the seams will be linear and therefore it is a simple matter to intersect the line containing the seam with the offset sweeps of each boundary element; at worst a solution of a quadratic equation for each element will be required, since the intersection is the substitution of a set of linear parametric equations into an implicit equation which is at worst quadratic. With this intersection scheme, it is a simple matter to determine the endpoint of any seam.

A similar approach is possible for nonlinear seams, for example a seam resulting from two planes and a non-convex edge. The equations are more complicated in this case, but certainly the approach provides an alternative to our ODE approach. Why then do we choose to use the ODE approach? The principal reason is one of simplicity; our method eliminates the need for exhaustive case-checking. Furthermore, the ODE trace is fairly painless and fast. Finally, it is extensible to more complicated objects with curved boundaries.

6.5 The Completeness of the Algorithm

In order to use the recursive part of the algorithm above, we must ensure that every element in the set of edges and vertices of the Medial Axis (which we will call the *1-skeleton*) may be reached during the recursive part of the algorithm. We have already shown in theorem 3.16 that the Medial Axis of a polyhedron is connected, but this unfortunately does not always imply the connectivity of the 1-skeleton. Consider a long, wide, but very short rectangular box, and drill a small square hole from the top face to the bottom face (see figure 6-3). Then there is a sheet equidistant to the top and bottom faces which is bounded by two

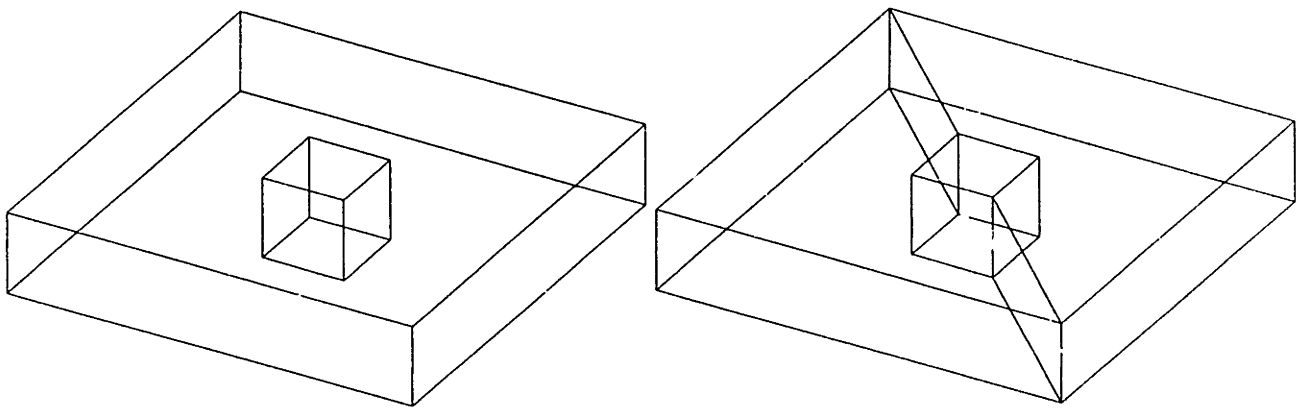

Figure 6-3: Subdividing multiply connected faces of a polyhedral solid into simply connected faces.

disconnected pieces of the 1-skeleton; one piece is near the outer loop of the top face, and the other piece is near the inner loop.

Fortunately, as we will show below, if the faces of the polyhedral solid are *simply connected*, then all sheets will be simply connected as well and therefore each sheet will be bounded by a single external loop. Since by theorem 3.16 the set of sheets and bounding loops (which forms the Medial Axis) is path connected, the entire 1-skeleton is path connected. To see this, pick two points on the 1-skeleton, and join them by a path on the Medial Axis. Move this path so that instead of crossing through the middle of each sheet, it simply walks along the boundary and therefore stays on the 1-skeleton.

Suppose we are dealing with a polyhedral object with multiply connected faces, like the thin box with the small hole mentioned above. Then all that is needed is to split each multiply connected face into a number of simply connected faces; for example, the top and bottom faces of the thin box may be split in half by adding two edges connecting the hole to the outer boundary loop, as shown in figure 6-3. The new edges are not considered boundary elements since they are coplanar with their adjacent faces, but each new face is considered a separate boundary element. In Appendix B, we give an algorithm for performing the decomposition in the general case.

Recall that in Chapter 5 we prohibited coplanar adjacent faces because the offset sweeps of the two faces would be identical and therefore the system of governing equations of a junction point would be singular. Of course, a similar problem occurs in the case of non-convex polyhedra, but we cannot join coplanar adjacent faces together without sacrificing simple connectivity. Therefore, some care is needed in tracing seams and refining junction

points.

Fortunately, the fix to the problem is an easy one. Suppose we are considering the two coplanar top faces in figure 6-3 as two governors for a seam. Note that the equidistantial set between the two coplanar faces is a plane perpendicular to both faces passing through the common edge. This set is also the equidistantial set between one of the faces and the common edge, so it may be expressed as the intersection between the offset sweep of a plane and the offset sweep of a line, thus removing the degeneracy. A similar approach is used in refining junction points; simply remove one of the two offset sweeps of the same plane in the system of governing equations and replace it with the offset sweep of the common edge.

6.5.1 Sheets are Simply Connected

There are many definitions of simple connectivity; the best for our purposes is found in Levinson and Redheffer [49]:

Definition 6.1 Let $D \in \Re^2$. D is said to be *simply connected* if the interior of every simple closed polygon in the domain is contained in the domain.

Simple connectivity is preserved over a homeomorphism. Since all of the sheets which occur in the Medial Axis of polyhedra are parts of planes or quadric surfaces (which are homeomorphic to $\Re^2$), we can speak of a polygon on these sheets by referring implicitly to this homeomorphism.

Theorem 6.2 Let S be a sheet on the Medial Axis of the polyhedral solid P without cavities. Assume every face of P is simply connected, and let P_1 and P_2 be the governors of S (where either governor may be a face, edge, or vertex). Then S is simply connected.

Proof: Note that any sheet S governed by two governors P_1 and P_2 is a piece of the set of points $A = \{\mathbf{x} \in \Re^n : d(\mathbf{x}, A_1) = d(\mathbf{x}, A_2)\}$, where A_1 is the analytic extension of P_1 and A_2 is the analytic extension of P_2. (That is, if P_1 is a face, A_1 is a plane containing the face, and if P_1 is an edge, A_1 is the line containing the edge. If P_1 is a vertex, then $A_1 = P_1$.) The portion of A which is also on $M(P)$ results from intersecting A with a finite number of other equidistantial sets and trimming A along the intersection curves. It has a finite number of connected components, and therefore is made up of a finite number of connected sheets. One of these sheets is S.

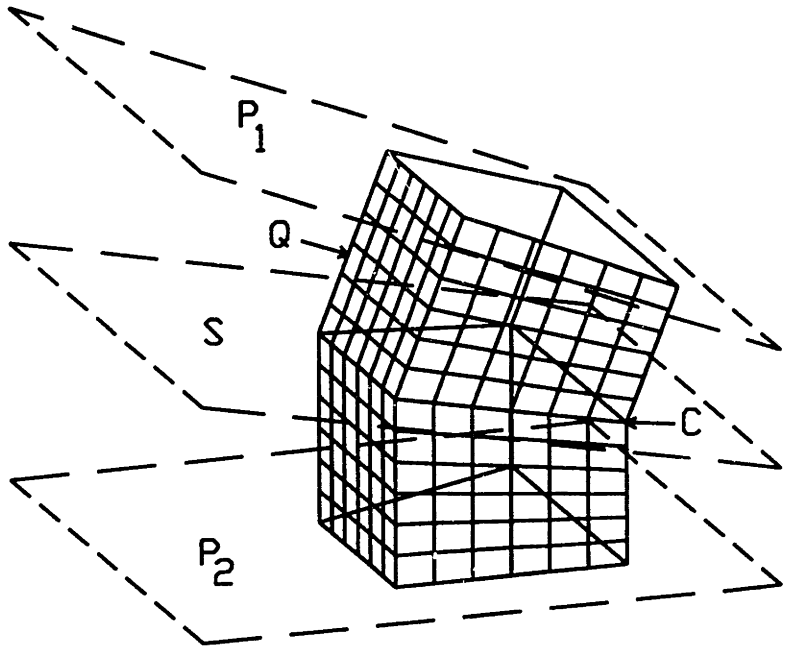

Figure 6-4: Constructing a closed region Q from a polygon C on the sheet S.

Now, of course A is simply connected; consider all possible equidistantial sets between points, lines, and planes to verify this. To prove that S is simply connected, take a closed polygon C contained in S. Since $S \subset A$ and A is simply connected, the interior of C is also contained in A. We need to show that it is also contained in S.

Draw lines from every point of C to its footpoints on P_1 and P_2. Because the radius function on the Medial Axis is continuous, C maps to closed curves on P_1 and P_2. A closed solid Q may now be constructed by taking the network of lines to form the side of the solid and taking the interior of the closed curves on P_1 and P_2 as the top and bottom respectively (see figure 6-4). The solid must be closed since C is continuous, and so are the directions and lengths of the lines drawn from C to P_1 and P_2 by the continuity of the distance function (see Wolter [87]). Furthermore, the top and bottom are completely contained within P_1 and P_2 respectively because the two governors are simply connected. Notice that every point on the boundary of Q has a closest point to ∂P either in P_1 or P_2 by construction.

Now pick a point $\mathbf{x}$ inside Q and assume that it is closer to some $y \in P' = \partial P - P_1 - P_2$ than to any point in P_1 and P_2. Then at least part of P' must lie inside Q. For, if P' were outside Q, the minimal join from $\mathbf{x}$ to $\mathbf{y}$ would intersect the boundary of Q at some point $\mathbf{z}$. Since every point on the boundary of Q is on a minimal join to P_1 or P_2, this would imply that $\mathbf{z}$ was on two minimal joins and therefore on the Medial Axis. But no minimal join may be extended backwards past a point on the Medial Axis and remain a minimal

join, as shown by Wolter [88]. Therefore at least part of P' must lie inside Q.

Since P contains no cavities, P' must be joined to the rest of ∂P. In addition, Q is a proper subset of P since a polyhedron has more than two boundary elements, so there is part of ∂P outside Q. Thus P' must cross the boundary of Q in order to be connected to the rest of ∂P. Of course, P' cannot be connected to ∂Q at the top or bottom of Q because every point on the top or bottom must lie wholly in P_1 or P_2. But P' cannot intersect the sides of ∂Q either because every point on the side has a nonzero distance to the boundary. Therefore we have achieved a contradiction, no $\mathbf{x}$ can exist which is closer to P' than to P_1 and P_2, and every point inside Q must have minimal joins to either P_1 or P_2.

Now pick a point $\mathbf{x}$ in the interior of C. Since $\mathbf{x} \in A$, $d(\mathbf{x}, A_1) = d(\mathbf{x}, A_2)$. Let $\mathbf{y}$ be the closest point to $\mathbf{x}$ on A_1. Clearly $\mathbf{y}$ lies inside P_1 as well; if not, then the minimal join to $\mathbf{y}$ would have to cross the boundary of Q in order to reach A_1. Since A_1 is tangent to P_1, it is a shorter distance to P_1 from the boundary of Q than to any other part of A_1. Similarly, the closest point to $\mathbf{x}$ on A_2 is also in P_2. Thus $d(\mathbf{x}, P_1) = d(\mathbf{x}, P_2)$. Furthermore, $d(\mathbf{x}, P') \geq d(\mathbf{x}, P_1)$, so $\mathbf{x} \in S$. Since $\mathbf{x}$ was an arbitrary point inside C, S is simply connected. □

6.6 Starting the Algorithm

In order to start the algorithm, it is necessary first to identify those objects which may become governors of branches of the Medial Axis. These objects are all the faces of the polyhedral solid, combined with all non-convex edges and vertices. To locate non-convex edges, we simply check the angle between the adjacent faces; if it is greater than π, then the edge is non-convex. Non-convex vertices are those vertices having at least one non-convex edge adjacent to them. All other edges and vertices are marked as convex for use as seam-endpoints and convex edge rims. In addition, any non-simply connected face must be split into a set of simply connected faces. The edges joining the resulting adjacent coplanar faces are convex, so they are not governors; however, they are not on the Medial Axis either, so they will not be kept for use as rims. Similarly, the new vertices joining such edges to one another are ignored except for bookkeeping purposes.

The next step is to find a starting vertex for the algorithm. Clearly any vertex which has at least one convex edge incident is part of the Medial Axis since it is a limit point of the

convex edge. Therefore, select such a vertex as a starting vertex for the algorithm. Choose three of the adjacent active boundary elements as a governor set for a seam connected to that vertex. Check the seam for feasibility and degeneracy, as before. (There will always be at least one subset of elements that generates a seam since the vertex is connected to the 1-skeleton). Trace the seam using our seam traversal routine until a junction point is reached. Use this junction point to start the recursion as in section 6.2.

6.7 Determining Sheets

Once the algorithm above has been run, it is a simple matter to locate all sheets in the Medial Axis. This process is easier than locating seams since there are no degenerate sheets. (This is easy to see since there is no way to make the offset sweeps of three distinct boundary elements intersect in a surface.) After the traversal described above, we are left with a connected graph of vertices and edges, where the vertices are junction points or seam-endpoints and the edges are seams or convex edges. The sheet location process starts at a vertex and travels around a loop of edges and vertices adjacent to each sheet, connecting each element to the sheet as we go. At any given vertex, we take every subset of two elements as a candidate sheet and add it to a list of candidate sheets waiting to be traversed. For each candidate sheet, we try to form a simple closed curve formed of edges and vertices such that each edge and vertex on the curve has the two governors of the candidate sheet among its governors. We know that every sheet in the MA is simply connected, so each is bounded by only one simple closed curve.

