81-016

THREE-DIMENSIONAL SHAPE DESCRIPTION USING THE SYMMETRIC AXIS TRANSFORM

by

Lee Richard Nackman

THREE-DIMENSIONAL SHAPE DESCRIPTION USING THE SYMMETRIC AXIS TRANSFORM

by

Lee Richard Nackman

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill

1982

Approved by:

Stephen M. Pizer

Adviser: Stephen M. Pizer

Frederick P. Brooks Jr

Reader: Frederick P. Brooks, Jr.

Reader: Thomas H. Brylawski

LEE RICHARD NACKMAN.
Three-Dimensional Shape Description Using the Symmetric Axis Transform
(Under the direction of STEPHEN M. PIZER.)

Abstract

Blum's transform, variously known as the *symmetric axis transform*, *medial axis transform*, or *skeleton*, and his associated two-dimensional shape description methodology are generalized to three-dimensions. Bookstein's two-dimensional algorithm for finding an approximation to the symmetric axis is also generalized.

The symmetric axis (SA) of an object with a smooth boundary is the locus of points inside the object having at least two nearest neighbors on the object boundary. In three dimensions, the SA is, in general, a collection of smooth surface patches, called simplified segments, connected together in a tree-like structure. Together with the radius function, the distance from each point on the SA to a nearest boundary point, the SA forms the symmetric axis transform. The three-dimensional symmetric axis transform defines a unique, coordinate-system-independent decomposition of an object into disjoint, two-sided pieces, each with its own simplified segment and associated object boundary patches.

Four principal contributions are presented. (1) A relationship among the Gaussian and mean curvatures of a simplified segment, the Gaussian and mean curvatures of the associated object boundary patches, and radius function measures is derived. (2) A further decomposition is proposed wherein each two-sided piece is partitioned into primitives drawn from three separate, but not completely independent, primitive sets: width primitives, boundary primitives, and axis primitives. Width primitives are regions derived from derivatives of the radius function; hence, they capture the behavior of the boundary patches with respect to the simplified segment. Axis and boundary primitives are regions of constant signs of Gaussian and mean curvatures of the simplified segment and boundary patches respectively. The aforementioned curvature relationship is used to derive relationships among the primitive sets. (3) In the course of studying width primitives, it is proved that, under certain non-degeneracy assumptions, the regions of the graph defined by the critical points, ridge lines, and course lines of a scalar valued function over a surface have one of three types of cycle as boundary. (4) An almost linear algorithm that takes a polyhedral approximation to a three-dimensional object and yields a polyhedral surface approximation to that object's SA is developed.

To My Family

ACKNOWLEDGEMENTS

It is a great pleasure to thank the many people who have helped me with this work.

My adviser, Steve Pizer, encouraged this work, served as the reader's advocate, and often responded to my ideas with probing questions. Frederick P. Brooks, Jr., Thomas Brylawski, Henry Fuchs, and Gyula Magó served on my committee. Frederick P. Brooks, Jr. and Thomas Brylawski also each read the dissertation and made useful comments and suggestions. Robert Gardner read an early draft of Chapter 3.

During the formative stage of this work, I was supported by a Graduate Fellowship from IBM Corporation. The final stage has been supported by my wife.

The greatest joy of being a graduate student is the opportunity to exchange ideas and camaraderie with fellow students. Dave Tolle and Mike Pique were always ready to talk and to listen. Their knowledge, enthusiasm, curiosity, and friendship were a great help. Dave read drafts of several chapters. Mike and I shared many hours working on one piece of software or another; I learned much from him. I have also benefited from and enjoyed technical discussions with Steve Bellovin, Geoff Frank, Charles Gunn, Bruce Smith, Tom Williams, and John Zimmerman. I also thank Vicki Baker, Judy Popelas, Hollins Williams, Teresa Thomas, John Zimmerman, and many other students, especially those of West House, for their friendship and encouragement.

My greatest debt is to my family. Without their constant support this work would not have been completed. My parents, sisters, and late paternal grandmother were a source of seemingly unlimited love, understanding, and encouragement, sharing both excitement and frustration. My grandmother often said that she didn't understand my work, but somehow, perhaps through the wisdom of her years, she usually understood me. I shall miss that immensely. Ava, my wife, provided more emotional support and love than I ever thought possible. Even when her own schooling demanded much of her time and energy, she endured my often strange hours and habits, humored me when I needed it, encouraged me when I was frustrated, and, most important, shared her thoughts and dreams with me.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

"The study of form may be descriptive merely, or it may become analytical. We begin by describing the shape of an object in the simple words of common speech: we end by defining it in the precise language of mathematics; and the one method tends to follow the other in strict scientific order.... The mathematical definition of a 'form' has a quality of precision which was quite lacking in our earlier stage of mere description; it is expressed in few words or in still briefer symbols, and these words or symbols are so pregnant with meaning that thought itself is economised...."

"Nor must we forget that the biologist is much more exacting in his requirements, as regards to form, than the physicist; for the latter is usually content with either an ideal or a general description of form, while the student of living things must needs be specific."

—–D'Arcy Thompson, [Thompson42a, p. 1026 and 1030]

1.1. Background

Rapid advances in data acquisition techniques, especially computed tomography, challenge us to seek effective shape description techniques with which to attack problems in biological and medical shape measurement in three dimensions. Such problems appear in many guises.

For many years biologists have sought quantitative methods for studying biological shape and shape change in order to examine the relation between form and function[Alexander71a], to study growth (both ontogenetic and phylogenetic)[Thompson42a, le Gros Clark45a, Sprent72a], and for taxonomic classification[Hursh76a]. In his recent book[Bookstein78a] Bookstein argues persuasively that shape measurements computed solely from landmarks, points of either anatomical or geometric significance, are inherently inadequate to the task. Structural shape descriptors offer a potential source of new, more useful measures.

Already, Webber and Blum[Webber79a] have used properties derived from a structural descriptor (the symmetric axis transform) to quantify shape properties of lateral projections of human mandibles. Turner-Smith and colleagues[Turner-Smith80a] are working toward a better understanding of the progression of spinal curvature and rotation in patients with idiopathic scoliosis. They believe that the progression of the disease can be monitored from back surface shapes and are actively seeking techniques for describing such shapes[Turner-Smith81a]. Shape description may also be a useful tool for planning and assessing reconstructive facial surgery[Todd-Pokropek81a] and for reliably predicting the result of orthodontic procedures[Bookstein78a].

In hospitals around the world, huge numbers of computed tomography (CT) studies, each a sequence of images of cross-sectional slices of the human body[Brooks75a], are now being performed and archived. Each slice is a two-dimensional map of the distribution of the X-ray attenuation coefficients of the tissue "cut" by the slice. Together the slices reveal the morphology of the organs contained within. Systematic study of organ shapes, both normal and pathologic, across the large populations contained in the archived studies could be expected to yield results of scientific and clinical value. Perhaps we could then speak quantitatively of normal and abnormal ranges of organ shape, much as we are beginning to be able to speak quantitatively of normal and abnormal distributions of attenuation values[Pullan78a]. Unfortunately, neither suitable shape analysis nor automatic organ extraction techniques[1] yet exist. The potential applicability of structural shape descriptors to the former problem is clear.

[1] In the limited cases where thresholding is able to partition the picture elements of each slice of the study into two classes, organ and non-organ, a three-dimensional boundary following algorithm (e.g. [Artzy80a]) may be able to extract an organ automatically.

It may also be possible to use structural shape descriptors as a source of *a priori* anatomical information for computer-assisted analysis of medical images. In automatic procedures, organ shape models might guide organ extraction and aid in constructing three-dimensional displays from two-dimensional slices. In interactive procedures they might allow the diagnostician to interact with the machine at the structure level. For example, a diagnostician working with a three-dimensional display of a CT study might be able to request that an organ be removed from view to see behind it, to request dose calculations for individual organs in radiation treatment-planning applications, and to request shape measures to be compared with population norms.

Experimental systems for small subsets of several of these applications have been built[Ballard78a, Sunguroff78a, Soroka79a, Shani80a]. Their capabilities, hampered as they are by inadequate techniques for measuring and describing three-dimensional shape, suggest that the potential payoff of successful use of structural shape descriptors is likely to be large.

1.2. Rationale

This dissertation sets forth the early development of an attractive, though yet untested, three-dimensional structural shape description technique. Why study three-dimensional shape description when two-dimensional shape description is not yet well understood? Originally, exciting medical applications, made possible by advancing technology, motivated our work. While that motivation still exists, more compelling fundamental reasons have come to light.

The objects we study are three-dimensional or, if one considers shape change as well as shape itself, four-dimensional. Therefore, to execute a two-dimensional shape analysis, one must, before the analysis is even begun, choose a mapping from the three- or four-dimensional space in which the object is imbedded to the two-dimensional space of the analysis. Of course, in some cases the choice of the mapping is obviated, either by a symmetry of the object under study, by limitations of the data acquisition apparatus, or by the goals of the analysis. More often, one either makes an arbitrary choice, or one orients the object along some "standard" axis or plane defined by points of anatomical significance. The former is subjective; the latter is arbitrary and subject to error propagation, for, as the simple exercise of slicing a cone with various planes illustrates, slight error in orientation can lead to large changes in the resulting two-dimensional object. At best, a two-dimensional analysis of a three- or four-dimensional object is incomplete. At worst, it is biased and misleading.

Studying a shape description technique in three dimensions helps to distinguish between properties that are mere coincidences of the technique's formulation in the plane and properties that are more fundamental. This is important, for simplicity and elegance demand that a two-dimensional technique be a special case of a corresponding three-dimensional technique. Yet the usual course of development is, often by necessity, the converse: one begins with a promising two-dimensional technique, then seeks an appropriate higher dimensional generalization.

Consider, for example, an analogous situation in geometry. The formula for the sum of the interior angles of a convex planar n-gon ($180n$-360) was known to Euclid. Generalizing the formula to three dimensions first required a suitable generalization of convex planar n-gons. Obviously, convex polyhedra are the three-dimensional analogs of convex polygons. But, what of the "n"? It could generalize to the number of faces, to the number of edges, to the number of vertices, or perhaps to some combination of the three. Similarly, what concept replaces that of an "interior angle"?

The generalization itself, discovered only in 1874, is not germane here[2]. It

[2] The generalization is called Gram's relation for angle-sums and holds for all d-dimensional convex polytopes. See Sections 14.1 and 14.4 of [Grunbaum67a].

is, however, worth noting that the generalized formula depends not on the number of vertices, but instead on the number of faces. Indeed, in the *d*-dimensional case the formula depends on the number of (*d*-1)-dimensional "faces". This is true, as well, in the planar case, but is masked by the coincidence that in the plane the number of edges is necessarily identical to the number of vertices. The example, then, illustrates in small part what mathematicians have long known: generalizing to higher dimensions often illuminates lower dimensional cases as well.

I have presented a rationale for studying shape description in dimensions three or more. In this dissertation, I limit discussion to three dimensions because a full-blown treatment of a shape description technique in *n* dimensions requires mathematical sophistication beyond mine. Though this reason alone does not justify restricting this work to three dimensions, when combined with the natural division between three and four dimensions, between shape and shape change, it is compelling.

In the next section I define what I mean by shape and shape description and describe how they are related to (classical) statistical pattern recognition. Those notions established, I then sketch several shape description techniques to illustrate important shape description paradigms. Chapter 1 then concludes with an overview of the research described herein.

1.3. Shape and Shape Description

We begin with two-dimensional shape. Let an *outline* in the Euclidean plane be a regular, simple, closed plane curve. In other words, an outline is a closed plane curve with no self-intersections and with a well-defined, continuously turning tangent at all points. A *figure* is an outline together with its interior. The scope of our discussion of shape is limited to single outlines, figures, and their three-dimensional generalizations[3]. We therefore do not consider disconnected objects, objects with holes or corners, or point sets better characterized by notions of texture.

A *shape* is an equivalence class of figures (or outlines). Choosing the equivalence relation is both difficult and important – it determines the intuitive meaning of "shape". At one extreme, we might require all figures in a shape to be congruent. Usually, such a relation is too stringent to be useful. At the other extreme, we might use a single measure, such as the ratio of the square of the outline perimeter to the figure area, to determine which shape an outline is in. Unless the application domain is highly constrained, such a relation is too broad to be useful.

A *shape specification* is a finite specification of the members of a shape. For example, in statistical pattern recognition, a feature vector is constructed for each figure (pattern) to be classified. Assuming that the feature vectors are elements of some appropriately chosen metric space, a shape is specified by the region of the metric space that contains the feature vectors for precisely those figures in the shape. Such a region, then, is a shape specification[4]. Each element of a feature vector is a *shape measurement*, a mapping from the set of all outlines to the real numbers. Finding appropriate feature vectors, metrics, and regions are difficult problems.

Indeed, the statistical pattern recognition literature devotes considerable attention to systematic techniques of determining appropriate metrics and regions given statistical information about the feature vectors (see e.g.[Meisel72a, Duda73a]). On the other hand, finding low-dimensional feature vectors has remained an *ad hoc* and difficult art. Another pattern recognition

[3] Outlines and figures in three dimensions are defined in Chapter 3. Intuitively, a three-dimensional outline is a smooth surface that can be "stretched" into a sphere.

[4] This is true only if a finite representation of the region exists.

approach, often called structural pattern recognition, has evolved in response to the difficulty of finding effective feature vectors for use in statistical pattern recognition.

In structural pattern recognition, the primary goal is description rather than classification. Though it is difficult to formulate a precise definition of a description, two intuitive notions characterize most descriptions: (1) a description is in a form more suitable for further processing than (a representation of) the figure itself, and (2) a description captures the "essence" of a figure relative to some context[Evans69a, Pavlidis77a]. The distinction between statistical and structural pattern recognition is not sharp, but is mainly a difference in goals and approach. The shape measurements used in statistical pattern recognition tend to capture global figure properties such as width, elongation, and compactness, while structural descriptions capture relationships among the sub-figures that comprise a figure. Structural descriptions are often labeled graphs with nodes representing sub-figures, arcs representing relationships among sub-figures, and node labels characterizing sub-figures.