Usually a vertex has two edges incident with the same two-element subset, and these two edges are bounding edges to a sheet. However, sometimes there may be more than two edges incident. For example, two different sheets which are equidistantial to the same boundary elements may be tangent to each other at a common vertex, so that the boundary of the sheets forms a connected figure 8 curve. To ensure that we recognize this type of case as two separate sheets, we organize our algorithm so that as we traverse around loops, we keep the sheet to our left at all times. We can use the Singular Value Decomposition to determine the tangent plane to the sheet at the starting vertex (see section 4.4.2). Once we have the tangent plane, find a basis of three orthogonal unit vectors as follows: the first vector is the normal to the tangent plane, the second is the tangent to the edge at the

vertex, and the third vector is one of the two orthogonal unit vectors in the tangent plane. Since the first two vectors are known, a third is easily computed by making an orthogonal basis (see Golub and Van Loan [33]). Only one of the two possible orthogonal unit vectors will point towards the sheet; moving in the opposite direction will bring us too close to the boundary (or even outside the boundary, if the edge is a rim). If the edge is a rim, the correct direction is the one which keeps us inside the boundary; if the edge is a seam, the correct direction is the one which increases the distance from the other governors of the seam. Thus for each edge, we now have an associated direction to keep the sheet on the left as we traverse. If there is more than one edge which keeps the sheet on the left as we move away from the vertex (when, for instance, two sheets with the same governors meet at a junction point), we simply select the edge which turns most sharply to the left.

If we can move around a closed loop keeping the sheet on our left at all times, then we have determined a sheet. After checking all possible combinations of two elements, we mark the vertex as done, move to the next vertex (whether seam-endpoint or junction point), and perform the same process over again, looking for an adjacent seam or rim with the two elements as two of its governors. The process is complete when all vertices have been visited.

We note that visualization of sheets is fairly straightforward with modern graphics libraries. Since sheets may be easily expressed as trimmed NURBS patches, where the trimming curves are the approximations to the seams determined during the tracing of the seams, the sheets may be visualized as trimmed NURBS patches using standard graphics libraries.

6.8 Representation of Seams and Sheets

Although the ODE approach is simple to implement and efficient, it may be desirable to have an exact representation of the Medial Axis. Any sheet may be expressed as a trimmed quadric surface (where the trimming curves are the adjacent seams and edges). As described by Dutta and Hoffmann [26], the equidistantial set between any two objects, either or which is a point, line, or a plane, will always be a simple quadric surface. The seam curves themselves may be expressed as intersections of two of their adjacent sheets; the representations of these curves are also described in [26].

6.9 Algorithm Analysis

6.9.1 Complexity

Clearly, the major bottleneck in this algorithm is in the seam tracing phase. The amount of time required in this phase is a function of the number of steps taken along each seam, s, and the number of boundary entities n. At each step, we may need to perform up to n distance checks to see if another boundary element is becoming active, so in the worst case, each seam will be traced in $\mathcal{O}(ns)$ time. However, in the average case, we anticipate performing no more than n distance checks at the beginning, and then one or two distance checks at each step, when we use the distance event handling described above. Thus in the average case, the cost should be approximately $\mathcal{O}(n \log n + s)$ per seam and therefore $\mathcal{O}(n_S(n \log n + s))$ to trace all the seams and junction points, where n_S is the number of seams. The $\log n$ term comes in because in order to process distance events, we have to maintain a priority queue sorted by distance. The complexity for an insertion, deletion, or lookup is $\mathcal{O}(\log n)$ per operation [22].

In general, other components of the algorithm (such as the feasibility/degeneracy check and the determination of sheets) should execute much faster than the seam tracing phase, so we conclude that the total time complexity of the algorithm is $\mathcal{O}(n_S(n \log n + s))$ in the average case.

6.9.2 Stability

Most of the stability considerations outlined in chapter 5 still hold for this algorithm. However, there are a couple of important distinctions.

Junction points are no longer determined by solving a system of linear equations. Instead, their existence is ascertained during the tracing of seams, and they are subsequently refined with a minimization method or an equation solver. The accuracy of the junction points is therefore dependent on the initial approximation obtained by the ODE trace as well as the method of refinement.

Although it possible to implement this method in rational arithmetic, that option is extraordinarily unattractive, since the lengths of numerators and denominators will grow extremely long during the course of tracing each seam. A more sensible option to control error is the use of interval arithmetic. However, interval arithmetic adds its own complexity;

a method for computing the Singular Value Decomposition of an interval matrix is needed, as well as an ODE solver which works in interval arithmetic. Therefore, we have chosen to implement our algorithm with floating point arithmetic and two tolerances. The first tolerance specifies that as we trace along each seam, the accuracy of every intermediate point along the seam must be within ϵ. The second tolerance states that two points are considered the same point when they lie within δ of one another. This tolerance is also used for checking distances to the boundary; if, for example, a governor lies at distance d from the point p, then another boundary element is considered a governor of p also if its distance to p is within δ of d. These tolerances may be quite strict; the Singular Value Decomposition method is generally stable, and we can compute offset sweeps and their derivatives without ever leaving the stability of the Bernstein basis. It is important, however, that we determine junction points accurately, or else questions of degeneracy may be answered incorrectly. As a result, we force junction points to be determined within δ of the true value during refinement.

6.10 Examples

In this section, we show a few examples of our algorithm in action. These examples were run at an ϵ tolerance of 10^{-6} and a δ tolerance of 10^{-4} (all examples have a length scale of about 1). Our first example (figure 6-5) shows a rectangular box which has had a rectangular bite taken out of it. Notice the degenerate seam in the upper part of the L-shaped box, and the quarter circle seam due to the non-convex edge. The quarter circle forms the common base of two quarter cones, one pointing out of the page, and one pointing into the page. The quarter cone pointing out of the page is equidistant from the front face of the box and the non-convex edge.

Our second, third, and fourth examples (figures 6-6, 6-7, and 6-8) show different levels of discretization of a half-cylindrical shell. The shell has a width of 0.25 units; the outer radius is 0.5 units and the inner radius is 0.25. The discretized shell is shown above its Medial Axis. Notice that at a fairly coarse level of discretization it is possible to achieve a visually accurate Medial Axis. For clarity, only the interior portion of the Medial Axis is shown for each object. The whole Medial Axis with the boundary is shown in figures 6-9, 6-10, and 6-11.

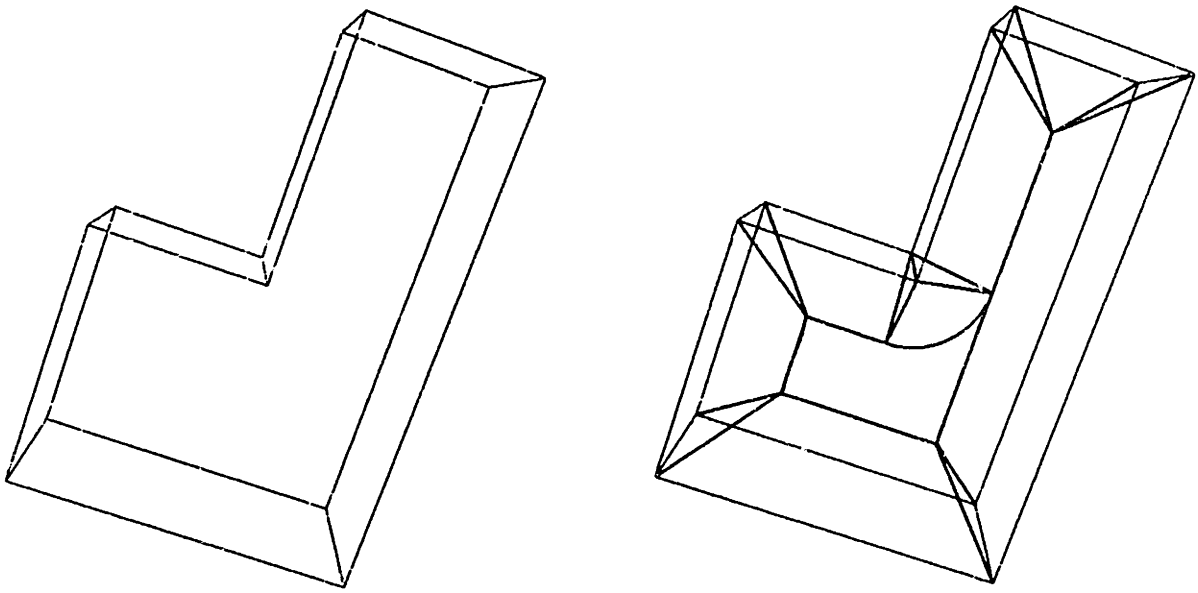

Figure 6-5: An L-shaped box and its MA.

Our fifth and sixth examples (figures 6-12 and 6-13) demonstrate the affect of collapsing one face of a tetrahedron. Note the non-convex, non-reflex vertex which has three seams incident to it in figure 6-13.

Our seventh example (figure 6-14) shows a cubical shell (a cube with a cube removed from the inside, creating a cavity) next to its interior Medial Axis. This example demonstrates the ability of the algorithm to handle solids with cavities in some cases. It also shows the behavior of the algorithm around reflex vertices. The Medial Axis with the boundary is shown in figure 6-15.

The last example shows a toroidal object and its Medial Axis. For clarity, the object is shaded in figure 6-16, and the interior portion of its Medial Axis is shown in wireframe. A wireframe of the object and its MA are shown together in figure 6-17.

Table 6.1 shows the CPU time needed to generate the preceding figures in terms of the number of boundary elements. We stress that these timings result from an unoptimized, experimental version of the algorithm, and should be taken only as an indication of the order of growth of complexity.

We expect that for similar shapes, such as the shapes in figures 6-6, 6-7, and 6-8, the complexity should grow approximately linearly. The graph in figure 6-18 shows that this is in fact the case. We calculate the Medial Axis Transform for five different objects representing different levels of discretization of a half-cylindrical shell (the three shapes

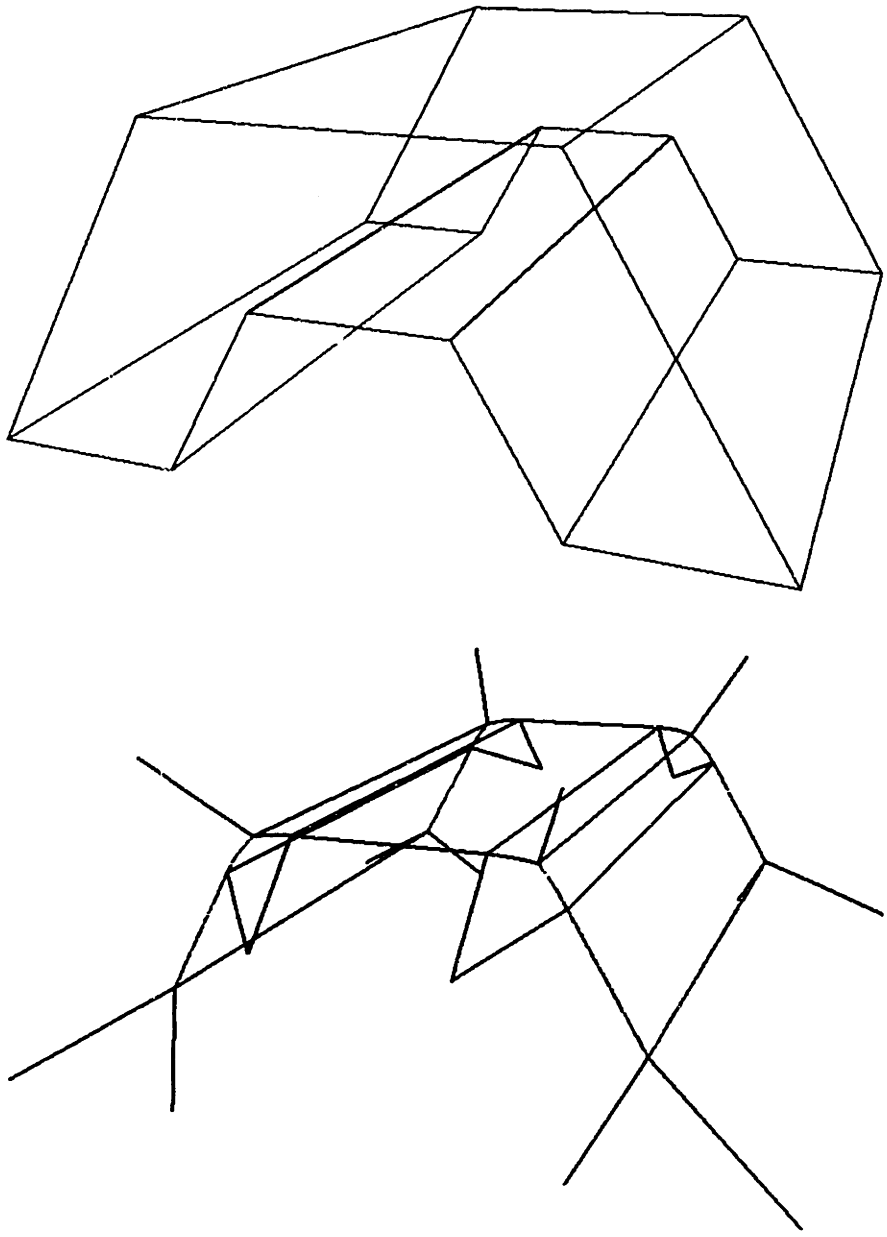

Figure 6-6: A discretized half-cylindrical shell with 12 boundary elements and its MA.

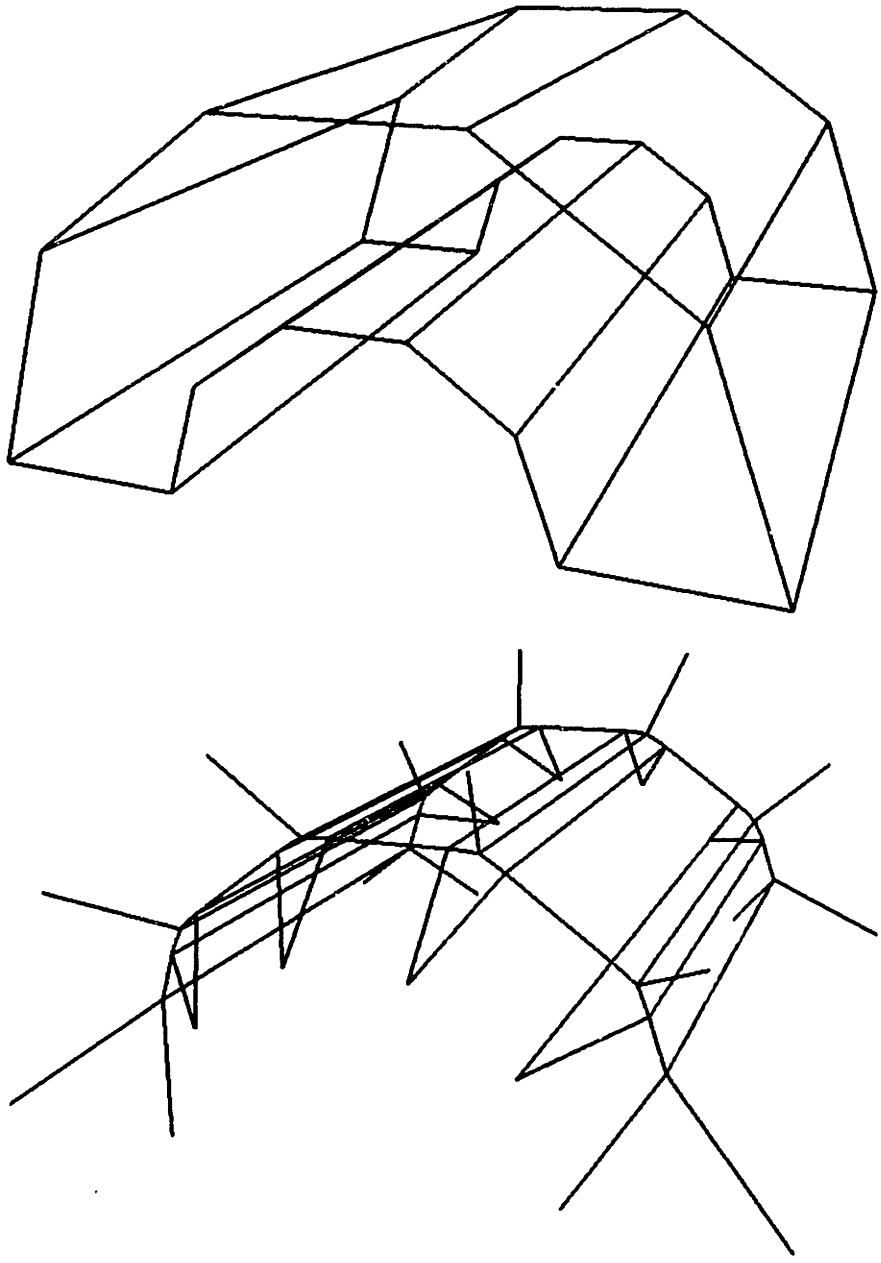

Figure 6-7: A discretized half-cylindrical shell with 18 boundary elements and its MA.

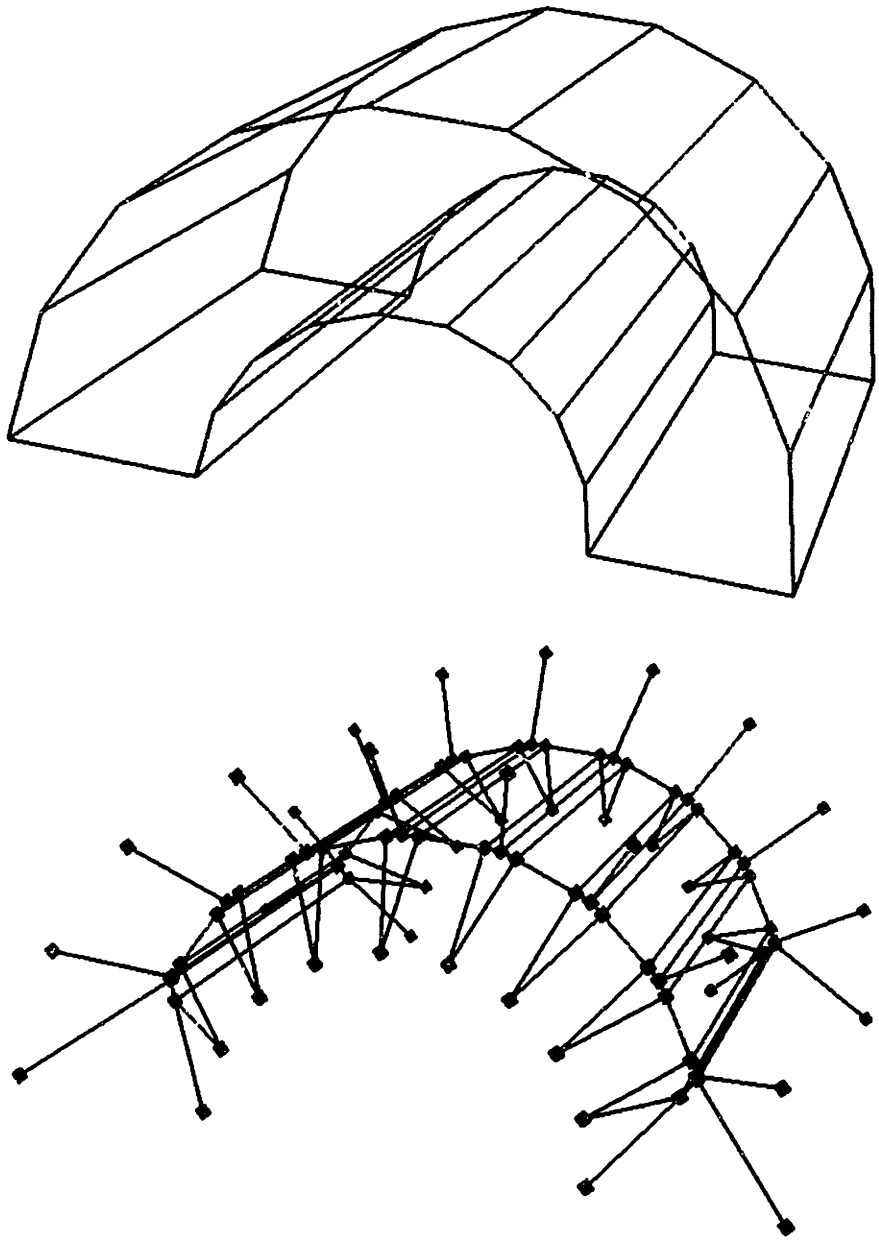

Figure 6-8: A discretized half-cylindrical shell with 30 boundary elements and its MA.

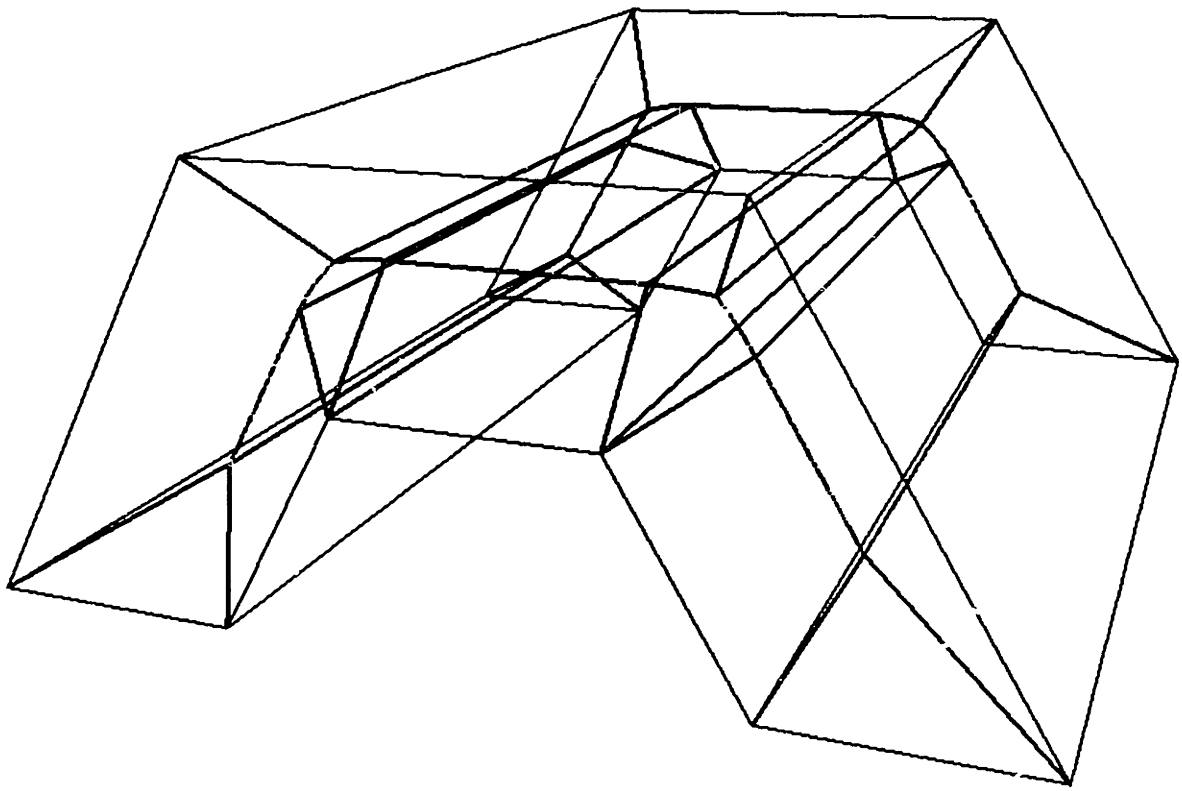

Figure 6-9: Boundary and MA of the 12-element shell shown together.

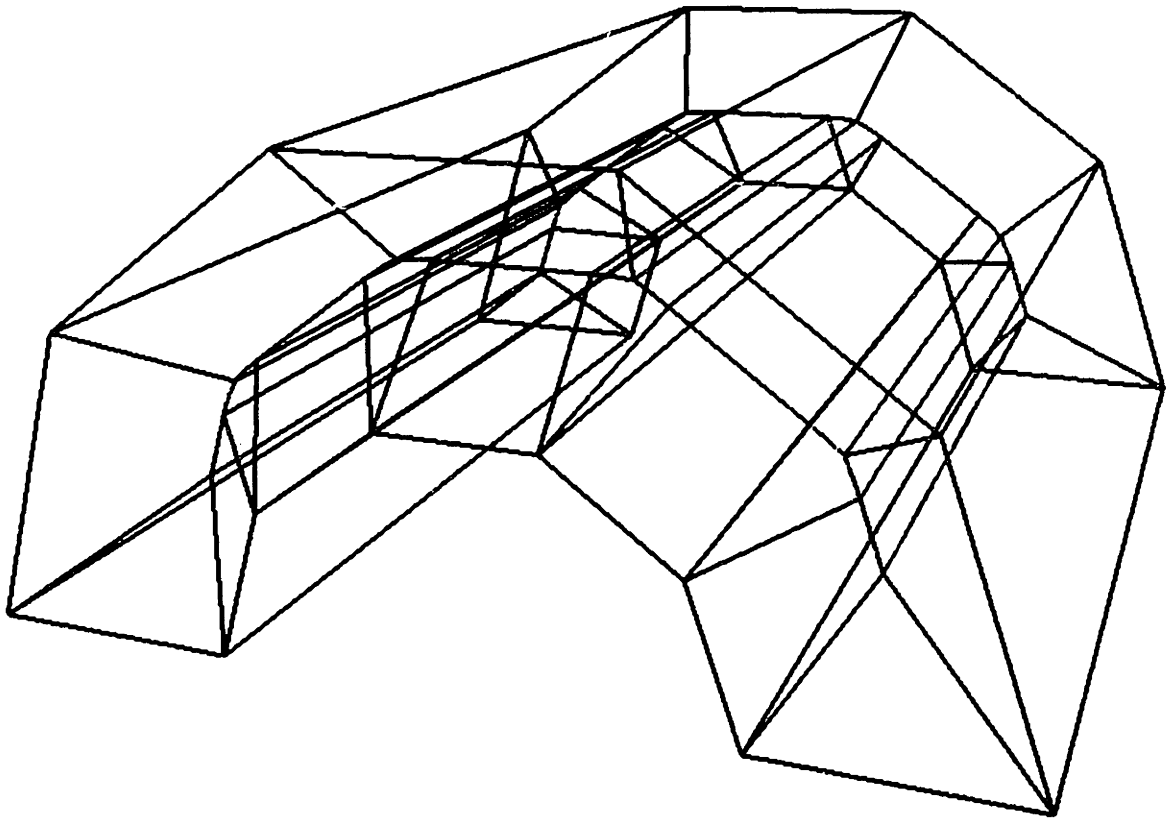

Figure 6-10: Boundary and MA of the 18-element shell shown together.