Ideally, all members of a shape have the same description which then serves directly as a shape specification. Since this is rarely the case in practice, systematic techniques are being developed for approximate matching of structural descriptions[Haralick78a, Shapiro80a, Shapiro81a]. Once such techniques are in hand, structural descriptions can be used for classification by matching a description of the figure to be classified against a prototype figure description for each class (shape). Structural descriptions can also be used as a source of shape measurements to be used to construct feature vectors for statistical pattern recognition.

1.4. Shape Description Paradigms

In his oft-quoted essay, Kuhn[Kuhn70a] likens "normal science" to solving puzzles within the restricted framework of paradigms, abstractions of the common lessons learned solving problems that appear different, but that are, upon closer examination, identical in essential aspects[5]. Indeed, most shape description techniques are elaborations of one of three paradigms[6]: *represent, then discard*; *decomposition*; and *prototypes*. I present each, together with two- and three-dimensional examples, below.

1.4.1. Represent, then Discard

> "...we must learn from the mathematician to eliminate and to discard; to keep the type in mind and leave the single case, with all its accidents, alone; and to find in this sacrifice of what matters little and conservation of what matters much one of the peculiar excellences of the method of mathematics."
>
> ——D'Arcy Thompson, [Thompson42a, p. 1032]

In the first paradigm, which I call *represent, then discard*, a shape description (or measurement) is constructed in two steps:

(1) Find a discrete representation of the outline or figure.

(2) Discard "irrelevant" information contained in the representation.

For example, a two-dimensional outline can be represented by a Fourier series by choosing a parameterization of the outline that is periodic. The first few coefficients of the Fourier series contain information about overall outline properties and have been used as shape measurements[Granlund72a, Zahn72a, Persoon77a]. Walsh function expansions have also been used in like manner[Searle70a].

[5] See, especially, part 3 of the postscript to the 2nd edition.

[6] Pavlidis[Pavlidis78a, Pavlidis80a] classifies shape description techniques into but two categories: *information preserving* and *information non-preserving*.

A similar technique has been applied to star-shaped[7] figures in three-dimensions[Schudy79a, Brown79a]. Since the figure is star-shaped, there is a polar coordinate system in which the outline can be expressed as a single-valued function over the unit sphere. Using Laplace's spherical harmonic functions[8], which form an orthonormal, complete system of functions over the unit sphere, the outline can be represented by a series expansion. Schudy[Schudy79a] has developed an optimization technique that finds the series expansion coefficients of the outline that best fits canine heart wall contours from ultrasound data. Since the beating heart, and hence the coefficients themselves, are (near) periodic, each coefficient time series can be approximated by a Fourier series expansion, yielding a series expansion representation in four-dimensions.

Two- and three-dimensional figures can also be represented by an infinite sequence of moments[Alt62a]. Though the moments themselves are very sensitive to size, location, and orientation, it is possible to derive *moment invariants*, algebraic combinations of low order moments that are invariant under changes in size, location, and orientation[Hu62a, Sadjadi80a].

Other measures, often single numbers, capture only particular aspects of a figure that are important in a specific application. Generally, such measures summarize aspects of the entire outline or figure; hence, they have been called *gestalt-variables*[Attneave56a]. Examples include measures of compactness (e.g. the ratio of the square of the outline perimeter to the figure area), symmetry, and elongation. Gestalt-variables have been used widely in psychophysics, where investigators seek an understanding of the relation between the physical characteristics of a stimulus and the subject's response to it. Brown and Owen[Brown67a], in their critical review of the psychophysics literature, identi-

[7] A set is star-shaped if there is some point in the set to which all other points can be connected by a line segment contained completely within the set.

[8] See e.g. Chapter 7, Section 5 of [Courant53a].

fied over 100 shape measures that had been used as stimulus dimensions. Similar measures have also been used in geography[Clark73a, Bosch78a] and in computerized picture processing[Rosenfeld76a].

Though gestalt-variables are easy to compute from many discrete outline representations, too much information is discarded for them to be useful *except* in highly constrained situations. On the other hand, the *amount* of information discarded from series expansion representations is determined by the number of coefficients kept and the convergence speed of the series. The former is easily controlled. Unfortunately, since local perturbations of the outline are reflected in all coefficients, it is difficult to control the *type* of information discarded. This is one of two major disadvantages of all shape descriptors derived from series expansion representations of the outline. The other disadvantage is inherent in all techniques that treat the points of an outline as points of the outline alone, rather than as points of the outline and the space containing the outline. I elaborate below.

1.4.2. Decomposition

The *decomposition* paradigm is, in part, a response to the problem illustrated in Figure 1.1: points near to each other in the plane may be far apart when distance is measured within the outline. Decomposition techniques are various embodiments of the divide-and-conquer strategy often used in algorithm design: divide the figure into sub-figures, called *primitives*, describe each primitive, then combine the results to yield a single description. Many examples of this paradigm have appeared in the computer shape description literature.

For example, in two dimensions, polygonally bounded figures have been decomposed into possibly overlapping convex subsets[Pavlidis68a, Pavlidis72a, Pavlidis77a]. As the polygon is traversed clockwise, for each edge of the polygon a so-called basic half-plane is defined by the right-hand side of the line

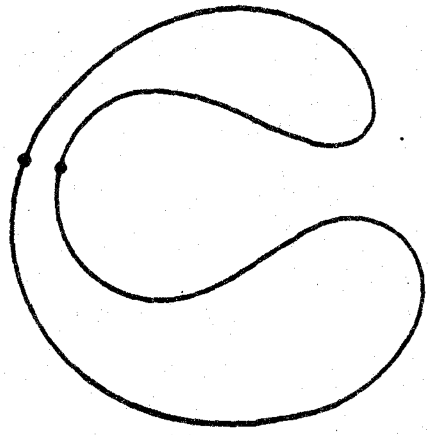


Figure 1.1.: Motivation for the Decomposition Paradigm (after [Pavlidis77a])

containing the edge. Beginning with the intersection of all basic half-planes, non-decreasing sequences of convex sets can be formed by taking the intersection of successively fewer basic half-planes. Some set in each sequence must be maximal, in the sense that all subsequent members of the sequence are not subsets of the figure. Each such maximal set is called a *primary convex subset* (PCS). Together, the PCS's cover the figure. Besides being expensive to compute, the PCS decomposition results in non-disjoint primitives which can be radically altered by small changes in the polygon. In addition, the convexity requirement causes thin curved figures to be decomposed into many small pieces.

Others have relaxed the convexity requirement to obtain "more natural" decompositions. Feng and Pavlidis[Feng75a] decompose polygonally bounded figures into convex subsets and non-convex polygonal "spirals". Shapiro and Haralick[Shapiro79a] define a visibility relation on pairs of line segments in the polygon. Two segments are related if they are mutually visible, if any line from an endpoint of one segment to an endpoint of the other segment is completely contained in the figure. Primitive regions are determined by executing a graph clustering algorithm on the graph of the visibility relation. The graph clustering approach has been generalized by Bjorklund and Pavlidis[Bjorklund81a] to include multiple relations between segments.

Existing decomposition schemes have several disadvantages. For most decomposition schemes, there is a limited domain of figures for which the scheme yields an intuitively pleasing decomposition that captures important figure properties. Perhaps more important, the primitives resulting from the decomposition are often poorly constrained, and thus not much simpler to describe than the original figure itself. Finally, the time complexity of many decomposition algorithms grows as the square or cube of the number of elements (pixels, line segments) in the discrete outline approximation. However, as we shall see in Chapter 5, this need not be the case.

Three-dimensional curved figures have been decomposed into primitives called *generalized cones*[Nevatia77a, Agin76a]. A generalized cone is defined by a space curve, called the *axis*, and planar *cross-sections* normal to the axis. The generalized cone is the volume swept out by moving cross-sections of arbitrary shape and size along the axis. Various heuristic techniques have been developed to decompose three-dimensional figures into collections of restricted classes of generalized cones. For example, Agin and Binford[Agin76a] use generalized cylinders, generalized cones with circular cross-sections whose radii are linear functions of distance along the axis[9]. Soroka[Soroka78a, Soroka79a, Soroka79b] allows elliptical cross-sections whose major and minor axis lengths vary linearly as a function of distance along a linear axis. The centers of the ellipses are constrained to lie on the axis and no twisting about the axis is permitted. Finally, Shani[Shani80a] allows both the cross-sections and the axis to be parametrically defined cubic splines.

[9] Such generalized cylinders are closely related to a degenerate case of Blum's symmetric axis transform in three dimensions.

Unfortunately, there are major problems with generalized cone decompositions. There is little understanding of the domain of figures that admit generalized cone decompositions. Further, such decompositions are usually not unique and constraints sufficient to force uniqueness are not, in general, known. Hence all programs that compute generalized cone decompositions are forced to use *ad hoc* rules to choose a single decomposition.

1.4.3. Prototypes

It is sometimes useful to assume that two outlines are members of the same shape and then to examine the nature of the transformation that maps one outline to the other. This approach has been taken to study biological growth[Thompson42a, le Gros Clark45a, Richards55a, Bookstein78a] and geographical relationships[Tobler78a]. Similar techniques can be used to describe shape by representing a *prototype* outline and a transformation that distorts it into other members of the shape. These techniques are characterized by their freedom from the need to describe explicitly or to decompose the outlines. The prototype is the description.

Few applications of the prototype paradigm have appeared in the computer shape description literature. Widrow[Widrow73a] proposed using flexible templates, which he called "rubber masks," as an alternative to using matched filters[10] in pattern recognition. Typical application of matched filters in pattern recognition entails defining a filter for each possible class (shape). Then, to classify a pattern, the matched filter for each class is applied to the pattern. The pattern belongs to the class corresponding to the filter that yields the largest output signal. Unfortunately, the matched filter is very sensitive to the size and orientation of the known signal or template.

[10] A matched filter is an optimal filter for detecting (and locating) a known signal in a noisy background. A classical result of signal processing (i.e., known at least since the early days of radar) shows that the matched filter simply cross-correlates the noise contaminated signal with the known signal[Castleman79a].

Widrow proposed developing flexible templates that could be distorted, within well-defined limits, to the unknown pattern. Then, the classification of the unknown pattern could be determined by fitting all possible flexible templates to the pattern. Indeed, he applied this approach with some success to classifying chromosomes, chromatograms, electroencephalogram (EEG) recordings, and electrocardiogram (EKG) waveforms. Each application requires a hand-crafted flexible template and an associated (iterative) fitting algorithm.

Techniques with similar flavor have long been used, albeit with different aims, in biology and even in art. One such scheme is elaborated at length here, not for its direct relevance to the present work, but because I believe it has, in concert with the present work, potential application to the full four-dimensional problem: description of both shape and shape change[11].

Imagine an outline drawn upon a planar rubber sheet marked with a rectangular grid. Then imagine stretching the sheet, without tearing, so as to distort the figure into another figure of the same shape and the superimposed rectangular grid into another grid, not necessarily rectangular. The distorted grid provides a vivid graphical representation of the "growth" of the original figure. This technique, expounded by D'Arcy Thompson[Thompson42a] in 1917 and used before him by Albrecht Dürer[12] to study proportion, has been applied to a number of biological problems (see, for instance, [le Gros Clark45a] and [Richards55a] for surveys, and Chapter 5 of [Bookstein78a] for a recent critique).

Despite the elegance of the idea and its frequent mention in the literature, it is not widely used. Bookstein explains [Bookstein78a, pp. 76-77]):

> It seems impossible to extract quantity from the Cartesian grid, as Thompson

[11] The scheme is also related to the "inbetweening" problem[Catmull78a] in computer-assisted animation.

[12] As cited by Thompson.

> formulated it, in any straightforward way.... For any "realistic" grid fitting the data more closely than Thompson's (which is not a difficult accomplishment), various ebbs and flows of the lines become apparent... In the effort to talk about what is there we open our mouths and become speechless....

Bookstein argues further that Thompson's fundamental error was the unsymmetric treatment of the two figures. Instead of choosing one of the figures as special, to have a rectangular grid superimposed upon it, the grids should be defined by the change between the two figures.

To examine Bookstein's reworking of Thompson's idea, I cast the rubber sheet analogy into mathematical terms[13]. The stretching operation is a diffeomorphism, a one-to-one differentiable transformation[14]. Near any point in its domain, a diffeomorphism can be approximated by a nonsingular linear transformation. A linear transformation maps a unit circle in its domain to an ellipse about the origin of its range. Further, the lines in the domain that map to the axes of the ellipse must be perpendicular[15]. Therefore, unless the nonsingular linear transformation is a similarity transformation[16], in which case the ellipse degenerates into a circle, there is a unique pair of perpendicular vectors (differentials of the diffeomorphism) that are mapped to perpendicular vectors. As a result, two curvilinear grids, one the image of the other, can be superimposed on the figure and its image respectively, so that the grid curves are perpendicular wherever they intersect.

These *biorthogonal grids* replace Thompson's grids. Not only are the two figures treated symmetrically, but the meaning of the biorthogonal grids is

[13] Bookstein's formulation is different, and slightly more general, than the presentation here.

[14] A lucid treatment of such transformations appears in Chapter 3 of [Osserman68a].

[15] Since the linear transformation that approximates the diffeomorphism is nonsingular, its inverse exists and is also a linear transformation. Recall that any linear transformation can be obtained by a rotation, followed by independent stretching along the rotated coordinate axes, followed by another rotation. Hence, it is easy to see that the inverse transformation maps the axes of the ellipse to a pair of perpendicular lines. Since a transformation composed with its inverse is the identity, the transformation must map the same pair of perpendicular lines to the ellipse axes.

[16] A similarity transformation is the composition of one or more rotations, reflections, or scale changes.

apparent. At each point of the figure, the transformation is completely defined by the dilation along each of the two biorthogonal grid curves through the point. The technique, then, reduces the transformation to differential "growth" in perpendicular directions at each point. Any rotation is a consequence of the "growth" along the grids.