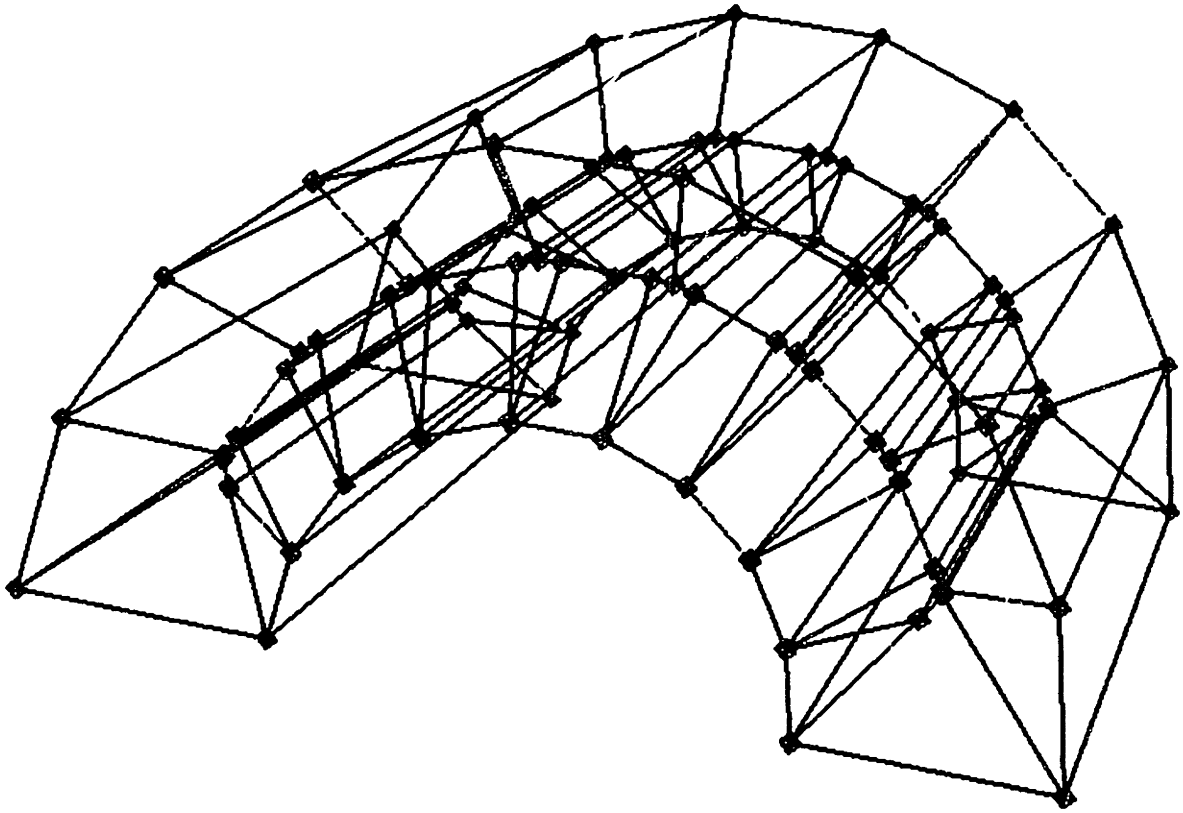

Figure 6-11: Boundary and MA of the 30-element shell shown together.

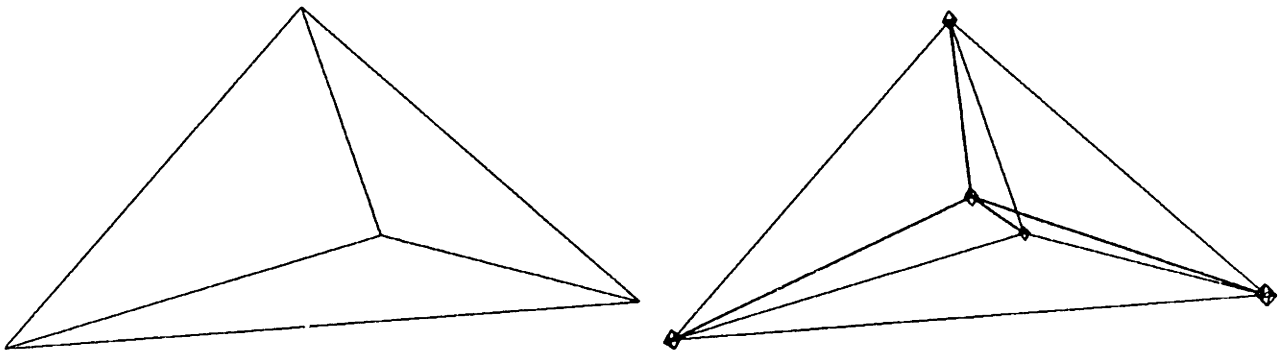

Figure 6-12: A tetrahedron and its MA

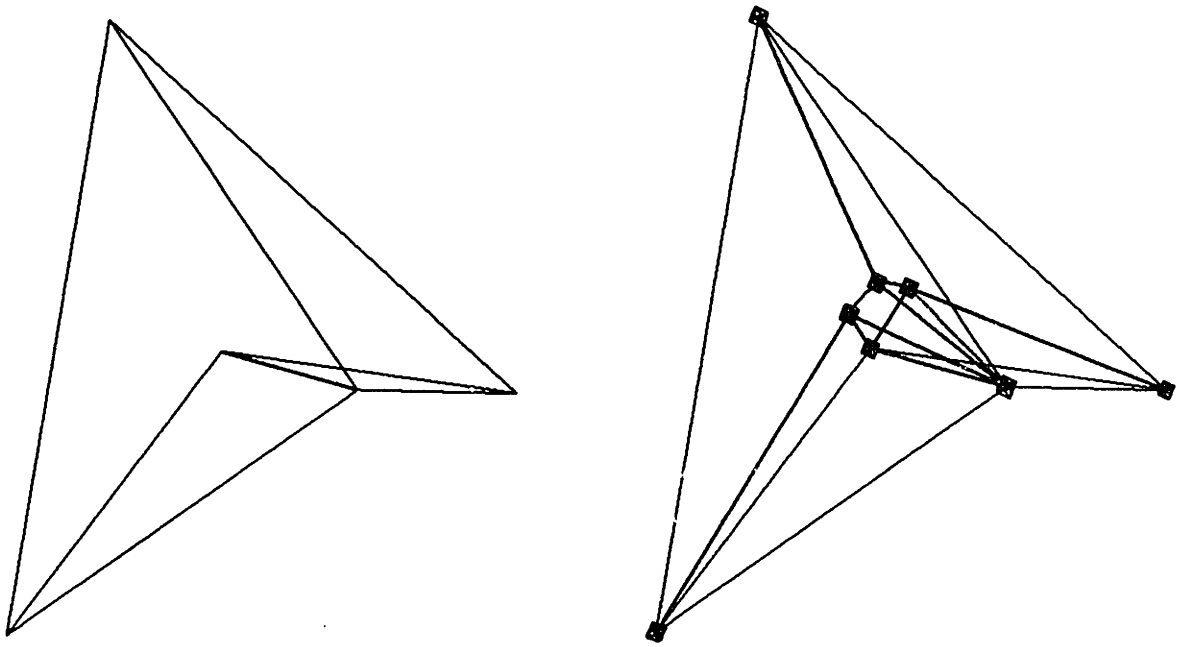

Figure 6-13: A tetrahedron with one of its faces collapsed into two faces with a non-convex edge and its MA.

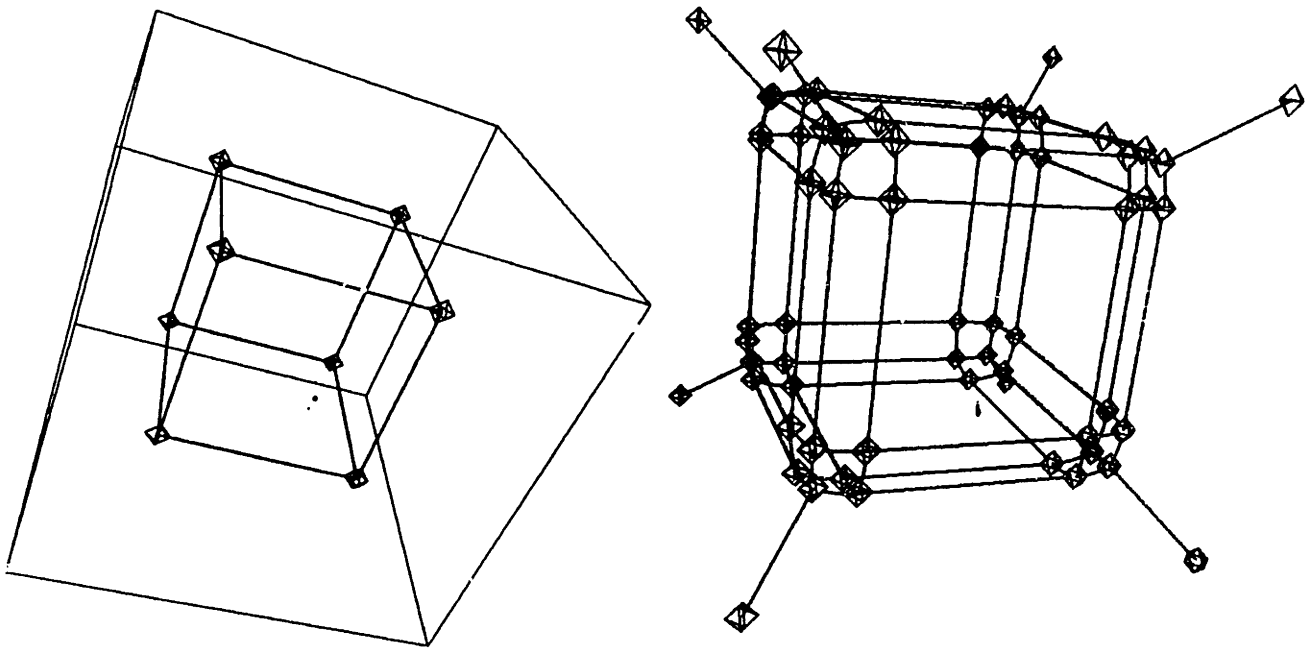

Figure 6-14: A cubical shell with 28 boundary elements and its MA.

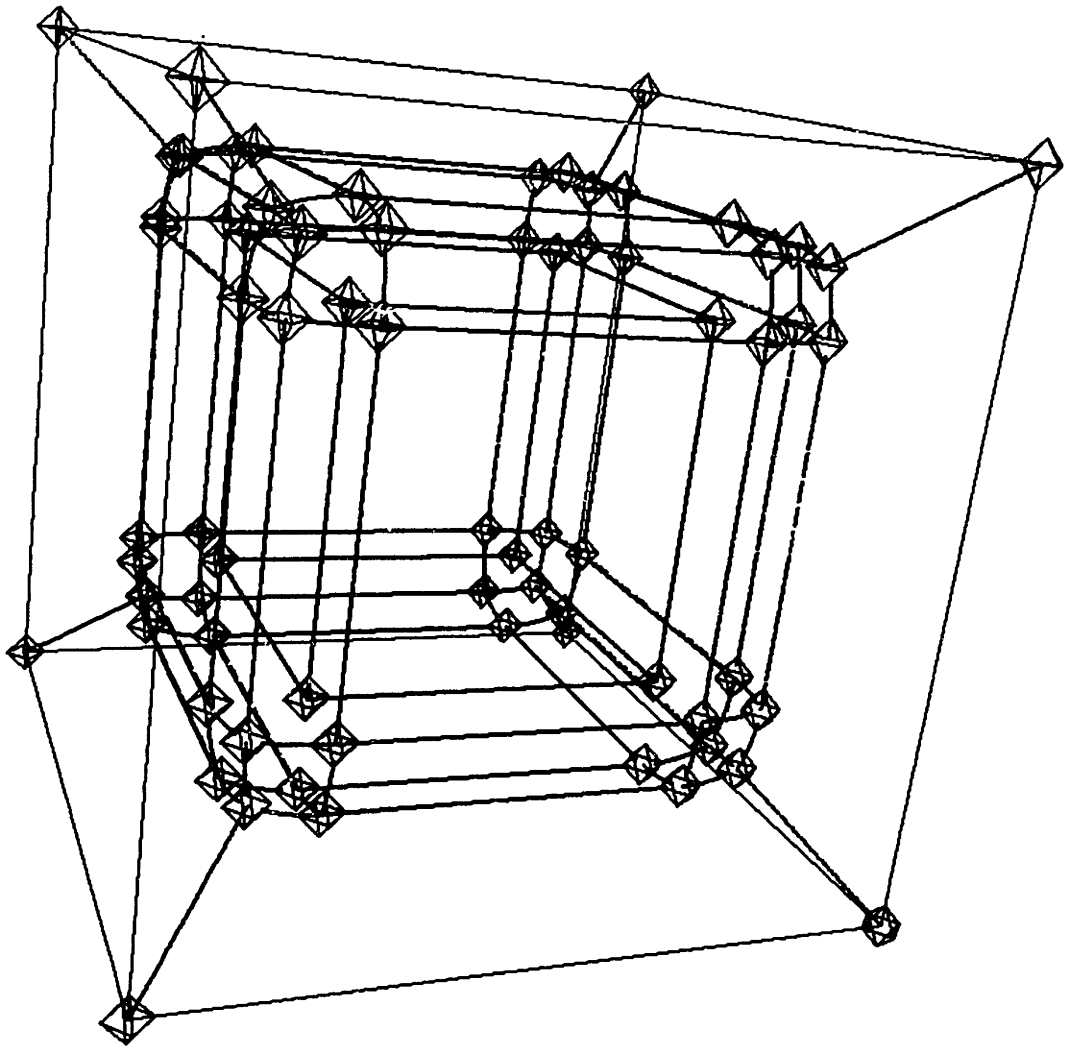

Figure 6-15: Boundary and MA of the cubical shell shown together.

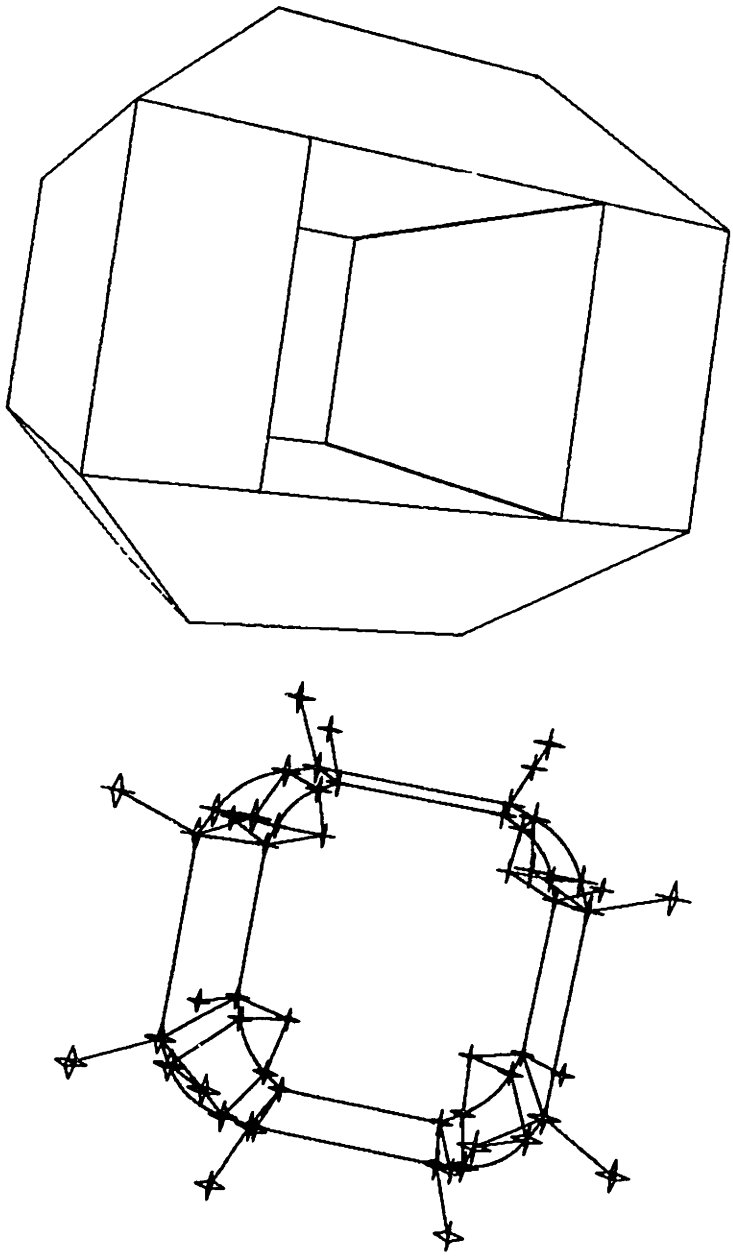

Figure 6-16: A toroidal object and its MA. The object is shaded for clarity.

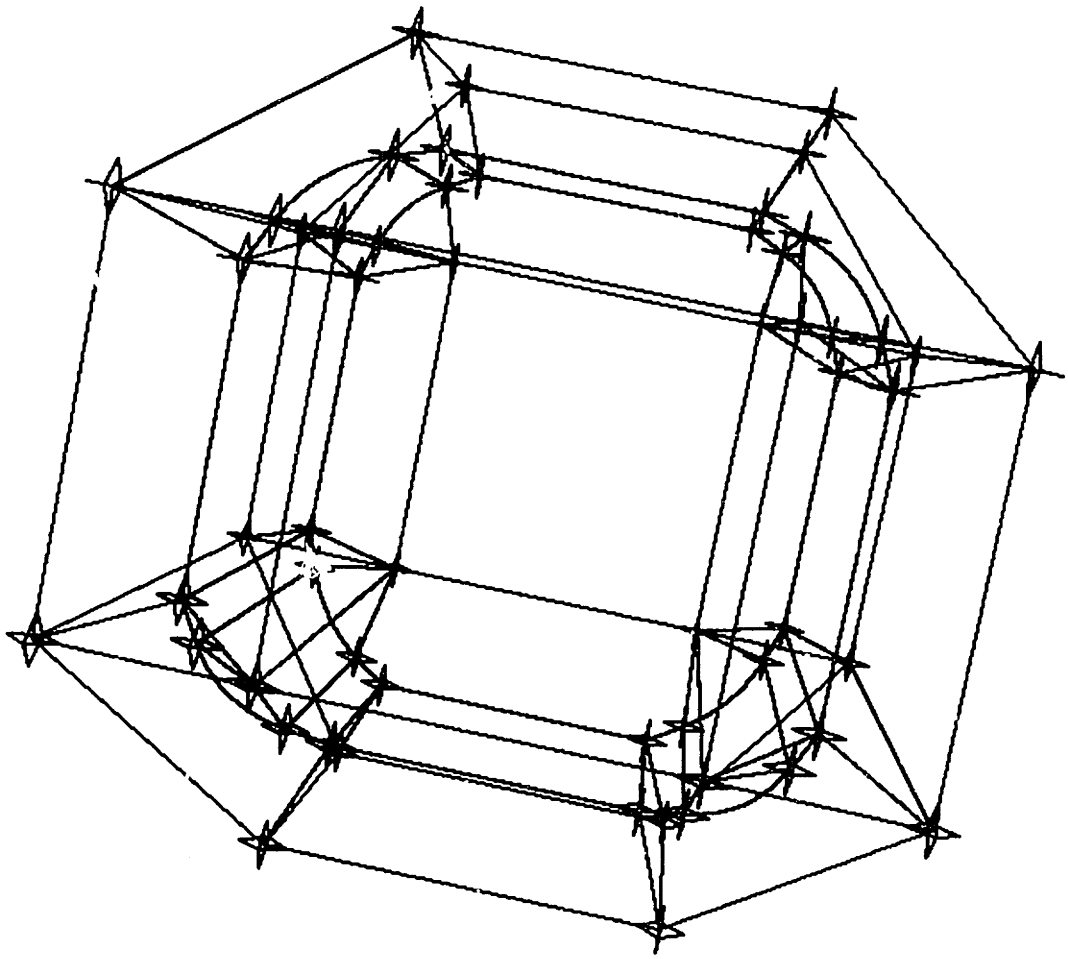

Figure 6-17: Boundary and MA of the toroidal object shown together.

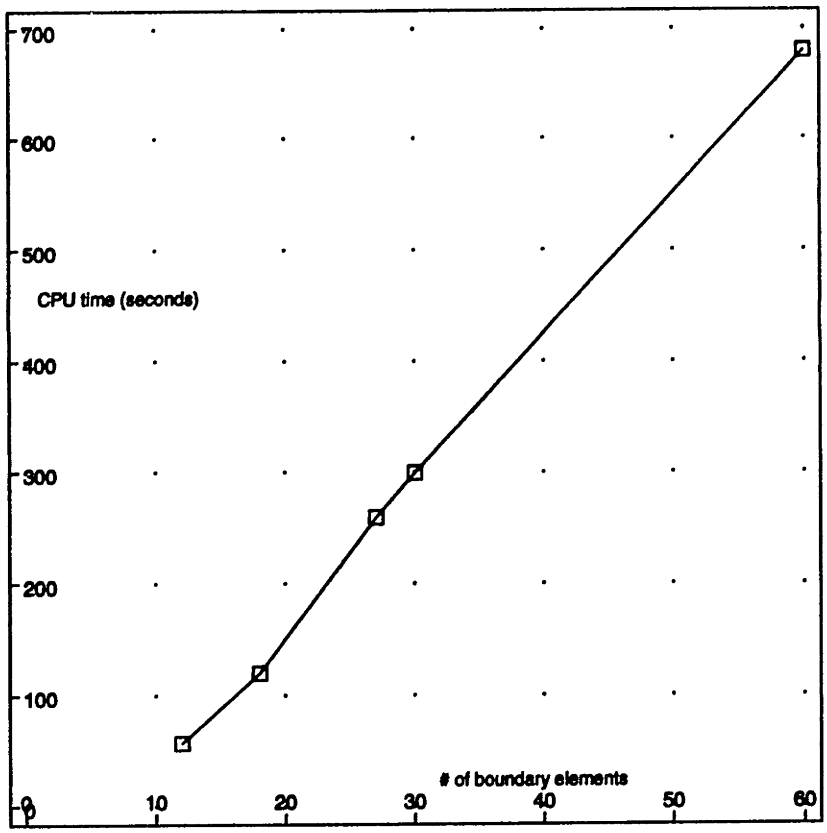

Figure 6-18: CPU times plotted against number of boundary elements for objects which represent increasing levels of discretization of a half-cylindrical shell

shown in figures 6-6 to 6-8) and plot the CPU times against the number of boundary elements.

Figure	# of elements	CPU time
6-5	9	5.8
6-6	12	60
6-7	18	120
6-8	30	300
6-14	28	350
6-16	28	124

Table 6.1: CPU timings for selected examples of the algorithm for non-convex polyhedra.

Chapter 7

Extension of the Algorithm to Solids with Curved Boundaries

7.1 Introduction

In the last chapter we saw how the algorithm for determining the MAT of convex polyhedra needed to be modified in order to work for non-convex polyhedra. In this chapter, we demonstrate that the basic ideas of the algorithm for non-convex polyhedra also extend to even more complicated boundary geometry with curvature. The organization of this chapter is similar to that of Chapter 6. We begin with a discussion of the essential differences between the MAT of solids with curved boundaries and the MAT of polyhedra. Then we describe each component of the algorithm (recursion at a junction point, traversing from a seam-endpoint, traversing a seam, etc.) in the same order as in Chapter 6.

Although we have not implemented the algorithm we will present in this chapter, we believe it to be feasible. Most of the algorithm structure remains the same (for example, the recursion from a junction point is virtually identical), so it is unnecessary to repeat the pseudocode from the previous chapter. When there is a difference in implementation, it will be stated in the appropriate place.

7.2 Differences in the Classification Scheme

Because of the curvature of the boundary, new types of points may arise. Although a point on the MA of a polyhedron may not have more than one footpoint on a given boundary

element, a point on the MA of a curved object certainly may. For example, the axis of a circular cylinder forms part of the MA of the cylinder, and each point on the axis has an infinite number of footpoints on the boundary of the cylinder. Another example is an object bounded by a parabolic cylinder and a plane; there is a sheet in the MA of this object such that every point on the sheet is equidistant from two points on the parabolic cylinder.

In addition, for curved objects, seam-endpoints and rims no longer need to be convex vertices and convex edges of the object respectively. As we saw in Chapter 3, such points may arise due to locally maximal curvature. Also, because of cylindrical or spherical boundary elements, it is possible for edges and vertices of the Medial Axis to have infinite numbers of footpoints.

It is apparent that the notion of boundary element that was so useful to us in designing algorithms for polyhedra becomes much less so when curved objects are concerned. Instead, we turn to the generalization of classification explored in section 4.3, where *neighborhoods of footpoints* are governors rather than boundary elements.

7.3 Traversing from a Junction Point

The recursion at a junction point generalizes from the polyhedral case. We take all three-element subsets of the set of governors of the junction point and investigate each subset for feasibility and degeneracy. Again, each governor is a neighborhood of a footpoint of the junction point.

The pseudocode of section 6.2 corresponds exactly to pseudocode for curved objects. The set of tangents is generated in exactly the same way, by finding equiangular vectors between the distance vectors to each subset of three governors, or by using the second order check when needed. As before, we trace along each feasible seam until we reach another junction point or a seam-endpoint.

7.3.1 The Feasibility Check

Feasibility and degeneracy are checked in much the same way as before. However, because the boundary is curved, it is possible for us to incorrectly flag a seam as feasible when it is not. Imagine a seam that is immediately infeasible due to a boundary element which is closer to the seam than its governors. However, after a short distance, suppose this

boundary element curves away rapidly so that it is farther away than the three governors. Our check could conceivably flag such a seam as feasible when in fact it is infeasible. A similar problem could arise when trying to decide questions of degeneracy; we could trace a small distance to perform the degeneracy check and a non-governing boundary element could be mistakenly flagged as a governor if at the test point it was coincidentally at the same distance as the seam governors.

There are a number of possible approaches to this problem, including the examination of higher-order derivatives. However, the simplest approach is to decide on a stepping parameter based on the curvature of the boundary. Ideally we would pick a stepsize in our feasibility/degeneracy check so small that the curvature of the boundary is insufficiently high to generate one of these problems. In other words, at the scale of our stepsize, the governors of the junction point will appear almost flat. Then we would use this stepsize for the check.

If it is difficult to determine such a parameter, we suggest performing a number of checks with different stepsizes. Clearly a degenerate governor must satisfy all of these checks, and a seam will be feasible only if all of the checks indicate that it is feasible. The more different stepsizes we use, the more likely it is that we have discovered the correct feasibility and degeneracy for the seam.

7.4 Tracing a Seam

As before, we have an initial tangent to the seam and a set of governing equations for the seam. Because the offset sweeps of each governor may not be easily expressed in the form $f^i(x, y, z, r) = 0$, we must turn to either the envelope formulation or the normal formulation of a seam curve shown in section 4.4.2. As in the polyhedral case, it is possible to differentiate the system of equations and use the SVD to find the tangent vector to the seam at a given point. As before, we trace until we reach another junction point or a seam-endpoint.

If we reach another junction point, we must refine it as before. The process of refinement is more difficult since the system of equations may be more complicated. However, a minimization technique on the sum of the squares of the governing equations may still be used just as before. If the envelope formulation is used, the Projected-Polyhedron method

[77] [76] of Section 2.6 may be used also; however, preliminary experiments suggest that the method may be too slow attempting to solve such a complex system.