To compute the biorthogonal grids from data one must first compute the diffeomorphism, the transformation from one figure to the other. Bookstein[Bookstein78a] describes a scheme for interpolating such a transformation from homologous landmarks of the two figures. For figures with few landmarks, there is little information to guide the interpolation. I shall propose an alternative, based on the symmetric axis transformation, in Chapter 2.

Tobler, in related work dealing with geographical problems[Tobler78a, Tobler78b], describes a technique similar in spirit to Bookstein's biorthogonal grids. In the context of cartography, Sen[Sen76a] discusses various types and measures of distortion produced by diffeomorphisms.

Broit[Broit81a, Bajcsy81a] has used the transformation approach to develop a registration scheme that finds a mapping from one three-dimensional object to another, each represented by a "stack" of two-dimensional slices. The mapping consists of two parts, one global, one local. The global mapping is limited to translation, rotation, and scale changes, while the local mapping is based on a mathematical model of a physical system that allows elastic deformation of a local region of one object into a corresponding local region of the other object. He has applied this technique to the problem of matching computed tomography studies of the brain to other such studies contained in a "brain anatomy atlas."

1.5. Overview of the Research

In the mid-60's, Blum[Blum67a] introduced a transformation, variously known as the symmetric axis transform (SAT), medial axis transform, or skeleton[17], that induces a decomposition of a figure into simpler figures. More recently, Blum[Blum73a, Blum74a, Blum78a] has proposed an elegant methodology, based on the SAT, for describing the shape of two-dimensional figures. I believe that, consciously or not, Blum exploited simultaneously and naturally two of the three paradigms described above. Therein lies a goodly portion of the elegance of his contribution. This notion is elaborated in Chapter 2.

For several reasons in addition to its elegance, the scheme introduced by Blum shows a great deal of promise as a shape description scheme for three-dimensional figures as well as two-dimensional figures:

(1) the decomposition induced by the SAT is unique, coordinate-system-independent, and, for a large class of figures with smooth boundary, natural;

(2) the decomposition induced by the SAT decomposes the figure into disjoint primitives;

(3) the resulting primitives are constrained, both individually and in the way they are juxtaposed;

(4) there is a functional relationship among properties of the SAT and curvature properties of the outline that can be used to show the intuitive meanings of SAT derived measures; and

(5) the definition of the SAT is easily generalized to three dimensions.

The present work focuses on generalizing Blum's methodology to three dimensions. The planned attack is three-pronged. First, a theoretical understanding of the properties of the transform in three dimensions is sought. Second, this understanding is used to generalize Blum's methodology and to develop an algorithm for computing a discrete approximation to the transform.

[17] The term skeleton has also been used in the picture processing literature as a generic name for the graph-like objects produced by a variety of "thinning" algorithms. It has yet another meaning in topology. Here I use the term "symmetric axis transform" or its acronym, SAT.

Finally, this generalized methodology must be applied to realistic data, such as organs extracted from clinical CT studies, to evaluate its utility. This dissertation addresses the first two of the three tasks, thereby laying the foundation for experimental work to evaluate the efficacy of the method in particular applications.

The symmetric axis (SA) of an object with a smooth boundary is the locus of points inside the object having at least two nearest neighbors on the object boundary. Together with the radius function, the distance from each point on the SA to a nearest point on the boundary, the SA forms the symmetric axis transform (SAT). The SAT and the boundary are equivalent; one can be reconstructed from the other. The usefulness of the SAT derives from the ease with which shape information can be extracted from it.

In three dimensions, the SA is a collection of smooth surface patches, possibly degenerating into space curves, connected together in a tree-like structure. Associated with each point of the SA are the boundary points comprising its set of nearest neighbors. Each SA patch, then, "goes up the middle" of a piece of the object bounded by the boundary points associated with the patch. Since the partition into patches follows naturally and uniquely from the SA definition, the shape description problem is reduced to describing the shape of each piece and the manner in which they are joined.

Each SA patch can be further partitioned into sub-patches, with associated sub-pieces, determined by the SA principal curvatures and a notion of radius curvature that I shall define in Chapter 3. Simple relationships that hold among the curvatures of the boundary, SA, and radius, enable each sub-piece to be labeled as one of a limited number of possible sub-piece types. Each subpatch, then, is described by the SA curvature and radius curvature and by the shape of the subpatch itself. The latter can be described by applying a version of the two-dimensional SAT generalized to measure distances along geodesics in a surface rather than along lines in a plane.

CHAPTER 2

THE 2D SYMMETRIC AXIS TRANSFORM

2.1. Definition

The *symmetric axis* of a figure F is the locus of centers of all maximal discs of F, those discs contained in F but in no other disc in F. Equivalently, if C is the outline that bounds F, the symmetric axis, SA(C), is the set of points in F having at least two nearest neighbors on C. Together with the *radius function*, the distance from each point on the SA to the nearest point on the outline, the SA forms the *symmetric axis transform* (SAT). The SAT and the boundary are equivalent[Calabi68a]; one can be reconstructed from the other. Its usefulness derives from the ease with which shape information can be extracted from the representation.

2.2. Point Types

The points of SA(C) can be classified into three types depending on the *order* of the point, the number of disjoint connected subsets of C comprising its set of nearest neighbors. *End* points are of order one, *normal* points of order two, and *branch* points of order three or more, corresponding to maximal discs touching in one, two, or more disjoint arcs respectively. Additionally, points are called *point contact* if each touching subset is a single point and *finite contact* otherwise. For C an outline[1], as defined in section 1.3, SA(C) is the union of

[1] This statement is true under somewhat weaker conditions than those imposed by our definition.

simple arcs, each a sequence of normal points bounded at each end by a branch or end point, that intersect each other at branch points only[Blum78a]. See Figure 2.1.

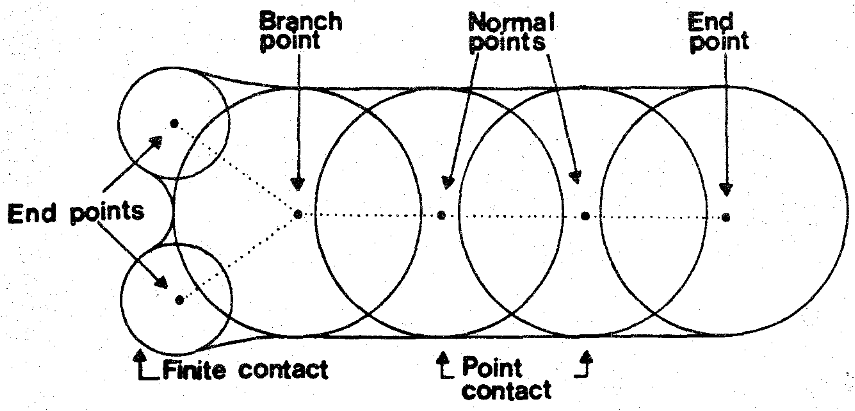


Figure 2.1.: Symmetric Axis Point Types

See [Calabi68a].

2.3. SAT Induced Decomposition

Let τ be the mapping from C onto SA(C) that maps a point P_C in C to the center of the maximal disc tangent to C at P_C. With each contiguous open interval of normal points, which Blum and Nagel call *simplified segments*, the inverse relation τ^{-1} associates two disjoint arcs of C. Consequently, as illustrated in Figure 2.2, F can be decomposed into a collection of two-sided parts, each associated with a simplified segment of SA(C), together with a collection of (possibly degenerate) circular sectors, each associated with a branch point or an end point.

To describe the connection structure of the decomposition, Blum and Nagel[Blum78a] define a labeled, directed graph with a node for each branch point and each end point, as illustrated in Figure 2.3. A pair of edges, one in each direction, connect a branch point and an end point or a pair of branch points whenever those two points bound the same simplified segment. Choose a direction of traversing a simplified segment and call the two associated arcs of C the *left* and *right boundary* arcs. The directed edges can be arranged so that, if one traverses a simplified segment in the direction indicated by an edge, there

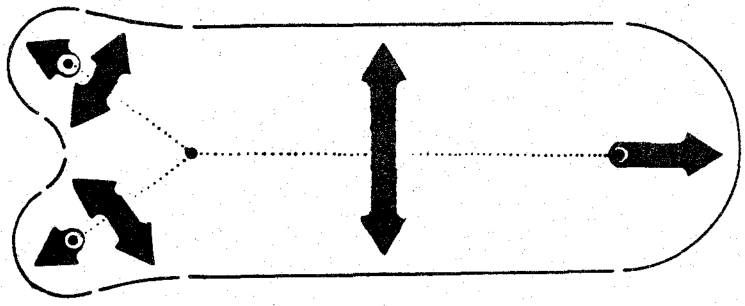


Figure 2.2.: SAT Induced Decomposition

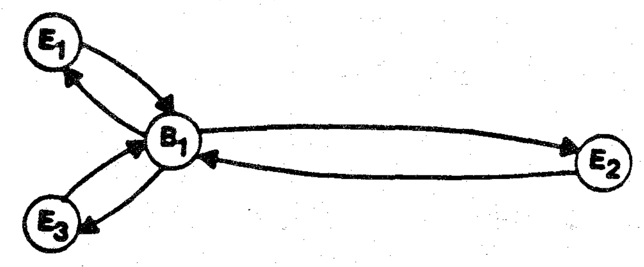


Figure 2.3.: Decomposition Connection Graph (after [Blum78a])

is an Eulerian circuit[2] of the graph that causes one of the two boundary arcs to traverse the outline. Labels attached to graph nodes describe properties of the corresponding branch or end points such as maximal disc radius and angular extent of finite contact, while labels attached to graph arcs describe the behavior of the two-sided parts associated with each simplified segment.

2.4. Simplified Segment Analysis

Again, choose a direction of traversing a simplified segment. The angle β between the tangent to C at a point P_C and the tangent to SA(C) at $\tau(P_C)$ is called the *object angle*, and is shown by Blum and Nagel[Blum78a] to be the arcsin of the first derivative of the disc radius at $\tau(P_C)$ with respect to axis arc length. See Figure 2.4. The algebraic signs of the object angle and its derivative with respect to axis arc length, $\frac{d\beta}{ds}$, called the *object curvature*, partition the segment into canonical primitives, called *width shapes*, juxtaposed one after the other.

[2] An Eulerian circuit visits each edge of the graph exactly once.

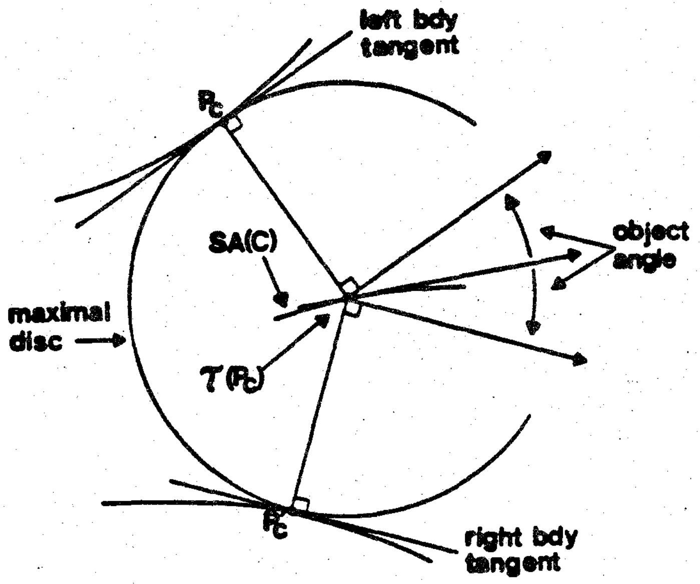


Figure 2.4.: Normal Point Geometry (Point Contact)

The width shapes, shown in Figure 2.5 for a straight interval of the symmetric axis, are completely determined by the first and second derivatives of

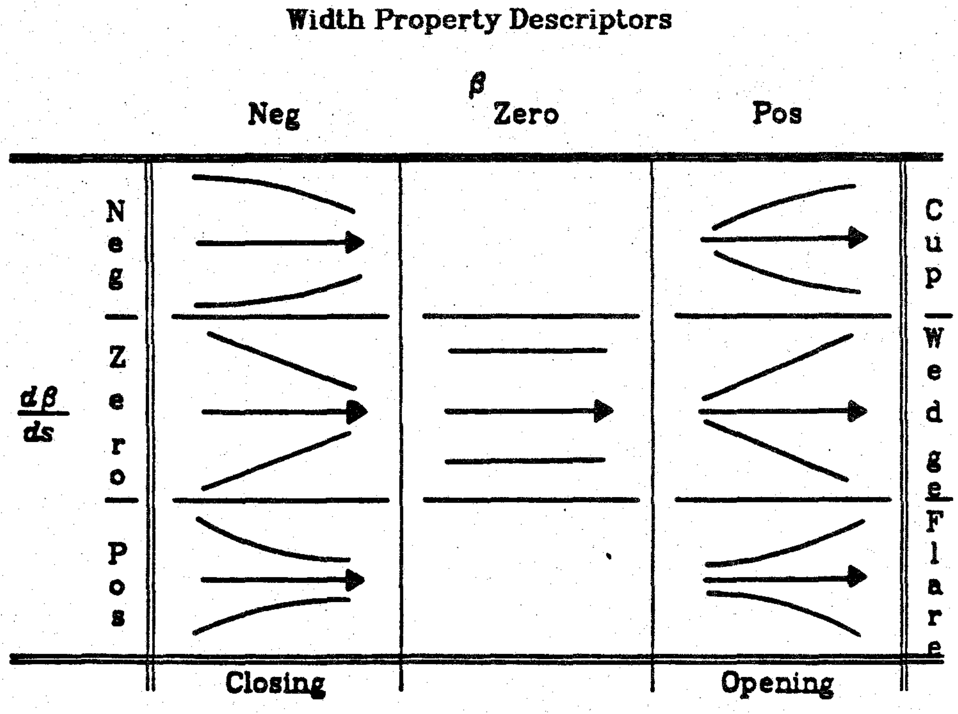


Figure 2.5.: Width Shapes (after [Blum78a])

the radius function with respect to arc length along the axis. Hence, the partition obtained is independent of the curvature of the SA itself. Moreover, outline smoothness imposes simple syntax constraints on the string of width shapes associated with each simplified segment. Similarly, the algebraic signs of the symmetric axis curvature and its first derivative partition a simplified segment into canonical *axis shapes* juxtaposed one after the other. See Figure 2.6. The two partitions are independent of each other; each characterizes different properties of the simplified segment and associated boundary arcs.