The process of refining a seam-endpoint is somewhat different from the polyhedral case. As before, a seam-endpoint occurs when the distance to the boundary approaches zero; however, a seam-endpoint also occurs when two or more footpoints are approaching one another as we trace. Thus these cases must be handled separately. If the radius function is approaching zero, then we simply look for a convex vertex on the boundary, as in section 6.4. If, on the other hand, two or more seam governors are overlapping, we can solve a system of equations to refine the point. Consider the most common case, where there are three seam governors. Suppose the first two governors overlap at the seam-endpoint, and the neighborhood of the common footpoint is defined by the parametric equation $[x\ y\ z]^{\mathrm{T}} = \mathbf{f}^1(u_1, v_1)$. Suppose further that the third governor is defined by $[x\ y\ z]^{\mathrm{T}} = \mathbf{f}^2(u_2, v_2)$. The first defining equation is that the offset sweeps of the two neighborhoods must intersect:

$$\mathbf{O}^1(u_1, v_1, r) - \mathbf{O}^2(u_2, v_2, r) = \mathbf{0}. \tag{7.1}$$

As we saw in Theorem 3.6, end points result when the maximum principal curvature of a boundary element is equal to the curvature of the associated maximal sphere. Therefore, the following relationship must hold where the first two governors overlap:

$$\kappa_{\max}(u_1, v_1) - \frac{1}{r} = 0 \tag{7.2}$$

where $\kappa_{\max}$ is the maximum principal curvature of $\mathbf{f}^1$. Furthermore, the same theorem tells us that at the common footpoint the maximum curvature function must achieve a local maximum along the normal section in the direction of maximum principal curvature. Therefore, the directional derivative of the maximum curvature function in the direction of maximum curvature must be equal to zero.

$$\frac{\partial \kappa_{\max}(u_1, v_1)}{\partial u_1} \cos\theta_{\max}(u_1, v_1) + \frac{\partial \kappa_{\max}(u_1, v_1)}{\partial v_1} \sin\theta_{\max}(u_1, v_1) = 0 \tag{7.3}$$

where $\theta_{\max}$ is the angle of the normal section on which maximum principal curvature is achieved. (7.1) gives three equations (since the fourth condition $r - r = 0$ obtained from the equation is redundant), so therefore we have a system of five equations in the five

unknowns u_1, v_1, u_2, v_2, r. The root of this system may be found by minimizing the squares of the left-hand sides of the system, just as before.

We note that the optimizations explored in Section 6.4.1 apply to seam tracing for curved objects as well. The notion of distance events is still applicable and much more useful than before since distance checking is no longer as immediate as it was for polyhedra. It is important that each distance check itself be as quick as possible but ensure robustness in determining a solution. A robust method based on the Projected-Polyhedron method for determining distances from a point to a surface is discussed in Zhou *et al.* We refer the reader to [90] for more details.

7.5 Traversing from a Seam-Endpoint

Traversing from a seam-endpoint for curved boundaries is considerably more involved than it was for polyhedra. Seam-endpoints need no longer simply be convex vertices; they may result, as we saw in the previous section, from local maxima of curvature. If the seam-endpoint does in fact result from a convex vertex, we may proceed as before by connecting all adjacent convex edges to it. However, if it results from a maximum of curvature, we must trace out the adjacent rims by a differential equation method analogous to the one we use to trace out seams.

Assume that in a neighborhood of the footpoint of each endpoint, the boundary is defined by the parametric equation $[x\ y\ z]^{\mathrm{T}} = \mathbf{f}(u, v)$. Since each endpoint on the rim satisfies the curvature conditions given in section 7.4, the rim may be defined by the system of equations

$$\kappa_{\max}(u, v) - \frac{1}{r} = 0 \tag{7.4}$$

$$\frac{\partial \kappa_{\max}(u, v)}{\partial u} \cos\theta_{\max}(u, v) + \frac{\partial \kappa_{\max}(u, v)}{\partial v} \sin\theta_{\max}(u, v) = 0. \tag{7.5}$$

This is a system of two equations in the three unknowns u, v, and r. Using the SVD as in the seam tracing phase, it is possible to trace this system of equations in order to generate the rim curve. As with seam curves, we must check the distances to other boundary elements as we trace to ensure that the rim is distance minimal.

7.6 Starting the Recursion

The simplest way to start the recursion is to find a convex vertex on the object, and then proceed exactly as before. However, such a vertex might not exist in general. Therefore if a convex vertex cannot be found, we suggest finding the maximum of the maximum principal curvature function on various boundary elements. This procedure is discussed in Maekawa and Patrikalakis [51]. For each maximum that we find, travel inwards a distance equal to the reciprocal of maximum curvature, on the normal vector from the boundary. The point reached is on the Symmetric Axis; to check whether or not it is also on the Medial Axis, we must find distances to other boundary elements to ensure that the point is distance minimal. If it is not, we repeat the procedure with another maximum.

If the point is distance minimal, it lies on the Medial Axis, and since it satisfies the curvature conditions of an end point, it must lie on a rim curve. We trace this rim curve until we reach a seam-endpoint. Then, from the seam-endpoint, we trace along the associated seam to a junction point, and thus start the recursion.

7.7 Determining Sheets

Once we have a network of seams, junction points, seam-endpoints, and rims, the process of sheet determination is exactly as before. We locate the closed loops in the graph where each element of the loop has two governors in common with every other element of the loop. These loops form the boundary of a specific sheet.

The PDE approach outlined in Section 4.4.3 may be used to trace out a set of points on each sheet.

7.8 Exceptions to the Rule

Unfortunately, when we are dealing with curved objects, some exceptions to our general classification rule will arise. In dealing with polyhedra, every type of MA point falls nicely into a category: either junction, seam, end, seam-end, or sheet. However, as we saw in figure 4-3, MA points may have far more complicated sets of footpoints. Of these, the most common are those arising from objects such as cylinders, cones, and spheres which have constant positive maximum principal curvature over some area. These boundary elements

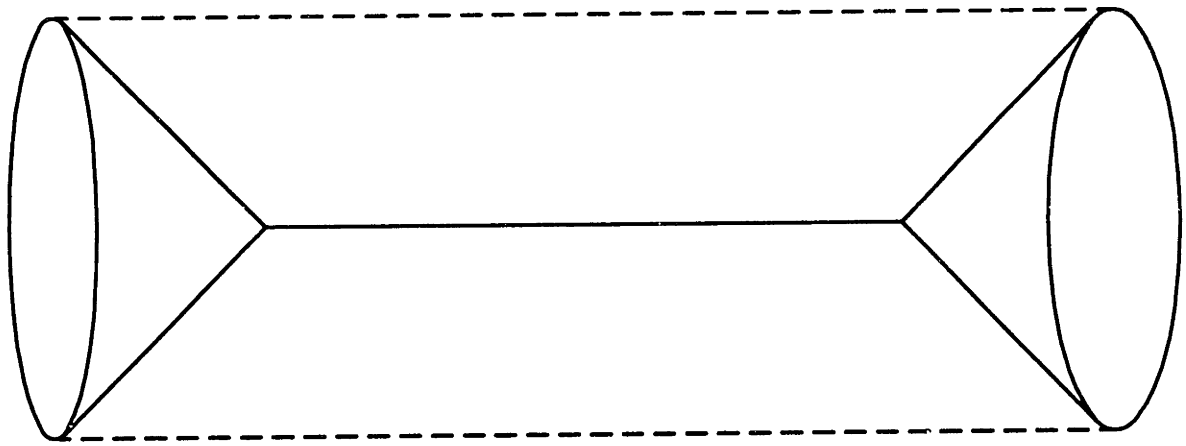

Figure 7-1: A right circular cylinder and its Medial Axis. The MA is outlined with a solid black line.

give rise to MA points with infinite numbers of footpoints.

When such boundary elements occur in an object, the algorithm will not work as given. To see why this is so, consider a right circular cylinder. There is a degenerate seam on the axis of the cylinder, which is connected to two conical sheets. Each conical sheet is also bounded by a circle lying at an end of the cylinder, since each circle is a convex edge of the cylinder. Notice that although the MA is connected, the seams and rims of the MA (the seam on the axis of the cylinder and the two circles at the base of the cylinder) are *not* connected, unlike our previous examples, because the boundary of each conical sheet is disconnected. Our algorithm depends on the fact that sheets are usually bounded by one connected closed loop, and because it traces out seams and rims by assuming they are connected, it will fail in this case.

This problem requires further investigation. Nevertheless, we can make some useful observations. The major problem with the cylinder is that the lateral boundary element is itself bounded by two disconnected loops. As a result, there are sheets on the MA which have disconnected boundaries. If we were to split the cylinder into a top portion and a bottom portion, considering each portion as a separate boundary element, we would no longer have any boundary elements with disconnected bounding loops. Of course, the MA which would be generated from such a boundary would be a superset of the actual MA, but we could trim away spurious portions of the superset after running the algorithm. For example, there will be a sheet generated which is equidistant from the top and bottom of the cylinder. This sheet would be replaced by the actual piece of the axis resulting from the cylinder, a degenerate seam. It is not clear how this process may be generalized, but

making boundary elements be bounded by closed loops seems to be a key component in adapting the algorithm to curved objects.

Chapter 8

Conclusions and Recommendations

In this chapter we summarize the chief contributions of this thesis, and also state some possible future research topics relating to our discussions in the previous chapters.

8.1 Contributions

1. In our definitions, we consolidate the many different definitions of the Medial Axis Transform and its related sets into a comprehensive whole.

2. Relationships between maximal balls in an object and curvature of the boundary of the object are developed for piecewise C^2 submanifolds of $\Re^3$.

3. A strong deformation retract is established between a submanifold of $\Re^n$ with a piecewise C^2, completely G^1 boundary and its Medial Axis. Therefore the MA of such an object is path connected if the object is path connected.

4. The Medial Axis of polyhedral solids of arbitrary genus with a connected interior is shown to be connected.

5. A classification scheme for different types of points on the Medial Axis is developed for 3-D convex polyhedra and extended to piecewise C^2 submanifolds of $\Re^3$. Governing equations for such points are also established for boundaries described by piecewise C^2 parametric curves and surface patches, and solution methods for such equations

are shown as well.

6. An efficient algorithm for accurately determining the MA of convex 3-D polyhedral solids is developed and demonstrated.

7. The algorithm is extended to arbitrary 3-D polyhedral solids of arbitrary genus without cavities, with non-convex vertices and edges. The completeness of this algorithm is shown by proving the connectivity of the 1-skeleton.

8. Both algorithms are tested, and their stability and complexity are analyzed. Implementation issues such as optimization are also examined.

9. The extension of the algorithm to submanifolds of $\Re^3$ with piecewise C^2 boundaries is explored.

8.2 Future Research Areas

1. Further investigation is needed into the mathematics of the Medial Axis Transform in $\Re^n$. For example, generalizations of the curvature theorems to higher than three dimensions would be useful. In addition, an analysis of how the Medial Axis Transform behaves under various types of perturbations of the boundary would give greater insight on the stability of the set.

2. Improved solution techniques for determining the different types of points that make up the Medial Axis are needed. For example, determining junction points resulting from four parametric boundary elements is an expensive computation.

3. Fast, robust distance computations are needed to run our algorithms. Further investigation into optimal algorithms for determining the distance from a point to different types of geometric objects would be useful.

4. The stability of the algorithms for polyhedra needs to be further investigated. In addition, both algorithms should be implemented in interval arithmetic to ensure robustness. Also, a variant of the algorithm for non-convex polyhedra could be developed that did not require tracing, based on the exact representations of the equidistantial sets involved.

5. The general polyhedral algorithm should be extended to all polyhedral solids with cavities. In addition, the MAT algorithm outlined in Chapter 7 for 3D solids with curved boundaries needs to be further developed, implemented, and extensively tested.

6. An investigation into a variant of the MAT could prove useful, especially in generating finite element meshes. This variant would compute the MAT on a surface in 3-D, but instead of using the Euclidean distance as a metric, the geodesic distance between two points would be used (i.e. the length of the shortest path between two points on a surface.)

7. The inverse problem (constructing a solid from its Medial Axis and radius function) needs to be investigated further in order to incorporate the MAT into a solid modeling environment.

8. Improved visualization methods for illustrating the Medial Axis and offsets of 3-D solids should be developed.

9. There is still extensive work to be done on applications of the algorithm. For example, a finite element meshing system could be constructed using our algorithm. The boundary of an object with a curved boundary could be meshed using the algorithm of Gursoy and Patrikalakis [37] and then this mesh could be taken as the boundary of a polyhedral solid. Using our algorithm, the Medial Axis Transform of this polyhedral solid could be determined, and then meshed using a similar approach.

Appendix A

Offset Sweeps of Simple Geometric Objects

There are three types of offset sweeps of entities in a polyhedron. The first is the offset sweep of a plane $\mathbf{x} \cdot \mathbf{n} + d = 0$ where $\mathbf{n}$ is the inward-directed normal to the plane and $\mathbf{x} = [x\ y\ z]^{\mathrm{T}}$. Using definition 4.1, we construct the offset sweep as

$$\mathbf{x} \cdot \mathbf{n} - r + d = 0. \tag{A.1}$$

The offset sweep of a non-convex edge is a growing circular cylinder. Suppose one of the two endpoints of the edge is given by the point (X, Y, Z), and let $[A\ B\ C]^{\mathrm{T}}$ be the vector pointing from (X, Y, Z) to the other endpoint of the edge. Then from Faux and Pratt [29], the equation of a cylinder of radius r is given by

$$(x - X)^2 + (y - Y)^2 + (z - Z)^2 - [A(x - X) + B(y - Y) + C(z - Z)]^2 - r^2 = 0. \tag{A.2}$$

If r is allowed to vary, then (A.2) is the offset sweep of a non-convex edge. Finally, the offset sweep of a non-convex vertex is a growing sphere. If (X, Y, Z) is the vertex, then a sphere of varying radius r is given by

$$(x - X)^2 + (y - Y)^2 + (z - Z)^2 - r^2 = 0. \tag{A.3}$$

Appendix B

Subdividing Multiply Connected Polygonal Regions

The algorithm presented in Chapter 6 requires all faces of the polyhedron to be simply connected in order to work correctly. Therefore, it is necessary during preprocessing to subdivide any multiply connected polygonal faces into a set of coplanar, simply connected faces. In this appendix, we discuss some possible approaches to this problem and also propose a possible algorithm to perform a simple decomposition.