The axis curvature of a simplified segment reflects the degree to which the associated boundary arcs curve in the same direction, while object curvature

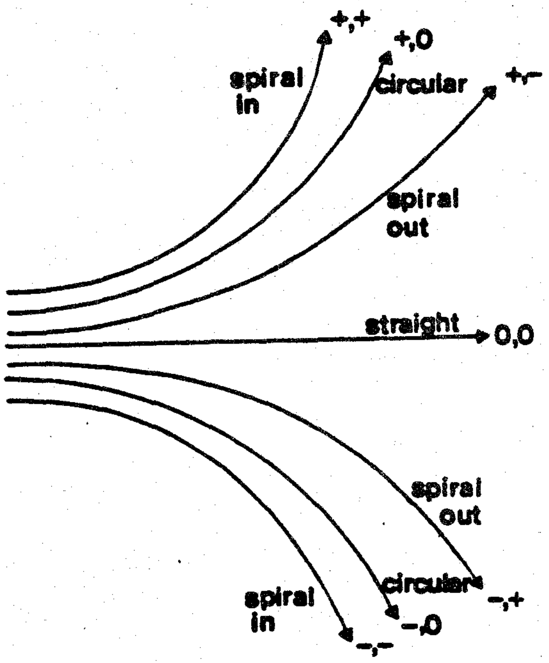


Figure 2.6.: Axis shapes (after [Blum78a])

reflects the symmetry of the associated boundary arcs about the simplified segment. If, for example, disc radius is held constant while axis curvature is changed, the associated boundary arcs may change from convex, to straight, to concave in a manner depending on the curvature. Indeed, for normal points, Blum and Nagel[Blum78a] give an explicit functional relationship among axis curvature, object curvature, object angle, and associated boundary arc curvatures. Using this relationship, they have been able to characterize the behavior of the boundary arcs associated with each simplified segment in terms of the width and axis shapes of the segment.

Other simplified segment partitioning schemes can be devised. Bookstein[Bookstein78a], for example, sketches a multivariate statistical technique for analyzing simplified segments by computing principal components of samples of the vector valued function τ (defined on page 22). Most likely, no single partitioning scheme is adequate for all purposes.

2.5. On the Elegance of the SAT

In Section 1.5, I stated my belief that the elegance of Blum's contribution is due, in part, to the manner in which he exploited simultaneously and naturally two of the three shape description paradigms described in Section 1.4. Now that his method has been sketched, I shall elaborate.

In many endeavors, an appropriate representation is the key to parsimony and clarity. The mathematician and physicist find coordinate systems peculiarly suited to their problems, the programmer data structures that simplify his tasks. So too, the practitioner of shape description must find representations that bring to the fore that information considered essential. Marr and Nishihara[Marr78a] identify three criteria for judging the effectiveness of a figure representation:

Accessibility. Can the representation be computed from the data available? Is the time and space required for the computation acceptable?

Scope and uniqueness. For what class of figures is the representation suitable? Is there a unique representation for each figure?

Stability and sensitivity. There is an inherent conflict between stability and sensitivity. To be useful, it must be possible to derive from the representation "similar" descriptions for all figures of the same shape. Yet, simultaneously, it must be possible to represent subtle differences between figures. These conflicting desiderata can be met only if it is possible to decouple the stable, more constant figure properties from properties sensitive to subtle variations. In the terminology of Section 1.4, it must be possible to "represent, then discard."

Blum's representation fares well by these criteria. As we shall discuss in Chapter 5, a number of algorithms for computing the SAT from other representations, all of practical space and time complexity, have been developed. The scope of the SAT includes not only figures as I have defined them, but also objects with holes and/or a finite number of corners. Moreover, the representation is unique, coordinate-system independent, and imposes no loss of information. Other representations, too, satisfy the first and second criteria. The third criterion is the most demanding: no known representation completely satisfies it. The SAT comes close.

By combining two of the three shape description paradigms, "represent, then discard" and decomposition, Blum's methodology for deriving descriptions from the SAT representation provides an attractive mechanism for dealing with the demands of the third criterion. As I have described, the SAT induces a unique decomposition of the figure. The decomposition is particularly attractive because the resulting primitives are highly constrained: each is either a two-sided part or a collection of circular sectors. Each two-sided part is determined by a simplified segment whose curvatures describe the overall "curvature trend" of the part. Varying sensitivity is obtained by discarding more or less of the curvature information, for example, by partitioning the simplified segment into intervals where the axis curvatures lie within certain ranges, rather than just considering the algebraic signs of the axis curvatures. Similarly, the same process applied to object angle and object curvature achieves more or less sensitivity to the symmetry of the boundary arcs about the simplified segments.

We now turn to a different aspect of the same problem. Two symmetric axes are said to have identical *topologies* if their directed graphs are isomorphic (ignoring labels). Over the range of figures for which the topology of the symmetric axis is constant, the process described above seems adequate for choosing an application-dependent tradeoff between stability and sensitivity. For ranges of figures where the topology is not constant other techniques are required. If a threshold on the radius of the maximal discs that comprise a figure is imposed from below, the sensitivity of the symmetric axis topology to small, local perturbations of the outline is reduced. Similar results can be achieved by placing a threshold on the ratio of boundary arc length to simplified segment length[Blum78a].

2.6. Unsolved Problems and Research Directions

Though the two-dimensional symmetric axis transform is reasonably well understood, several open problems, some fundamental, remain.

(1) Given a subset of the plane (or, more generally, a subset of $\mathbf{R}^n$) and a real-valued function defined over that subset, several sufficient conditions are known that ensure that the subset–function pair are the symmetric axis and radius function for some outline. Do necessary conditions exist and, if so, what are they? (This problem was posed in [Calabi68a]).

(2) Consider again the rubber sheet analogy of Section 1.4.3. Draw an outline on a rubber sheet and find its symmetric axis and radius function. Then stretch the rubber sheet, transforming the outline into another, and compute the symmetric axis and radius function of the new outline. The diffeomorphism that maps one outline to the other need not, in general, map

one symmetric axis to the other. Is it possible to characterize diffeomorphism–symmetric axis pairs for which the symmetric axis topology remains constant?

(3) To use Bookstein's biorthogonal grids (described in Section 1.4.3) to study shape change, one must first compute the transformation from one figure to the other by interpolating from homologous landmarks. For figures with few landmarks, there is little information to guide the interpolation. I propose computing the symmetric axis of each figure separately and then interpolating the transformation from corresponding branch points and end points of each. When both symmetric axes share the same topology, the correspondence is easy to find. When the topologies differ, the correspondence is more difficult to find and may require elimination of ''unimportant'' simplified segments from each symmetric axis. This problem is intimately tied to the next one.

(4) Recently, a few researchers have begun to use *relational homomorphisms*, homomorphisms from one relation to another, to match structural shape descriptions against prototype descriptions[Haralick78a, Shapiro80a, Shapiro81a]. This formulation is particularly interesting because the model on which it is built deals explicitly with inexact matching. Can this work be applied to the problem of comparing two labeled graphs derived from Blum's methodology?

(5) Outlines in the plane are but a specific case of closed curves on two-dimensional manifolds. The definition of the SAT is easily extended to this more general case by measuring distance along geodesics in the manifold. Blum's shape description methodology can probably be adapted to this more general situation.

(6) One application of shape description is the study of variation within classes of objects. For simplicity, assume a constant symmetric axis topology over all objects of the study. Are there any meaningful statistics that could be applied to measures derived from the SAT to analyze within-class variation?

(7) Image segmentation algorithms able to use *a priori* information have been developed (see e.g. [Ashkar78a]). Can SAT based models be used as a source of such information?

CHAPTER 3

THE 3D SYMMETRIC AXIS TRANSFORM

3.1. Basic Definitions and Properties

We must begin by defining the domain of our discussion, outlines and figures in three-dimensional space. An *outline* is a smooth, closed surface that partitions the complement of the outline into two disjoint sets, one bounded, called the inside, and one unbounded, called the outside. This excludes surfaces having no distinct inside and outside, such as the Klein bottle, surfaces with corners, edges, or cusps, such as polyhedra, and surfaces with boundary curves, such as a sphere with a circle cut out of it. As in two dimensions, a *figure* is an outline together with its inside. For simplicity, throughout the remainder of the dissertation we consider explicitly only outlines that are topologically equivalent to a sphere, those outlines that can be formed by stretching, but not tearing, a sphere. However, except where otherwise noted, this restriction only simplifies the exposition of the ideas and results presented hereafter; it does not reduce their generality.

We turn now to the SAT in three dimensions. The definitions of the two-dimensional SAT given in Section 2.1, apply in three dimensions as well, the only difference being that maximal discs become maximal spheres. As in two dimensions points on the symmetric axis can be classified into three types: end points,

[1] Much of this chapter will appear in the journal *Computer Graphics and Image Processing.* It is

normal points, and branch points. End points and branch points are, in general, no longer isolated points, but rather, curves in space. Open connected sets of normal points[2], again called *simplified segments*, are bounded by possibly degenerate space curves of branch and end points. In general, each simplified segment is a surface rather than a curve, though they sometimes degenerate into a space curve. As before, the figure can be decomposed into a collection of two-sided parts each associated with a simplified segment, together with pieces of canal surfaces[3] each associated with a branch or end point curve[Blum79a].

In the remainder of this chapter, we develop the mathematical tools we shall need in our analysis of simplified segment behavior in three dimensions. In particular, we define a notion of radius curvature and derive a relationship among boundary curvature, simplified segment curvature, and radius curvature. Then, in Chapter 4, we shall use that relationship to partition each simplified segment into a collection of canonical primitives.

3.2. Background

It is necessary to digress briefly to discuss curvature of smooth surfaces in general. Denote the tangent plane to S at P by T_PS. In a small neighborhood of P the curvature of S can be characterized by examining the curvature of curves on S through P. Consider the *normal sections* at P, those curves defined by the intersection of S with planes containing the normal at P. Each normal section is a curve in the plane defining it, and hence has a well-defined curvature at P that measures the deviation of the curve from its tangent line through P. Further,

used here by permission of Academic Press, Inc.

[2] I implicitly assume that such bounded, connected sets exist. Though I offer no proof, I offer the following argument. The maximal sphere centered on a normal point touches and is tangent to the outline in two distinct points. By making an infinitesimal change in the sphere radius it can be moved slightly while maintaining contact with the tangent plane at each touching point. Since the outline is smooth, its tangent plane at any point on the outline approximates the outline in an open neighborhood about that point. Hence, the new position of the sphere defines a new normal point in a neighborhood of the original one.

[3] A canal surface is the envelope of a family of spheres, possibly of varying radius, with centers lying on a space curve[Hilbert52a].

since the tangent line lies in T_pS, the *normal section curvature* also measures the deviation of S from T_pS in the direction of the tangent line. By rotating the defining plane about the normal, we get all normal sections and their curvatures, and hence a complete characterization of the deviation of the surface from its tangent plane. See Figure 3.1.

To express all of the normal section curvatures in a finite way, we (arbitrarily) call one side of the tangent plane the positive side and the other the negative side, and attach a sign to the normal section curvatures according to whether the normal section lies on the positive or negative side of the tangent plane. It can then be shown (e.g., Section 4-8, [Millman77a]) that as the defining plane is rotated about the normal, either the normal section curvature assumes its maximum and minimum values, called *principal curvatures*, in two orthogonal directions, called *principal directions*, or the normal section curvatures are constant. Further, each normal section curvature is completely determined by the principal curvatures and the angle between the defining plane and the principal directions.

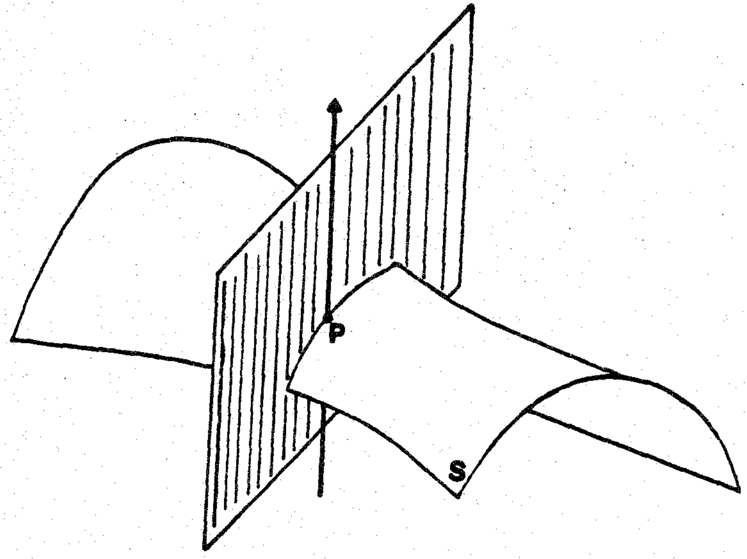


Figure 3.1.: Geometry of Surface Curvature

The product of the principal curvatures is called the *Gaussian curvature* of S at P, and is denoted K_S, while their average is its *mean curvature*, H_S. The behavior of S at P is characterized by the signs of the Gaussian and mean curvatures. For $K_S > 0$, in a local neighborhood of P, all normal sections lie on one side of the tangent plane, the side determined by the sign of the mean curvature. The surface is cup-shaped at P. On the other hand, for $K_S < 0$ the normal sections about one principal direction lie above the tangent plane and those about the other lie below, giving S a saddle shape at P. The remaining case, $K_S = 0$, is a transition between the two: in one principal direction the surface has flattened while in the other it may remain curved. When both principal curvatures are zero, S is planar at the point and the principal directions cease to exist. For a fascinating discussion of this and other interpretations of both Gaussian and mean curvature, see Ch. IV of [Hilbert52a].