An obvious approach to decomposing the region is by meshing the region with triangles or quadrilaterals. However, an automated robust algorithm to do this is not necessarily trivial, if the region is complex; in addition, it may generate considerably more polygons than are necessary. A better approach could be to use the Voronoi Diagram of a multiply connected region in order to decompose the region. The Voronoi regions will all be simply connected (one can use a much simpler version of the argument in theorem 6.2 to justify this fact). Furthermore, an algorithm by Srinivasan and Nackman [79] executes in $\mathcal{O}(n \log n + nh)$ time, where n is the number of boundary elements (edges and non-convex vertices) and h is the number of holes. Although this decomposition does generate some parabolic edges, the approximation of these edges with straight line segments maintains simple connectivity.

A simpler approach is an algorithm which decomposes the region by adding one hole at a time. It relies on the fact that a doubly connected polygonal region (a region with one hole) can always be quickly converted into two simply connected regions by adding two edges. To do this, pick two vertices on the inner loop, one with the highest y coordinate,

and one with the lowest. From the highest vertex, draw a ray pointing in the positive y direction and intersect it with every edge in the outer loop. The intersection point with the lowest y coordinate forms one endpoint of a new edge, and the starting vertex forms the other. Perform a similar process for the lowest vertex on the inner loop, except that the ray extends vertically downward, and the intersection with the highest y coordinate is one of the endpoints. The result will be two simply connected regions like the example shown in figure 6-4. Note that the subdivision process executes in linear time proportional to the number of edges and vertices in the region.

Based on the ability to subdivide doubly connected regions, we can design an incremental algorithm to decompose any multiply connected polygonal region. Assume that at a given step we have a set of simply connected regions, and we cut out a new hole from the domain. There are two possibilities for this new hole: either it lies completely inside one of the simply connected regions, or it lies partially in two or more such regions. In the first case, we will be left with a number of simply connected regions and one doubly connected region, which may be subdivided into two simply connected regions. In the second case, the hole intersects the boundary of some number of regions, and therefore the boundary of the hole becomes part of the boundary of these regions. Either way, we end up with a set of simply connected regions after each step.

We can start the algorithm with the bounding loop and one internal hole, then subdivide the region into two simply connected regions. Then we add one hole at a time until done. Let h be the number of holes and let n be the number of total edges (so that $\frac{n}{h+1}$ is the average number of edges per loop). For each of h steps, we have to check the intersection of the new loop with up to $2h$ edges, which is $\mathcal{O}(n)\mathcal{O}(h) = \mathcal{O}(nh)$ total time over all loops. (Notice that only the edges which have been added by subdividing double connected regions need to be checked, since the bounding loops cannot intersect each other.) Then, if each hole generates a doubly connected region, $\mathcal{O}(n)$ work is required to subdivide the region. Otherwise, if the hole intersects transversally, $\mathcal{O}(n)$ work is needed to insert the new edges into the boundary of the intersecting regions. Thus in the worst case, the algorithm takes $\mathcal{O}(nh)$ time.

Figures B-1 and B-2 demonstrate the algorithm for a complicated multiply connected polygon.

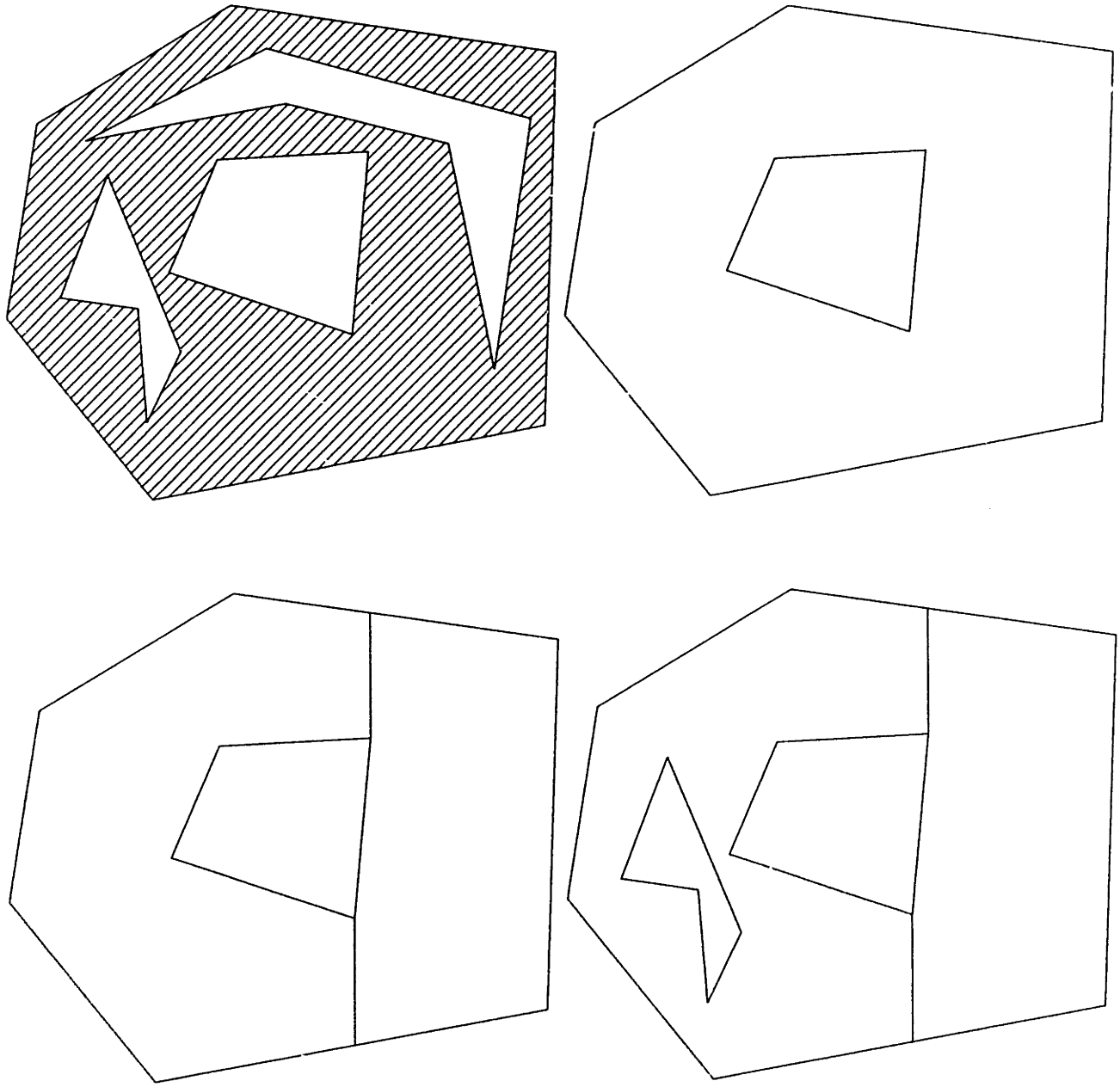

Figure B-1: Subdividing a multiply connected polygonal region incrementally.

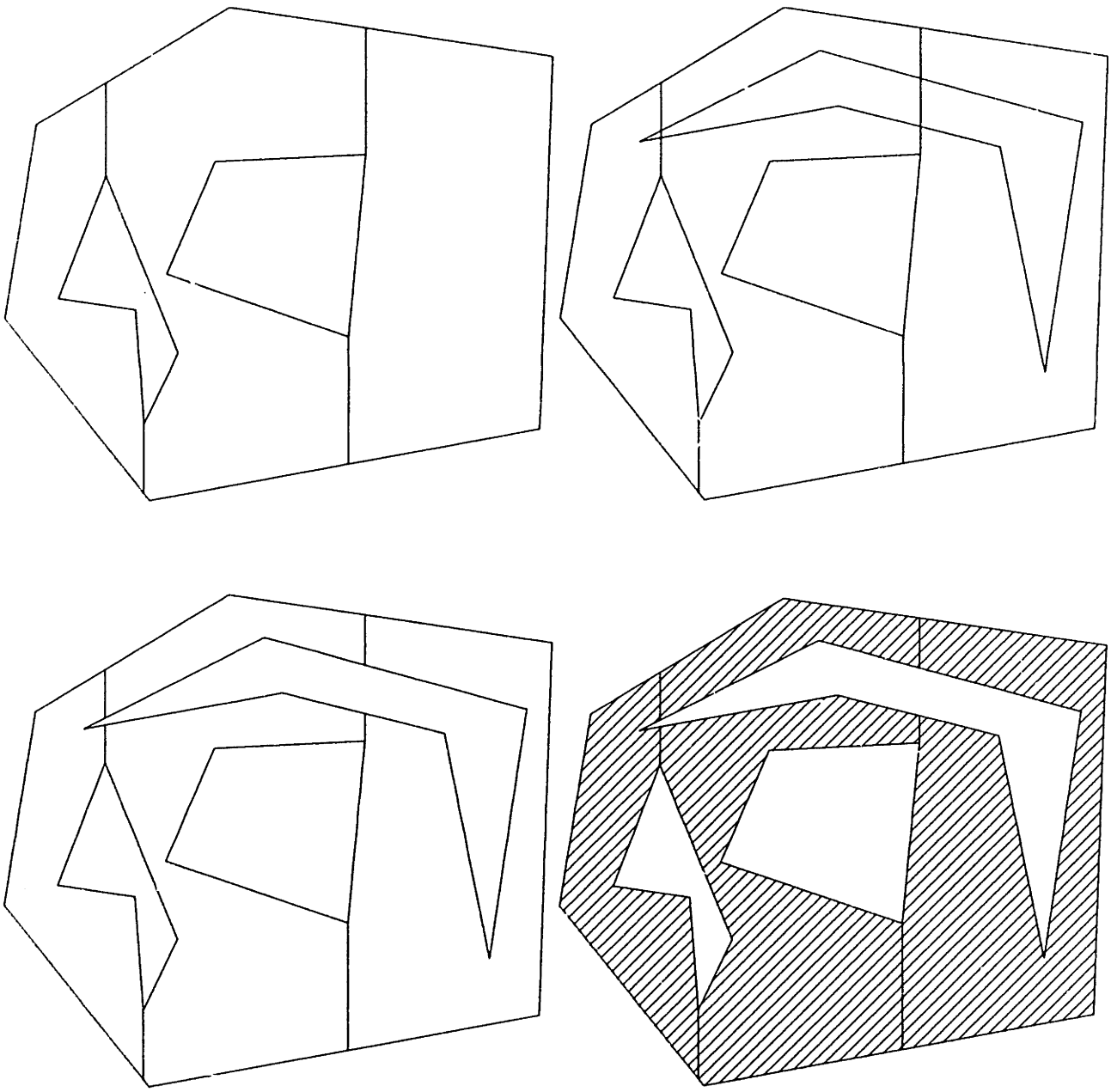

Figure B-2: Subdividing a multiply connected polygonal region incrementally (continued).

Bibliography

[1] C. G. Armstrong, T. K. H. Tam, D. J. Robinson, R. M. McKeag, and M. A. Price. Automatic generation of well structured meshes using medial axis and surface subdivision. In G. A. Gabriele, editor, *Proceedings of the 17th ASME Design Automation Conference: Advances in Design Automation, Vol. 2*, pages 139–146, Miami, FL, September 1991. New York: ASME.

[2] F. Aurenhammer. Voronoi diagrams — a survey of fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.

[3] L. Bardis and N. M. Patrikalakis. Topological structures for generalized boundary representations. Technical Report MITSG 94-22, Cambridge, MA: MIT Sea Grant College Program, September 1994.

[4] C. Bliek. *Computer Methods for Design Automation.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, July 1992.

[5] H. Blum. Biological shape and visual science (part I). *Journal of Theoretical Biology*, 38:205–287, 1973.

[6] H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, pages 362–381, ed: Weinant Wathen-Dunn MIT Press, 1967.

[7] H. Blum and R. N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180, 1978.

[8] W. Boehm. Inserting new knots into B-spline curves. *Computer Aided Design*, 12(4):199–201, 1980.

[9] F. L. Bookstein. The line skeleton. *Computer Graphics and Image Processing*, 11:123–137, 1979.

[10] J. W. Brandt. *Theory and Application of the Skeleton Representation of Continuous Shapes.* PhD thesis, University of California, Davis, CA, December 1991.

[11] J. W. Brandt. Describing a solid with the three-dimensional skeleton. In J. D. Warren, editor, *Proceedings of The International Society for Optical Engineering, Volume 1830, Curves and Surfaces in Computer Vision and Graphics III*, pages 258–269. SPIE, Boston, Massachusetts, 1992.

[12] J. W. Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. *CVGIP: Image Understanding*, 59(1):116–124, January 1994.

[13] J. W. Brandt and V. R. Algazi. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55(3):329–338, May 1992.

[14] J. W. Brandt and V. R. Algazi. Lossy encoding of document images with the continuous skeleton. In P. Maragos, editor, *Visual Communications and Image Processing '92, SPIE 1818*, pages 663–673, 1992.