3.3. Characterization of Sphere Radius

Now, let S be a simplified segment in $\mathbf{R}^3$ and let P be a point contact normal point on S, i.e. the maximal sphere centered at P touches the outline in two disjoint *touching points*, sometimes called the *boundary points associated with* P. Further, we assume that S and the radius function, r (defined precisely below), are twice continuously differentiable at P.[4] See Figure 3.2.

We now turn to characterizing the behavior of the sphere radius. In two dimensions, disc radius was analyzed as a function of a single parameter, arc length along the symmetric axis. Unfortunately, in three dimensions no single parameter suffices. Instead, we examine the first and second derivatives of the radius function along curves in infinitely many directions through the point P.

[4] This assumption can be "justified" using an argument much like that used in footnote 2, but using second, rather than first order approximations to the outline.

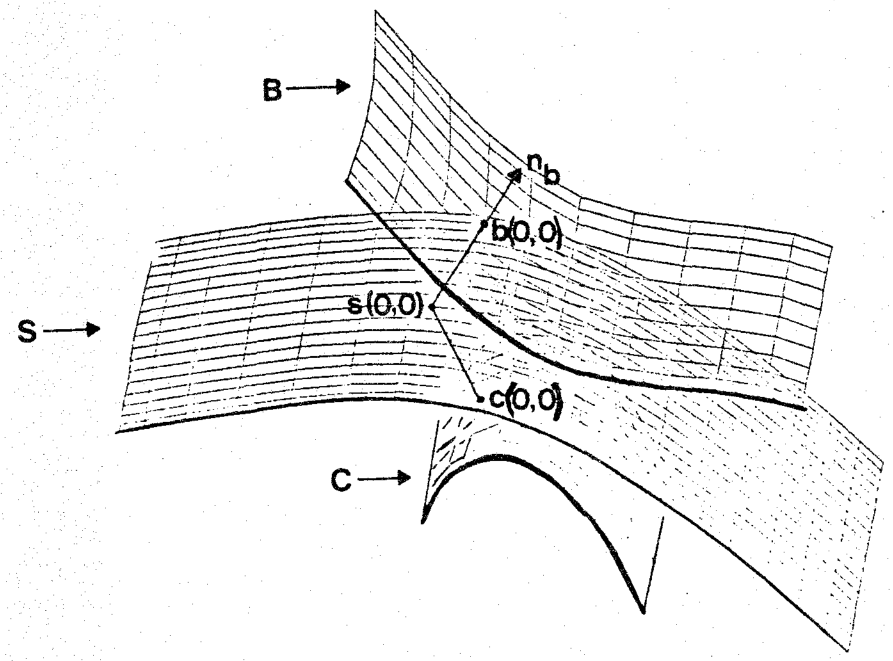


Figure 3.2.: 3D SAT Geometry

Pick any direction about P. Then, the *first directional derivative* of the radius function at P in the specified direction is the first derivative of the radius function with respect to arc length along any curve with tangent vector lying in that direction. It is easy to show that the first directional derivative is well-defined, i.e. is independent of the choice of the curve in the specified direction[cf. [Millman77a], sec. 4-7].

Similarly, the *second directional derivative* of the radius function at P in the specified direction can be defined to be the second derivative of the radius function with respect to arc length along the curve. Unfortunately, this is not well-defined without constraining the choice of the curve. Since we are interested in the behavior of the radius function, not in the curvature of the curve in S, we require the curve to be straight in a small neighborhood of P. More precisely, we require that in an infinitesimal neighborhood about P, the orthogonal projection of the curve onto T_PS be a line in the specified direction. There is a unique curve, called a *geodesic*, that satisfies this condition (Section 4-5, [Millman77a]). Hence, we define the second directional derivative in a specified direction to be the second derivative of the radius function with respect to arc length along the geodesic in that direction.

Below, we prove that, like normal section curvatures, the second directional derivative of the radius function assumes its maximum and minimum values in two orthogonal directions which, by analogy, I call the *principal curvatures* and *principal directions of the radius function*, respectively. Further, the second directional derivative in any direction is completely determined by the principal curvatures and the angle between the direction and a principal direction. I also define the *Gaussian* and *mean curvatures of the radius function* analogously, and denote them K_R and H_R.

3.4. Curvature Relations

We can now state our goal more precisely. We seek a functional relationship among the Gaussian and mean curvatures of S at P, the Gaussian and mean curvatures of the outline at the associated boundary points, and the Gaussian and mean curvatures of the radius function at P. In Section 3.4, I present the desired relationship, limiting the mathematical prerequisites to the background material presented in Sections 3.2 and 3.3 and an exposure to vector calculus such as can be found in [Thomas60a]. I shall prove this relationship in Section 3.4.1.

3.4.1. Formulation

We begin by imposing local curvilinear coordinate systems about normal points on simplified segments, thus bringing the techniques of calculus to bear. Let S be a simplified segment in $\mathbf{R}^3$. Except at finite contact normal points, which we ignore hereafter, we assume S to be a C^2 surface. Hence if we let U be an open subset of $\mathbf{R}^2$ with coordinates u^1 and u^2, we can let $\mathbf{s}: U \rightarrow S$ be a C^2 coordinate patch (surface patch) on S with linearly independent partial derivatives $\frac{\partial \mathbf{s}}{\partial u^i}$ denoted by $\mathbf{s}_i$ and called *coordinate vectors.*

Choose a set of basis vectors for $\mathbf{R}^3$ and let **Y** and **Z** be two vectors represented in terms of that basis. To distinguish between a vector, **X**, and the n-tuple that represents it with respect to some basis, we denote the n-tuple by X. Then, an inner product of **Y** and **Z**, denoted <**Y**,**Z**>, is given by $Y^T GZ$, where G is a 3 by 3 matrix such that <**Y**,**Z**> = <**Z**,**Y**> and <**Y**,**Y**> > 0 for all non-zero **Y**. For the remainder of this chapter, we will use the particular inner product defined by $G = I$ (the identity matrix) when the basis vectors are orthonormal. This is nothing more than the dot product, $Y^T Z$, often used in $\mathbf{R}^3$. Though the representation of the inner product depends on the basis vectors chosen, the inner product itself is basis-independent. Hence we use < > to denote the inner product of two vectors, regardless of the basis used to represent them.

It is always possible to choose **s** so that the coordinate vectors are orthonormal at the point P = **s**(0,0) (Section 6-2, [Millman77a]). Thus, without loss of generality, we choose **s** so that $<\mathbf{s}_i(0,0), \mathbf{s}_j(0,0)> = \delta_{ij}$, where δ_{ij} is the Kronecker delta.

The tangent plane to S at $\mathbf{s}(u^1, u^2)$ is a two-dimensional subspace of $\mathbf{R}^3$ spanned by the coordinate vectors $\mathbf{s}_1$ and $\mathbf{s}_2$. Consequently, the unit normal at $\mathbf{s}(u^1, u^2)$, $\mathbf{n}_s(u^1, u^2)$, is $\frac{\mathbf{s}_1 \times \mathbf{s}_2}{|\mathbf{s}_1 \times \mathbf{s}_2|}$. Similarly, let B and C be the boundary surfaces associated with S as shown in Figure 3.2, let $\mathbf{b}(u^1, u^2)$ and $\mathbf{c}(u^1, u^2)$ be the points on B and C associated with $\mathbf{s}(u^1, u^2)$, and let $r: \mathrm{S} \to \mathbf{R}^1$ map a point on S to the radius of the maximal sphere centered at that point.

The maximal sphere centered at $\mathbf{s}(u^1, u^2)$ is tangent to the boundary surface B at $\mathbf{b}(u^1, u^2)$ with the boundary normal, $\mathbf{n}_b(u^1, u^2)$, lying along a radius of the sphere. See Figure 3.2. Letting $r(u^1, u^2)$ denote $r(\mathbf{s}(u^1, u^2))$,

$$\mathbf{b}(u^1, u^2) = \mathbf{s}(u^1, u^2) \pm r(u^1, u^2)\mathbf{n}_b(u^1, u^2),$$

with the choice of sign determined by the direction of $\mathbf{n}_b$. Since $\mathbf{n}_b$ itself is determined only up to sign, choose $\mathbf{n}_b$ pointing away from S as shown in Figure 3.2, giving

$$\mathbf{b}(u^1, u^2) = \mathbf{s}(u^1, u^2) + r(u^1, u^2)\mathbf{n}_b(u^1, u^2). \tag{3.1}$$

Similarly,

$$\mathbf{c}(u^1, u^2) = \mathbf{s}(u^1, u^2) + r(u^1, u^2)\mathbf{n}_c(u^1, u^2). \tag{3.2}$$

Let $\alpha(t): \mathrm{I} \subset \mathbf{R}^1 \to S$ be the geodesic on S passing through P, where I is some interval of $\mathbf{R}^1$ containing 0, t is arc length along the curve, and $\alpha(0) = \mathrm{P}$. Let $\mathbf{X}$ be the tangent vector of $\boldsymbol{\alpha}$ at P, $\frac{d\alpha}{dt}(0)$. Since $\boldsymbol{\alpha}$ is parameterized by arc length and lies on S, $\mathbf{X}$ is a unit vector in the plane $\mathrm{T_PS}$, the tangent plane of S at P.

Definition 3.1: The *first directional derivative of r in the* $\mathbf{X}$ *direction* is
$r_{\mathbf{X}} = \frac{dr(\alpha)}{dt}(0)$. ■

Definition 3.2: The *second directional derivative of r in the* $\mathbf{X}$ *direction* is
$r_{\mathbf{XX}} = \frac{d^2r(\alpha)}{dt^2}(0)$. ■

We let λ_1 and λ_2, $\lambda_1 \leq \lambda_2$, denote the principal curvatures of S at P and let $\mathbf{e}_1$ and $\mathbf{e}_2$ be unit vectors in the corresponding principal directions. Since each principal direction is determined by a line in $\mathrm{T_PS}$, there are two unit vectors each from which to choose $\mathbf{e}_1$ and $\mathbf{e}_2$. As shown below, we can, without loss of generality, require that $\mathbf{e}_1 \times \mathbf{e}_2 = \mathbf{n}_s$; the results of this chapter are independent of the choice made from the remaining two possibilities. Similarly, let γ_1 and γ_2, $\gamma_1 \leq \gamma_2$, denote the principal curvatures the radius function and let $\mathbf{f}_1$ and $\mathbf{f}_2$ denote the corresponding principal directions.

3.4.2. Boundary Curvature Equations

In two dimensions, the object angle, the angle between the tangent to the symmetric axis at a point and the tangent to the associated boundary point, is determined by the arcsin of the first derivative of the radius function. A similar relation holds in three dimensions.

Theorem 3.1: Let **X** be a unit vector in T$_P$S. Then, the directional derivative of r in the **X** direction, $r_{\mathbf{X}}$, is $-<\mathbf{n}_b, \mathbf{X}>$. ■

That is, in three dimensions, the angle between a symmetric surface tangent vector at a normal point and the normal at the associated boundary point is determined by the arccos of the first directional derivative of the radius function in the direction of the tangent vector. An analogous result holds for $\mathbf{n}_c$.

The major result of the chapter follows:

Theorem 3.2: Let

$$h = \frac{\gamma_1(1 - r_{\mathbf{f}_2}^2) + \gamma_2(1 - r_{\mathbf{f}_1}^2)}{2<\mathbf{n}_s, \mathbf{n}_b>^2} + \frac{\lambda_1(1 - r_{\mathbf{e}_2}^2) + \lambda_2(1 - r_{\mathbf{e}_1}^2)}{2<\mathbf{n}_s, \mathbf{n}_b>}, \text{ and} \tag{3.3}$$

$$k = \lambda_1\lambda_2 + \frac{\gamma_1\gamma_2}{<\mathbf{n}_s, \mathbf{n}_b>^2} + \frac{\lambda_1 r_{\mathbf{e}_2\mathbf{e}_2} + \lambda_2 r_{\mathbf{e}_1\mathbf{e}_1}}{<\mathbf{n}_s, \mathbf{n}_b>}. \tag{3.4}$$

Then, the Gaussian and mean curvatures of the boundary surface B at $\mathbf{b}(0,0)$ are

$$H_B = \frac{h - rk}{1 - 2rh + r^2k} \tag{3.5}$$

and

$$K_B = \frac{k}{1 - 2rh + r^2k}. \tag{3.6}$$

■

These equations give the Gaussian and mean curvatures of the boundary surface, B, in terms of properties of the radius and symmetric surface, together with the angle between the boundary normal, $\mathbf{n}_b$, and the symmetric surface normal, $\mathbf{n}_s$. Analogous equations for boundary surface C are obtained when the qualifying subscripts b and B are replaced by c and C respectively.

At first glance, it appears that knowledge of the boundary normal is prerequisite to evaluating h and k, and hence the boundary curvatures. This is not the case. Since $\mathbf{n}_s$, $\mathbf{e}_1$, and $\mathbf{e}_2$ are orthonormal, $<\mathbf{n}_s,\mathbf{n}_b>^2+<\mathbf{n}_b,\mathbf{e}_1>^2+<\mathbf{n}_b,\mathbf{e}_2>^2 = 1$. Hence, up to sign, $<\mathbf{n}_s,\mathbf{n}_b>$ is determined by $r_{\mathbf{e}_1}$ and $r_{\mathbf{e}_2}$.

Choosing the sign of $<\mathbf{n}_s,\mathbf{n}_b>$ chooses either boundary surface B or C. As symmetry suggests, and application of theorem 3.1 proves, $\mathbf{n}_b$ and $\mathbf{n}_c$ are reflections of each other through the symmetric surface tangent plane. Thus, by symmetry about the tangent plane, $<\mathbf{n}_s,\mathbf{n}_b> = <\mathbf{n}_c,-\mathbf{n}_s>$ and hence $<\mathbf{n}_s,\mathbf{n}_b> = -<\mathbf{n}_c,\mathbf{n}_s>$. Consequently, if we replace $<\mathbf{n}_s,\mathbf{n}_b>$ by $\pm<\mathbf{n}_s,\mathbf{n}_b>$ in (3.3) and (3.4), the curvature relations hold for either boundary, the choice being determined by the sign.