[15] J. W. Brandt, A. K. Jain, and V. R. Algazi. Medial axis representation and encoding of scanned documents. *Journal of Visual Communication and Image Representation*, 2(2):151–165, June 1991.

[16] E. Brisson. Private Communication, 1995.

[17] B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In N. K. Bose, editor, *Multidimensional Systems Theory: Progress, Directions and Open Problems in Multidimensional Systems*, pages 184–232, 1985. Dordrecht, Holland: D. Reidel Publishing Company.

[18] J. Canny. Generalized characteristic polynomials. *Journal of Symbolic Computation*, 9:241–250, 1990.

[19] C.-S. Chiang. *The Euclidean Distance Transform.* PhD thesis, Purdue University, West Lafayette, IN, August 1992.

[20] J. J. Chou. Voronoi diagrams for planar shapes. *IEEE Computer Graphics*, 15(2):52–68, March 1995.

[21] E. Cohen, T. Lyche, and R. F. Riesenfeld. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer Graphics and Image Processing*, 14:87–111, 1980.

[22] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, MA, 1990.

[23] G. Dahlquist and A. Björck. *Numerical Methods.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.

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

[25] T. Dokken. Finding intersections of B-spline represented geometries using recursive subdivision techniques. *Computer Aided Geometric Design*, 2:189–195, 1985.

[26] D. Dutta and C. M. Hoffmann. A geometric investigation of the skeleton of CSG objects. In B. Ravani, editor, *Proceedings of the 16th ASME Design Automation Conference: Advances in Design Automation, Computer Aided and Computational Design*, volume I, pages 67–75, Chicago, IL, September 1990. New York: ASME, 1990.

[27] D. Dutta and C. M. Hoffmann. On the skeleton of simple CSG objects. *Journal of Mechanical Design, ASME Transactions*, 115(1):87–94, March 1993.

[28] R. T. Farouki and J. K. Johnstone. The bisector of a point and a plane parametric curve. *Computer Aided Geometric Design*, 11(2):117–151, April 1994.

[29] I. D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis Horwood, Chichester, England, 1979.

[30] S. Fortune. Voronoi diagrams and Delaunay triangulations. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, pages 193–233. World Scientific, Singapore, 1992.

[31] C. B. Garcia and W. I. Zangwill. Global continuation methods for finding all solutions to polynomial systems of equations in n variables. In A. V. Fiacco and K. O. Kortanek,

editors, *Extremal Methods and Systems Analysis*, pages 481–497. Springer-Verlag, New York, NY, 1980.

[32] S. M. Gelston and D. Dutta. Boundary surface recovery from skeleton curves and surfaces. *Computer Aided Geometric Design*, 12(1):27–51, 1995.

[33] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 1989.

[34] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, April 1985.

[35] H. N. Gursoy. *Shape Interrogation by Medial Axis Transform for Automated Analysis.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, November 1989.

[36] H. N. Gursoy and N. M. Patrikalakis. Automated interrogation and adaptive subdivision of shape using medial axis transform. *Advances in Engineering Software and Workstations*, 13(5/6):287–302, September/November 1991.

[37] H. N. Gursoy and N. M. Patrikalakis. An automated coarse and fine surface mesh generation scheme based on medial axis transform, part I: Algorithms. *Engineering with Computers*, 8(3):121–137, 1992.

[38] H. N. Gursoy and N. M. Patrikalakis. An automated coarse and fine surface mesh generation scheme based on medial axis transform, part II: Implementation. *Engineering with Computers*, 8(4):179–196, 1992.

[39] M. Held. *On the Computational Geometry of Pocket Machining.* Springer-Verlag, Berlin, Germany, 1991.

[40] C. M. Hoffman and G. Vanecek. On alternate solid representations and their uses. Technical Report CSD-TR-91-019, Computer Sciences Department, Purdue University, March 1991.

[41] C. M. Hoffmann. Computer vision, descriptive geometry, and classical mechanics. In B. Falcidieno and I. Herman, editors, *Proceedings of the Eurographics Workshop, Computer Graphics and Mathematics, October 1991, Genoa, Italy*, pages 229–244. Springer-

Verlag, October 1991. Also available as Technical Report CSD-TR-91-073, Computer Sciences Department, Purdue University.

[42] C. M. Hoffmann. How to construct the skeleton of CSG objects. In A. Bowyer and J. Davenport, editors, *Proceedings of the Fourth IMA Conference, The Mathematics of Surfaces, University of Bath, UK, September 1990*, pages 421–438, New York, 1994. Oxford University Press.

[43] C.-Y. Hu. *Robust Algorithms for Sculptured Shape Visualization.* Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, July 1993.

[44] R. B. Kearfott. Decomposition of arithmetic expressions to improve the behavior of interval iteration for nonlinear systems. *Computing*, 47:169–191, 1991.

[45] G. A. Kriezis, P. V. Prakash, and N. M. Patrikalakis. A method for intersecting algebraic surfaces with rational polynomial patches. *Computer Aided Design*, 22(10):645–654, December 1990.

[46] J. M. Lane and R. F. Riesenfeld. Bounds on a polynomial. *BIT: Nordisk Tidskrift for Informations-Behandling*, 21(1):112–117, 1981.

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

[48] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4):363–369, July 1982.

[49] N. Levinson and R. M. Redheffer. *Complex Variables.* Holden-Day, Inc., Oakland, CA, 1970.

[50] T. Maekawa and N. M. Patrikalakis. Computation of singularities and intersections of offsets of planar curves. *Computer Aided Geometric Design*, 10(5):407–429, October 1993.

[51] T. Maekawa and N. M. Patrikalakis. Interrogation of differential geometry properties for design and manufacture. *The Visual Computer*, 10(4):216–237, March 1994.

[52] D. Manocha. Solving systems of polynomial equations. *IEEE Computer Graphics and Applications*, 14(2):46–55, March 1994.

[53] U. Montanari. Continuous skeletons from digitized images. *Journal of the Association for Computing Machinery*, 16(4):534–549, October 1969.

[54] J. R. Munkres. *Topology: a First Course.* Prentice-Hall, Englewood Cliffs, NJ, 1975.

[55] L. R. Nackman. Curvature relations in three-dimensional symmetric axes. *Computer Graphics and Image Processing*, 20:43–57, 1982.

[56] L. R. Nackman and S. M. Pizer. Three-dimensional shape description using the symmetric axis transform I: Theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):187–202, March 1985.

[57] A. Neumaier. *Interval Methods for Systems of Equations.* Cambridge University Press, Cambridge, 1990.

[58] T. Nishita, T. W. Sederberg, and M. Kakimoto. Ray tracing trimmed rational surface patches. *ACM Computer Graphics*, 24(4):337–345, August 1990.

[59] N. M. Patrikalakis. Shape interrogation. In C. Chryssostomidis, editor, *Proceedings of the 16th Annual MIT Sea Grant College Program Lecture and Seminar, Automation in the Design and Manufacture of Large Marine Systems*, pages 83–104, Cambridge, MA, October 1988. New York: Hemisphere Publishing, 1990.

[60] N. M. Patrikalakis, W. Cho, C.-Y. Hu, T. Maekawa, E. C. Sherbrooke, and J. Zhou. Towards robust geometric modelers, 1994 progress report. In *Proceedings of the 1995 NSF Design and Manufacturing Grantees Conference, University of California, San Diego, California*, pages 139–140. Society of Manufacturing Engineers, Dearborn, Michigan, January 1995.

[61] N. M. Patrikalakis and H. N. Gursoy. Shape interrogation by medial axis transform. In B. Ravani, editor, *Proceedings of the 16th ASME Design Automation Conference: Advances in Design Automation, Computer Aided and Computational Design, Vol. I*, pages 77–88, Chicago, IL, September 1990. New York: ASME.

[62] N. M. Patrikalakis, C.-Y. Hu, T. Maekawa, and E. C. Sherbrooke. Towards robust geometric modellers. In *Proceedings of the 1994 NSF Design and Manufacturing Grantees Conference, Cambridge, Massachusetts*, pages 199–200. National Science Foundation, Society of Manufacturing Engineers, Dearborn, Michigan, January 1994.

[63] N. M. Patrikalakis, T. Maekawa, E. C. Sherbrooke, and J. Zhou. Computation of singularities for engineering design. In T. L. Kunii and Y. Shinagawa, editors, *Modern Geometric Computing for Visualization*, pages 167–191. Tokyo: Springer-Verlag, June 1992.

[64] N. M. Patrikalakis and P. V. Prakash. Surface intersections for geometric modeling. *Journal of Mechanical Design, ASME Transactions*, 112(1):100–107, March 1990.

[65] F. P. Preparata. The medial axis of a simple polygon. In G. Goos and J. Hartmanis, editors, *Lecture Notes in Computer Science: Mathematical Foundations of Computer Science*, pages 443–450. Springer-Verlag, 1977.

[66] M. A. Price, C. G. Armstrong, and M. A. Sabin. Hexahedral mesh generation by medial surface subdivision: I. Solids with convex edges. Submitted to *International Journal of Numerical Methods in Engineering*. Received November 1994.

[67] J. M. Reddy and G. Turkiyyah. Computation of 3d skeletons by a generalized Delaunay triangulation technique. *Computer Aided Design*. Received October 1994. To appear.

[68] A. Rosenfeld. Axial representations of shape. *Computer Vision, Graphics and Image Processing*, 33:156–173, 1986.

[69] J. R. Rossignac and A. G. Requicha. Offsetting operations in solid modelling. *Computer Aided Geometric Design*, 3(2):129–148, 1986.

[70] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 1976. 3rd edition.

[71] G. L. Scott, S. C. Turner, and A. Zisserman. Using a mixed wave/diffusion process to elicit the symmetry set. *Image and Vision Computing*, 7:63–70, 1989.

[72] T. W. Sederberg. Algorithm for algebraic curve intersection. *Computer Aided Design*, 21(9):547–554, November 1989.

[73] T. W. Sederberg and T. Nishita. Geometric Hermite approximation of surface patch intersection curves. *Computer Aided Geometric Design*, 8:97–114, 1991.

[74] D. J. Sheehy, C. G. Armstrong, and D. J. Robinson. Numerical computation of medial surface vertices. In *IMA Conference on Mathematics of Surface VI, Brunel University, 5-7 September 1994*. IMA, 1994. Received February 1995. To appear.

[75] D. J. Sheehy, C. G. Armstrong, and D. J. Robinson. Computing the medial surface of a solid from a domain Delaunay triangulation. In J. R. Rossignac and C. M. Hoffmann, editors, *Proceedings of the Third ACM Solid Modeling Conference, May 1995, Salt Lake City, Utah*, New York, 1995. ACM. Received February 1995. To appear.

[76] E. C. Sherbrooke. *Computation of the Solutions of Nonlinear Polynomial Systems.* Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, October 1993.

[77] E. C. Sherbrooke and N. M. Patrikalakis. Computation of the solutions of nonlinear polynomial systems. *Computer Aided Geometric Design*, 10(5):379–405, October 1993.

[78] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. Computation of medial axis transforms of 3d polyhedra. In J. R. Rossignac and C. M. Hoffmann, editors, *Proceedings of the Third ACM Solid Modeling Conference, May 1995, Salt Lake City, Utah*, New York, May 1995. ACM. To appear.

[79] V. Srinivasan and L. R. Nackman. Voronoi diagram for multiply connect polygonal domains, I: Algorithm. *IBM Journal of Research and Development*, 31(3):361–372, May 1987.

[80] V. Srinivasan, L. R. Nackman, J.-M. Tang, and S. N. Meshkat. Automatic mesh generation using the symmetric axis transformation of polygonal domains. *Proceedings of the IEEE, Special Issue on Computational Geometry*, 80(9):1485–1501, 1992.

[81] A. Sudhalkar, L. Gürsöz, and F. Prinz. Continuous skeletons of discrete objects. In J. Rossignac, J. Turner, and G. Allen, editors, *Proceedings of the Second Symposium on Solid Modeling and Applications, Montreal, Canada*, pages 85–94, New York, 1993. ACM.

[82] K. Sugihara. Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 55(6):522–531, November 1993.

[83] T. K. H. Tam and C. G. Armstrong. 2d finite element mesh generation by medial axis subdivision. *Advances in Engineering Software and Workstations*, 13(5/6):313–324, September/November 1991.

[84] M. E. Vafiadou and N. M. Patrikalakis. Interrogation of offsets of polynomial surface patches. In F. H. Post and W. Barth, editors, *Eurographics '91, Proceedings of the 12th Annual European Association for Computer Graphics Conference and Exhibition*, pages 247–259 and 538, Vienna, Austria, September 1991. Amsterdam: North-Holland.

[85] P. J. Vermeer. *Medial Axis Transform to Boundary Representation Conversion.* PhD thesis, Purdue University, May 1994.

[86] F. W. Warner. *Foundations of Differentiable Manifolds and Lie Groups.* Springer, New York, 1983.

[87] F.-E. Wolter. *Cut Loci in Bordered and Unbordered Riemannian Manifolds.* PhD thesis, Technical University of Berlin, Department of Mathematics, December 1985.

[88] F.-E. Wolter. Cut locus and medial axis in global shape interrogation and representation. *Computer Aided Geometric Design*, 1992. To appear. Also available as MIT Ocean Engineering Design Laboratory Memorandum 92-2, January 1992.

[89] W. I. Zangwill and C. B. Garcia. *Pathways to solutions, fixed points, and equilibria.* Prentice-Hall, Englewood Cliffs, NJ, 1981.

[90] J. Zhou, E. C. Sherbrooke, and N. M. Patrikalakis. Computation of stationary points of distance functions. *Engineering with Computers*, 9(4):231–246, Winter 1993.