To understand the geometric significance of h and k, consider the surface B′ defined by

$$\mathbf{b}'(u^1,u^2) = \mathbf{s}(u^1,u^2)+r'(u^1,u^2)\mathbf{n}_b(u^1,u^2),$$

where $r'(u^1,u^2) = r(u^1,u^2)-r(0,0)$. It is not difficult to see that B′ passes through the point P = $\mathbf{s}(0,0)$ and at each (u^1,u^2) has the same unit normal vector as does B. B′ and B are called *parallel surfaces*. See Figure 3.3. Since the derivatives of r' and r are identical, we can evaluate (3.5) and (3.6) at (0,0), substituting r' for r, obtaining $k = K_{B'}$, and $h = H_{B'}$. Thus, the terms h and k in (3.5) and (3.6) are the mean and Gaussian curvatures, respectively, of the

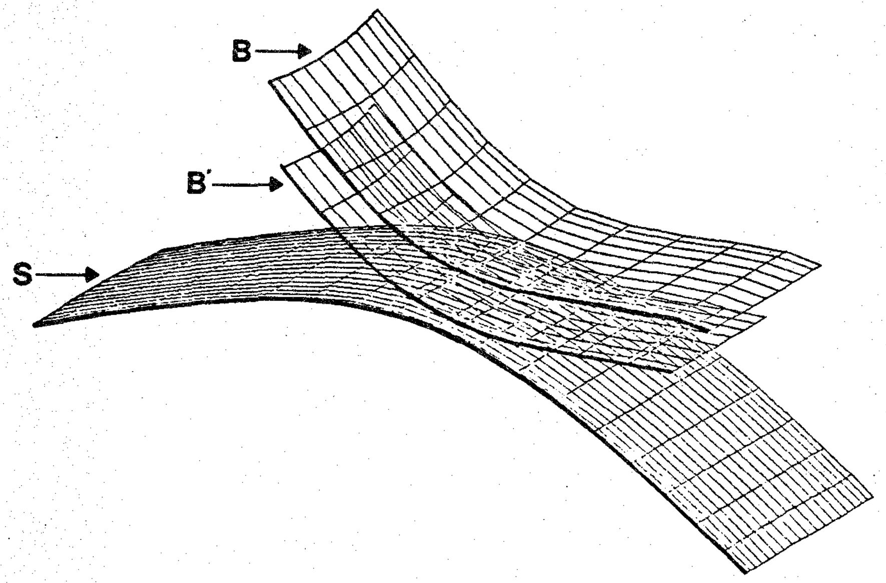


Figure 3.3.: Surface Parallel to Boundary Surface

surface parallel to B passing through P. Therefore, (3.5) and (3.6) express the change in boundary curvature due to change in distance from the symmetric surface. Blum and Nagel[Blum78a] use a similar relationship in the two-dimensional case to derive boundary curvature from parallel curve curvature. Analogous results hold for the surface parallel to C through P when the sign of $<\mathbf{n}_s, \mathbf{n}_b>$ is changed.

Though the symmetric surface and radius function together contain no information not contained in the boundary surfaces, examining each alone reveals different aspects of the shape of the boundary surface. Intuitively, symmetric surface curvature reflects the overall "curvature trend" of the two-sided piece, i.e. the degree to which the boundary surfaces curve in the same direction. Radius curvature, on the other hand, reflects the symmetry of the boundary surfaces about the symmetric surface, the degree to which both boundary surfaces curve in opposite directions.

To see this, observe in (3.3) that symmetric surface curvatures λ_1 and λ_2 contribute with equal magnitude but opposite sign to the mean curvature of the two boundary surface parallels, while radius curvatures γ_1 and γ_2 contribute equally to each. Since the boundary surface normals are directed away from the symmetric surface, boundary surface mean curvatures of opposite sign imply curvature in the same direction. Further, it can be shown that the signs of the Gaussian and mean curvatures of each boundary surface are the same as the signs of the curvatures of the corresponding parallel surface. Hence, our intuitive notions of the meanings of symmetric surface curvature and radius curvature are confirmed.

3.5. Proof of Curvature Relations

In this section, we prove the results presented in Section 3.4.2, assuming results from the elementary differential geometry of surfaces[Millman77a, Stoker69a].

3.5.1. Curvature Quadratic Forms

First, we show that the second directional derivative of the radius function is a quadratic form. Hence, by properties of quadratic forms (Section 17, [Gel'fand61a]), the principal curvatures and principal directions exist and behave as claimed in Section 3.3.

Lemma 3.3: $r_{\mathbf{XX}}$ is a quadratic form over unit vectors, $\mathbf{X}$, in T_PS.
Proof: For two scalar functions of t, α^1 and α^2, $\alpha(t) = \mathbf{s}(\alpha^1(t),\alpha^2(t))$. Since T_PS is a vector space spanned by $\mathbf{s}_1$ and $\mathbf{s}_2$, there are scalars X^i such that

$\mathbf{X} = \sum_{i=1}^{2} X^i \mathbf{s}_i$. Using the chain rule, $\frac{d\alpha}{dt} = \sum_{i=1}^{2} \frac{d\alpha^i}{dt}\mathbf{s}_i$, so $\frac{d\alpha^i}{dt}(0) = X^i$. Applying the chain rule again,

$$\frac{dr(\alpha)}{dt}(t) = r_{\mathbf{X}} = \sum_{i=1}^{2} \frac{\partial r(\mathbf{s})}{\partial u^i}\frac{d\alpha^i}{dt}. \tag{3.7}$$

Differentiating and substituting X^i for $\frac{d\alpha^i}{dt}$,

$$\frac{d^2r(\alpha)}{dt^2}(t) = \sum_{i=1}^{2} \frac{\partial r(\mathbf{s})}{\partial u^i}\frac{d^2\alpha^i}{dt^2} + \sum_{i=1}^{2}\sum_{j=1}^{2} X^i X^j \frac{\partial^2 r(\mathbf{s})}{\partial u^i \partial u^j}.$$

The geodesic $\boldsymbol{\alpha}$ is characterized by the differential equations

$$\frac{d^2\alpha^k}{dt^2} = -\sum_{i=1}^{2}\sum_{j=1}^{2} \Gamma_{ij}^k \frac{d\alpha^i}{dt}\frac{d\alpha^j}{dt}, \quad k = 1,2,$$

where the Γ_{ij}^k are the Christoffel symbols of the second kind of S [Millman77a, Stoker69a], which measure the tangential components of the second partial derivatives $\mathbf{s}_{ij}$. Combining the last two equations, denoting $\frac{\partial r(\mathbf{s})}{\partial u^i}$ by r_i and $\frac{\partial^2 r(\mathbf{s})}{\partial u^i \partial u^j}$ by r_{ij}, and rearranging terms, we see that since $r_{ij} = r_{ji}$ and $\Gamma_{ij}^k = \Gamma_{ji}^k$, $r_{\mathbf{XX}}$ is a quadratic form in $\mathbf{X}$:

$$r_{\mathbf{XX}} = Q(\mathbf{X}) = X^T Q X, \text{ with} \tag{3.8}$$

$$Q = [q_{ij}] = [r_{ij} - \sum_{k=1}^{2} r_k \Gamma_{ij}^k]. \tag{3.9}$$

■

For any unit vector $\mathbf{X}$ in $\mathrm{T_P S}$, the second directional derivative of r in the direction defined by $\mathbf{X}$ is given by $Q(\mathbf{X})$.

Since Q represents the quadratic form $Q(\mathbf{X})$ with respect to an orthonormal basis of $\mathrm{T_P S}$, over all unit vectors $\mathbf{X}$ in $\mathrm{T_P S}$, $Q(\mathbf{X})$ assumes its minimum value at the eigenvector of Q corresponding to the smallest eigenvalue, γ_1 and its maximum value at the eigenvector corresponding to the largest eigenvalue, γ_2. Further, the values assumed are γ_1 and γ_2 respectively and the eigenvectors are orthogonal if the eigenvalues are distinct (Section 17, [Gel'fand61a]). By solving the characteristic equation of Q, it is easy to see that $\gamma_1\gamma_2 = \det(Q)$ and $\gamma_1 + \gamma_2 = \mathrm{tr}(Q)$.

Similarly, the ***second fundamental form*** of S, $\mathrm{II}(\mathbf{X})$, is a quadratic form over unit vectors in $T_P S$ that gives the curvature of the normal section in the direction $\mathbf{X}$[Millman77a, Stoker69a]. Letting $L_s = [L_{s_{ij}}]$ be the matrix defining the second fundamental form with respect to the $\{\mathbf{s}_1, \mathbf{s}_2\}$ basis of $T_P S$, we have $\mathrm{II}(\mathbf{X}) = X^T L_s X$.

Thus, two quadratic forms are defined at each point of S. One, the second fundamental form, gives the curvature of normal sections through the point in any direction, while the other gives the second derivative of the radius along the geodesic in the same direction. Since the normal to a geodesic is everywhere normal to the surface on which it lies, the geodesic and the normal section share a common normal vector. By construction, they have the same tangent vector and hence, the same curvature (cf. [Stoker69a], sec IV-12). Therefore, one quadratic form measures the curvature of S along geodesics and the other measures the radius function second derivative along the same geodesics.

3.5.2. Matrix Formulation

In this section, we derive an equation relating the matrices, Q and L_s, that determine the radius and symmetric surface curvatures respectively, to the matrix defining the second fundamental form, and hence the curvature, of each boundary surface.

Dropping explicit mention of (u^1, u^2) and taking partial derivatives of (3.1),

$$\mathbf{b}_i = \mathbf{s}_i + r_i \mathbf{n}_b + r \mathbf{n}_{b_i} . \tag{3.10}$$

We can solve for r_i by taking the inner product of both sides of (3.10) with $\mathbf{n}_b$. Since $\mathbf{n}_b$ is a vector of constant magnitude, it is perpendicular to its derivative, $\mathbf{n}_{b_i}$. Thus, since $\mathbf{b}_i$ is perpendicular to $\mathbf{n}_b$ by definition,

$$r_i = -<\mathbf{s}_i, \mathbf{n}_b>. \tag{3.11}$$

Taking partial derivatives again,

$$r_{ij} = -<\mathbf{s}_{ij},\mathbf{n}_b> - <\mathbf{s}_i,\mathbf{n}_{b_j}>.$$

Using Gauss's formulas[Millman77a, Stoker69a], $\mathbf{s}_{ij} = L_{\mathbf{s}_{ij}}\mathbf{n}_s + \sum_{k=1}^{2}\Gamma_{ij}^{k}\mathbf{s}_k$, and the definition of the coefficients of the second fundamental form[Millman77a, Stoker69a], $L_{\mathbf{s}_{ij}} = <\mathbf{s}_{ij},\mathbf{n}_s>$, we obtain

$$r_{ij} = -L_{\mathbf{s}_{ij}}<\mathbf{n}_s,\mathbf{n}_b> - \sum_{k=1}^{2}\Gamma_{ij}^{k}<\mathbf{s}_k,\mathbf{n}_b> - <\mathbf{s}_i,\mathbf{n}_{b_j}>. \qquad (3.12)$$

Analogous results for boundary surface C follow from (3.2), though for brevity we defer further consideration of C until the end of this section.

Define the matrices $G_b = [G_{b_{ij}}] = [<\mathbf{b}_i,\mathbf{b}_j>]$ and $L_b = [L_{b_{ij}}]$ representing the first and second fundamental forms of B at $\mathbf{b}(u^1,u^2)$ with respect to the $\{\mathbf{b}_1(u^1,u^2),\mathbf{b}_2(u^1,u^2)\}$ basis of the tangent plane at $\mathbf{b}(u^1,u^2)$. Since $\mathbf{n}_b$ is a vector of constant magnitude, the $\mathbf{n}_{b_j}$ are perpendicular to it. Hence, they lie in the tangent plane and are expressed as a linear combination of the $\mathbf{b}_i$ by *Weingarten's equations*

$$\mathbf{n}_{b_j} = -\sum_{i=1}^{2} W_{b\,j}^{\,i}\,\mathbf{b}_i, \qquad (3.13)$$

where $W_b = [W_{b\,j}^{\,i}] = G_b^{-1}L_b$, and is called the *Weingarten map* of B [Millman77a, Stoker69a]. Letting $A = [<\mathbf{s}_i,\mathbf{b}_j>]$ and combining Weingarten's equations with (3.9), (3.11), and (3.12),

$$\begin{aligned} AW_b &= [r_{ij}] + <\mathbf{n}_s,\mathbf{n}_b>L_s - \sum_{k=1}^{2} r_k[\Gamma_{ij}^{k}] \\ &= Q + <\mathbf{n}_s,\mathbf{n}_b>L_s. \qquad (3.14) \end{aligned}$$

Equation (3.14) relates boundary curvature, as expressed by W_b, to radius curvature, as expressed by Q, and to symmetric surface curvature, as expressed by L_s. We seek the boundary curvatures in terms of properties of the radius and symmetric surface. Our approach is to solve for the two invariants of the matrix equation (3.14), the determinant and trace. We then solve the resulting two equations simultaneously.

3.5.3. Determinant Equations

Substitute Weingarten's equations (3.13) into (3.10) and solve for the $\mathbf{s}_i$, giving

$$\mathbf{s}_1 = (1+rW_b{}_1^1)\mathbf{b}_1 + rW_b{}_1^2\mathbf{b}_2 - r_1\mathbf{n}_b, \text{ and} \tag{3.15}$$

$$\mathbf{s}_2 = rW_b{}_2^1\mathbf{b}_1 + (1+rW_b{}_2^2)\mathbf{b}_2 - r_2\mathbf{n}_b. \tag{3.16}$$

Recalling that $A = [<\mathbf{s}_i,\mathbf{b}_j>]$ and defining $T = \begin{bmatrix} 1+rW_b{}_1^1 & rW_b{}_1^2 \\ rW_b{}_2^1 & 1+rW_b{}_2^2 \end{bmatrix}$, we use (3.15) and (3.16) to obtain $A = TG_b$ and consequently, since $W_b = G_b{}^{-1}L_b$, that $AW_b = TL_b$. Substituting into (3.14) then gives

$$TL_b = Q + <\mathbf{n}_s,\mathbf{n}_b>L_s. \tag{3.17}$$

To evaluate the determinant of the left side of (3.17), we use theorem 3.1 (which we now prove) and an additional result:

Theorem 3.1: Let **X** be a unit vector in T$_p$S. Then the directional derivative of r in the **X** direction, $r_\mathbf{X}$, is $-<\mathbf{n}_b,\mathbf{X}>$.

Proof: Let X^1 and X^2 be the components of **X** in the $\{\mathbf{s}_1,\mathbf{s}_2\}$ basis, i.e. $\mathbf{X} = \sum_{i=1}^{2} X^i\mathbf{s}_i$. So, $<\mathbf{n}_b,\mathbf{X}> = \sum_{i=1}^{2} X^i<\mathbf{n}_b,\mathbf{s}_i>$ which, by (3.11), is $-\sum_{i=1}^{2} X^i r_i$. Thus, by (3.7), $r_\mathbf{X} = -<\mathbf{n}_b,\mathbf{X}>$. ∎

Lemma 3.4: Letting $g_b = \det(G_b)$, $g_b\det^2(T) = <\mathbf{n}_s,\mathbf{n}_b>^2$.

Proof: Recall that $[<\mathbf{s}_i,\mathbf{s}_j>] = I$, where I is the two-by-two identity matrix. Then, using (3.15) and (3.16), by straightforward algebra, it is not difficult to show that

$$g_b\det^2(T)(T^T)^{-1}G_b{}^{-1}T^{-1} = I-R, \tag{3.18}$$

where $R = \begin{bmatrix} r_2{}^2 & -r_1r_2 \\ -r_1r_2 & r_1{}^2 \end{bmatrix}$. Taking the determinant of both sides and applying theorem 3.1,

$$g_b \det^2(T) = 1-r_1{}^2-r_2{}^2$$

$$= 1-<\mathbf{n}_b, \mathbf{s}_1>^2-<\mathbf{n}_b, \mathbf{s}_2>^2$$

$$= <\mathbf{n}_s, \mathbf{n}_b>^2 ,$$

where the last step follows because $\mathbf{n}_s$, $\mathbf{s}_1$, and $\mathbf{s}_2$ are orthonormal. ■

Thus the determinant of the left side of (3.17) is

$$\begin{aligned} \det(TL_b) &= \det(T)\det(L_b) \\ &= \frac{<\mathbf{n}_s, \mathbf{n}_b>^2 \det(G_b{}^{-1}L_b)}{\det(T)} \\ &= \frac{<\mathbf{n}_s, \mathbf{n}_b>^2 K_B}{\det(T)}, \end{aligned} \tag{3.19}$$

where $K_B = \det(W_b)$ is the Gaussian curvature of B.

We now evaluate the determinant of the right side of (3.17). Recalling that the determinant is invariant under change of basis, we change from the $\{\mathbf{s}_1, \mathbf{s}_2\}$ basis of T_PS to that defined by the eigenvectors of L_s. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be eigenvectors of L_s corresponding to the eigenvalues λ_1 and λ_2 respectively. Since eigenvectors are determined only up to a non-zero multiplicative constant and since $\mathbf{e}_1$ and $\mathbf{e}_2$ lie in the tangent plane T_PS and are orthogonal to each other, we can, without loss of generality, choose the $\mathbf{e}_i$ to be unit vectors so that $\mathbf{e}_1 \times \mathbf{e}_2 = \mathbf{n}_s$. Similarly, let $\mathbf{f}_1$ and $\mathbf{f}_2$ be unit eigenvectors of Q corresponding to the eigenvalues γ_1 and γ_2 so that $\mathbf{f}_1 \times \mathbf{f}_2 = \mathbf{n}_s$. In terms of their respective eigenvector bases, the transformations represented by L_s and Q in terms of the $\{\mathbf{s}_1, \mathbf{s}_2\}$ basis, are represented by $\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ and $\begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix}$, i.e. $L_s \approx \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ and $Q \approx \begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix}$, where $\approx$ denotes matrix similarity.

Representing both transformations in terms of the $\{\mathbf{e}_1, \mathbf{e}_2\}$ basis requires examining the relationship between the $\mathbf{e}_i$ and the $\mathbf{f}_i$. Let θ be the counterclockwise angle from $\mathbf{e}_1$ to $\mathbf{f}_1$. Then, with respect to the $\{\mathbf{f}_1,\mathbf{f}_2\}$ basis, $\mathbf{e}_i = \Theta\mathbf{f}_i$, where $\Theta = \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}$. As shown in Figure 3.4, θ is determined only up to a multiple of π; thus, Θ is determined only up to sign. Changing from the $\{\mathbf{f}_1,\mathbf{f}_2\}$ basis to the $\{\mathbf{e}_1,\mathbf{e}_2\}$ basis, $\begin{bmatrix}\gamma_1 & 0 \\ 0 & \gamma_2\end{bmatrix} \approx \pm\Theta^{-1}\begin{bmatrix}\gamma_1 & 0 \\ 0 & \gamma_2\end{bmatrix}(\pm\Theta) = \Theta^T\begin{bmatrix}\gamma_1 & 0 \\ 0 & \gamma_2\end{bmatrix}\Theta$. Therefore, $Q + <\mathbf{n}_s,\mathbf{n}_b> L_s$ is similar to $\Theta^T\begin{bmatrix}\gamma_1 & 0 \\ 0 & \gamma_2\end{bmatrix}\Theta + <\mathbf{n}_s,\mathbf{n}_b>\begin{bmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix}$, which is easily seen to have a determinant of

$$<\mathbf{n}_s,\mathbf{n}_b>^2\lambda_1\lambda_2 + \gamma_1\gamma_2 + <\mathbf{n}_s,\mathbf{n}_b>(\lambda_1\gamma_1+\lambda_2\gamma_2-(\gamma_1-\gamma_2)(\lambda_1-\lambda_2)\cos^2\theta)\,. \qquad (3.20)$$

Note that (3.20) is independent of θ if either $\gamma_1 = \gamma_2$ or $\lambda_1 = \lambda_2$. Consequently, when either pair of eigenvalues fail to be distinct and the principal directions are not well-defined, arbitrary directions can be chosen.

Combining (3.19) and (3.20) and rearranging terms,

$$\frac{K_B}{\det(T)} = \lambda_1\lambda_2 + \frac{\gamma_1\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>^2} + \frac{\lambda_1(\gamma_1\sin^2\theta+\gamma_2\cos^2\theta)+\lambda_2(\gamma_1\cos^2\theta+\gamma_2\sin^2\theta)}{<\mathbf{n}_s,\mathbf{n}_b>}. \qquad (3.21)$$

Recall that $\mathbf{e}_1 = \mathbf{f}_1\cos\theta - \mathbf{f}_2\sin\theta$ and $\mathbf{e}_2 = \mathbf{f}_1\sin\theta + \mathbf{f}_2\cos\theta$. Equation (3.21) can be simplified by observing that $Q(\mathbf{e}_1) = [\cos\theta \;\; -\sin\theta]\begin{bmatrix}\gamma_1 & 0 \\ 0 & \gamma_2\end{bmatrix}[\cos\theta \;\; -\sin\theta]^T =$

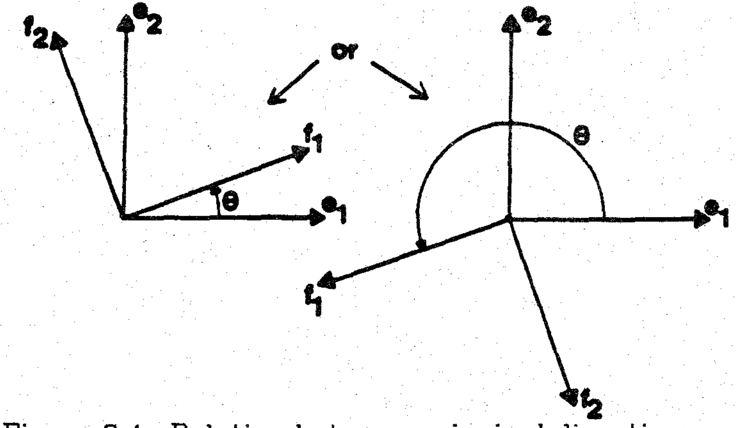


Figure 3.4.: Relation between principal directions

$\gamma_1\cos^2\theta+\gamma_2\sin^2\theta$ and $Q(\mathbf{e}_2) = \gamma_1\sin^2\theta+\gamma_2\cos^2\theta$. Hence, by (3.8), (3.21) becomes

$$\frac{K_B}{\det(T)} = \lambda_1\lambda_2 + \frac{\gamma_1\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>^2} + \frac{\lambda_1 r_{\mathbf{e}_2\mathbf{e}_2}+\lambda_2 r_{\mathbf{e}_1\mathbf{e}_1}}{<\mathbf{n}_s,\mathbf{n}_b>}. \tag{3.22}$$

3.5.4. Trace Equations

The second equation relating boundary curvature to radius and symmetric surface curvature results from taking the trace of (3.14). Recalling that $A = TG_b$, it follows from (3.14) and (3.18) that

$$g_b \det^2(T)(T^T)^{-1}W_b = (Q+<\mathbf{n}_s,\mathbf{n}_b>L_s)-R(Q+<\mathbf{n}_s,\mathbf{n}_b>L_s) .$$

Hence, since $\mathrm{tr}(W_b) = 2H_B$, $\det(W_b) = K_B$, $\mathrm{tr}(Q) = 2H_R$, and $\mathrm{tr}(L_s) = 2H_S$, taking the trace of both sides gives

$$2g_b(rK_B+H_B)\det(T) = 2(H_R+<\mathbf{n}_s,\mathbf{n}_b>H_S)-\mathrm{tr}(RQ)-<\mathbf{n}_s,\mathbf{n}_b>\mathrm{tr}(RL_s) . \tag{3.23}$$

Two observations enable us to evaluate $\mathrm{tr}(RL_s)$ and, by analogous reasoning, $\mathrm{tr}(RQ)$. First, simple algebra reveals that $\mathrm{tr}(RL_s)$ is nothing more than the second fundamental form of S, evaluated at $[r_2 \ \ -r_1]$, $[r_2 \ \ -r_1]L_s[r_2 \ \ -r_1]^T$. Second, with respect to the $\{\mathbf{e}_1,\mathbf{e}_2\}$ basis, the second fundamental form is represented by the diagonal matrix $\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}$. Hence, letting $[a^1 \ a^2]$ represent, with respect to $\{\mathbf{e}_1,\mathbf{e}_2\}$, the vector represented by $[r_2 \ \ -r_1]$ in the $\{\mathbf{s}_1,\mathbf{s}_2\}$ basis, $\mathrm{tr}(RL_s) = [a^1 \ a^2]\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}[a^1 \ a^2]^T$.

Let V be the matrix of transition from the $\{\mathbf{s}_1,\mathbf{s}_2\}$ basis to the $\{\mathbf{e}_1,\mathbf{e}_2\}$ basis[Gel'fand61a], i.e. the matrix such that $[r_2 \ \ -r_1]^T = V[a^1 \ a^2]^T$. Since the columns of V are the coordinates of the $\mathbf{e}_i$ in the $\{\mathbf{s}_1,\mathbf{s}_2\}$ basis, and since the $\mathbf{e}_i$ are orthonormal, $V^T V = I$, where I is the two-by-two identity matrix. Thus, $\det(V) = \pm 1$ which is non-zero. Therefore, we can solve for $[a^1 \ a^2]$ obtaining

$\pm[r_1 V_{12}+r_2 V_{22} \ -r_1 V_{11}-r_2 V_{21}]$. Since by the definition of V, $\mathbf{e}_i = \sum_{j=1}^{2} V_{ji}\mathbf{s}_j$, by using (3.11) we see that $[\alpha^1\ \alpha^2] = \pm[-<\mathbf{n}_b,\mathbf{e}_2>\ <\mathbf{n}_b,\mathbf{e}_1>]$ and hence, that $\mathrm{tr}(RL_s) = \lambda_1<\mathbf{n}_b,\mathbf{e}_2>^2+\lambda_2<\mathbf{n}_b,\mathbf{e}_1>^2$. Analogously, $\mathrm{tr}(RQ) = \gamma_1<\mathbf{n}_b,\mathbf{f}_2>^2+\gamma_2<\mathbf{n}_b,\mathbf{f}_1>^2$. Finally, combining these results with (3.23), Lemma 3.4, and the definition of mean curvature as the average of principal curvatures, we obtain

$$\begin{aligned} 2<\mathbf{n}_s,\mathbf{n}_b>^2\frac{rK_B+H_B}{\det(T)} &= \gamma_1(1-<\mathbf{n}_b,\mathbf{f}_2>^2)+\gamma_2(1-<\mathbf{n}_b,\mathbf{f}_1>^2)+ \\ &\quad <\mathbf{n}_s,\mathbf{n}_b>(\lambda_1(1-<\mathbf{n}_b,\mathbf{e}_2>^2)+\lambda_2(1-<\mathbf{n}_b,\mathbf{e}_1>^2)) \ , \end{aligned} \tag{3.24}$$

which, using theorem 3.1, can be rewritten as

$$\frac{rK_B+H_B}{\det(T)} = \frac{\gamma_1(1-r_{\mathbf{f}_2}^2)+\gamma_2(1-r_{\mathbf{f}_1}^2)}{2<\mathbf{n}_s,\mathbf{n}_b>^2}+\frac{\lambda_1(1-r_{\mathbf{e}_2}^2)+\lambda_2(1-r_{\mathbf{e}_1}^2)}{2<\mathbf{n}_s,\mathbf{n}_b>}. \tag{3.25}$$

3.5.5. Solution

The right sides of equations (3.25) and (3.22) are the right sides of (3.3) and (3.4) respectively. Recall that $K_B = \det(W_b)$ and $H_B = ½\mathrm{tr}(W_b)$. Then, by straightforward algebra, $\det(T) = 1+r^2K_B+2rH_B$. Substituting this into (3.22) and (3.25), we obtain a linear system of two equations in the two unknowns H_B and K_B, with solutions (3.5) and (3.6). This proves theorem 3.2.

3.6. Summary

Blum's symmetric axis transform defines a unique decomposition of a figure into disjoint, two-sided pieces, each with its own surface (axis) of symmetry and associated boundary surfaces. I have defined measures of the radius function and have shown how these measures and the symmetric surface curvatures are related to the boundary surface curvatures. In particular, I have shown that the Gaussian and mean curvatures of the boundary surfaces are determined by nine measures, each with a geometric interpretation:

(1) the symmetric surface curvature as determined by two principal curvatures and a principal direction;

(2) the radius curvature as determined by two principal curvatures and a principal direction;

(3) directional derivatives of the radius function as determined by the angles between either boundary normal and the two symmetric surface principal directions, called *width angles* after Blum[Blum73a]; and

(4) the radius function itself.

It will be shown in Chapter 4 that these measures, and the curvature relationship derived from them, subsume the two-dimensional measures and curvature relationship given by Blum and Nagel[Blum78a].

3.7. Unsolved Problems and Research Directions

In three dimensions, many problems remain open. Certainly, all of the problems sketched in Section 2.6 exist in three dimensions. Many appear even more difficult in three than in two dimensions. I list here problems peculiar to three or more dimensions.

(1) Though our discussion has been restricted to outlines topologically equivalent to a sphere, the definition of the SAT applies to other surfaces in $\mathbf{R}^3$. For example, the symmetric axis of a torus is a circle. To my knowledge, there has been no thorough study of the relationship between the topological classification of an outline[5] and properties of its symmetric surface. Intuition suggests that the connectivity number[6] of the symmetric surface of a closed surface is the same as that of the surface itself.

[5] All outlines (as I have defined them in general) are known to be topologically equivalent either to a sphere, to a torus, or to two or more tori "glued" together. See [Hilbert52a] or [Massey67a].

[6] On a surface with connectivity number n, $n-1$ closed curves can be drawn on the surface without cutting the surface in two, but any n closed curves must cut the surface in at least two parts.

(2) If one deforms an outline locally, as if pushing on a balloon with a finger, one of two things will happen to the SAT. Either both the radius function and the symmetric surface will change slightly, but with no change in symmetric surface topology, or at least one new simplified segment will emerge. Which case occurs clearly depends on the relative magnitudes of the radius function and the radii of curvature (principal curvature reciprocals) at the deformed point. Can this dependence be made precise and a catalog of possible topology changes be produced[7]?

(3) Blum[Blum79a] has noted that the touching sets of the maximal spheres can be closed curves on the sphere as well as points and areas. For instance, over intervals where the symmetric surface degenerates to a curve, all touching sets become circles. Blum suggests that examining the symmetric axis of the touching sets on the maximal spheres themselves, will yield information about the behavior of the symmetric surface near branch curves. This idea needs further, more detailed study.

(4) Many computer "vision" systems use a shape description as a source of *a priori* information. Since the description is used to model the expected scene, it is essential to be able to compute rapidly the appearance of the model from different viewpoints. With the aid of a computerized symbolic manipulation facility, analytic solutions for portions of this problem have been obtained for certain classes of generalized cones[Brooks79a]. If three-dimensional SAT's are ever to find use in computer "vision" systems, similar problems will need to be addressed.

(5) The definition of the SAT clearly extends to higher-dimensional closed manifolds embedded in metric spaces. It also seems clear that topological properties of the symmetric surface are related to the topology of the

[7] Though my knowledge of the subject is very shallow, I believe this situation could be modeled with catastrophe theory[Poston78a, Saunders80a].

manifold. What is this relationship and is it of any value in studying the many unanswered questions about the topology of higher dimensional manifolds? Can SAT's of higher dimensional manifolds be applied to describing shape change?

CHAPTER 4

SIMPLIFIED SEGMENT PARTITIONING

4.1. Overview

In this chapter, I use the measures defined in Chapter 3, together with the relationships among them, to demonstrate several *simplified segment partitions*. The symmetric surface of the figure under study is split at branch curves into simplified segments. Then, each simplified segment is partitioned into regions, like countries on a map, where some set of properties holds over each region. This partition induces, in turn, a partition of each associated boundary surface into corresponding regions. As a result, each two-sided part (defined by a simplified segment and its two associated boundary surfaces) is decomposed into a collection of two-sided primitives as illustrated in Figure 4.1.

Though I describe below a particular partitioning scheme, with particular sets of primitives, I make no claim that they are in any way optimal. Moreover, I believe that any general optimality claim is impossible, because the choice of primitives largely determines the compromise between sensitivity and stability (see Section 2.5). Since that compromise is inherently application-dependent, any optimality claim must be made in the context of a specific application. But even then there is little, if any, theory on which to base a criterion function to be optimized. Therefore, I make no optimality claim. Rather, I seek to

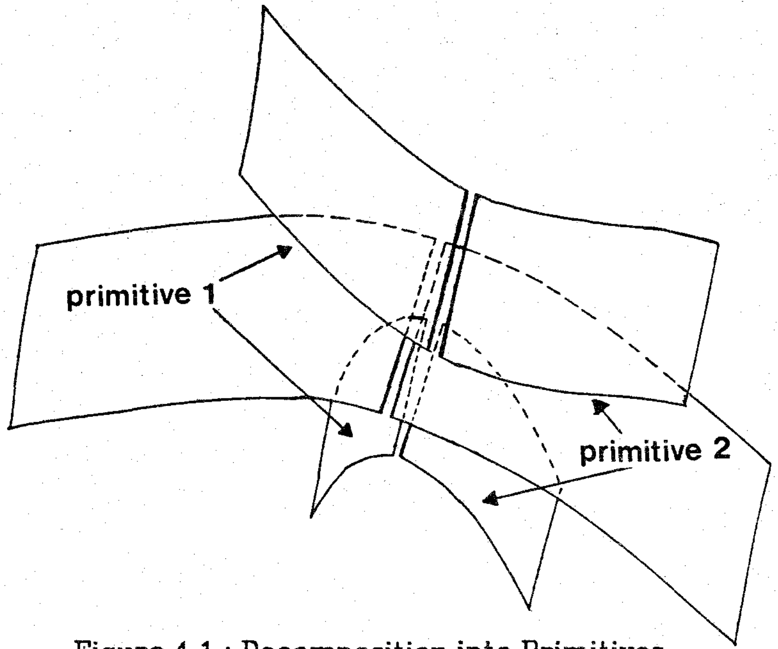


Figure 4.1.: Decomposition into Primitives

(1) demonstrate the feasibility of the partitioning approach;

(2) provide a catalog of primitives from which the shape description practitioner can pick and choose according to the goals of his analysis; and

(3) develop a mathematical framework useful for examining properties of the primitives set forth here and, hopefully, for extending this work or developing new primitives.

A *simplified segment partition* consists of two components: a *primitive set* and a *primitive adjacency graph.* The simplified segment is partitioned into a collection of disjoint primitives, each an element of the primitive set. The primitive adjacency graph, then, maintains information about the spatial relationships among the primitives comprising a simplified segment.

Rather than introducing a single primitive set, I define independently three sets of primitives: *width primitives*, based on radius function properties, *axis primitives*, based on simplified segment curvatures, and *boundary primitives*, based on boundary surface curvatures. Each set of primitives is derived from different properties of the simplified segment and radius function and hence captures different characteristics of the two-sided part associated with the simplified segment.[1] For some applications it might be appropriate to use more than one primitive set, either separately or combined together to produce a new, larger, cartesian product primitive set wherein each primitive is an ordered tuple of two or three primitives, one from each of two or three primitive sets. However, the relationship between symmetric surface curvature, radius function curvature, and boundary surface curvature given by Theorem 3.2 places constraints on which combinations of primitives from different primitive sets can exist.

4.2. Width Primitives

Using properties of the radius function alone, the simplified segment and its associated boundary surfaces can be partitioned into a collection of two-sided primitives called *width primitives*. Since the radius function behavior reveals the symmetry of the boundary surfaces about the simplified segment (cf. Section 3.4.2), the primitives differ, one from another, solely in the way their boundary surfaces move toward or away from the simplified segment. It is important to realize that this is different than the behavior of the boundary surfaces themselves; the latter is a function of the symmetric surface curvature as well as of the radius function behavior.

4.2.1. Overview

To be useful, the set of width primitives must capture the qualitative behavior of the radius function while simultaneously ignoring extraneous detail. Therefore, we begin our discussion of width primitives by sketching a technique

[1] Though the space I shall devote to discussing width primitives dwarfs the space devoted to the other two primitive sets, I do not mean to imply that width primitives are more important than the

for analyzing the qualitative properties of a function defined over a surface, here the radius function. Since the basic ideas differ little from those used to analyze the behavior of a surface defined by its height function over the x–y plane, I shall occasionally discuss the radius function as if it were such a surface. I shall also use terms such as "horizontal" and "above," even though they are not coordinate-system independent concepts. I take these liberties only to build intuition. We shall see that the width primitives are defined only in terms of coordinate-system independent properties of the radius function.

Many elementary calculus texts give a recipe for sketching rapidly the graph of a function of one variable $y = f(x)$, that is, for analyzing the qualitative behavior of the function. The recipe usually consists of three basic steps.[2] First, find the values of x for which the first derivative is positive and for which it is negative. Then, at sign transitions, apply the second derivative test to determine whether each transition point is a local minimum or local maximum. To determine which parts of the curve "hold water" and which "spill water," find the values of x for which the second derivative is positive and for which it is negative. Finally, sketch the curve between the values of the function at the sign transitions found above, having the curve rise or fall as indicated by the sign of the first derivative, and "spill" or "hold water" as indicated by the sign of the second derivative.

The width shapes defined by Blum and Nagel for the two-dimensional symmetric axis (see Section 2.4) can be derived by applying the aforementioned recipe to the radius as a function of arc length along a simplified segment. Each width shape is an interval over which the signs of the first and second derivatives of the radius function remain constant. Width shapes are juxtaposed at local extrema and inflection points of the radius function.

others.

[2] See, for example, Section 3-4 of [Thomas60a].

Unfortunately, in three dimensions there is no one-dimensional axis along which to perform the analysis. First and second derivatives of the radius function become first and second directional derivatives. It therefore makes no sense to talk about signs of the first and second derivatives without specifying a direction. Instead, we use the local extrema of the radius function together with radius function curvature to split the simplified segment into primitives. Both are intuitively appealing. Radius function extrema indicate "pinches" and "bulges" in the boundary surfaces with respect to the simplified segment, while radius function curvatures provide qualitative information about the manner in which the boundary surfaces are pulling away from or moving toward the simplified segment.

In the remainder of Section 4.2.1, I sketch briefly a partition comprised of two components, ***slope districts***, based on first derivative behavior, and ***curvature districts***, based on second derivative behavior. Then, in the following sections, I present a more detailed, more formal, discussion of each component.

Slope Districts. Slope districts are an old, and, in concept, simple idea. They were described more than a century ago by Cayley[Cayley59a] and by Maxwell[Maxwell70a] in the context of topography. More recently, Warntz[Warntz66a] has reviewed the earlier work and suggested that their techniques might also prove useful in studying demographic and economic trends. Pfaltz[Pfaltz76a, Pfaltz78a] has preliminarily investigated using a similar technique for organizing large spatial data bases. Finally, Johnson[Johnson78a] and Williams[Williams82a] are applying a variant of this approach to the higher-dimensional problem of interpreting electron density functions resulting from X-ray diffraction studies of crystals. Here, I describe the approach intuitively and informally, following Cayley and Maxwell.

To describe the idea concretely, let us return to the surface analogy mentioned above. In particular, imagine that the radius function is the height function of a mountainous island. Lacking a three-dimensional relief map, one would use a contour map to study the island's topography. At sea level there is a single contour, a closed curve surrounding the island. As one moves higher, contours bounding local maxima, called *peaks*, become smaller and smaller, ultimately becoming a point. Likewise, as one moves lower, contours bounding local minima, called *pits*, diminish in size, becoming points as well. However in some cases, as one smoothly changes elevation, two or more contours meet at a single point, forming a single contour that cuts itself. That point, called a *pass*, is neither a local maxima nor a local minima, for moving to-and-fro one ascends, while moving left and right one descends.

It is not hard to convince oneself that, in general[3], pits, peaks, and passes are isolated from each other. It therefore is meaningful to consider paths between them. A curve drawn so that it crosses at right angles every contour it meets is called a *slope line*. At every point on a slope line there are two possible directions of travel, one ascending, one descending. Moreover, the two directions are the directions of steepest ascent and of steepest descent, respectively, from any point on the slope line. Therefore, if one travels along a slope line in the ascending direction, one must eventually reach a peak or a pass; traveling in the opposite direction, one must eventually reach a pit, a pass, or the island coastline. From any point on a slope line, the portion traversed by traveling in the ascending direction is called the *ascending slope line* from that point, while the other portion is called the *descending slope line* from that point.

With the exception of pits, peaks, and passes, there is a unique slope line through every point. All points whose slope lines descend to the same pit form a

[3] Throughout Section 4.2.1, I use the term "in general" to mean that I am describing the situation that "almost always" occurs. Treating degenerate situations here would be distracting. I deal with them below.

region, called a *dale*, while those points whose slope lines ascend to the same peak form a *hill*. Therefore, the whole island can be divided into dales and, independently, into hills.

Since it is impractical to divide the island into hills and dales by examining individually every point on the island, we examine those slope lines that separate dales, called *ridge lines*, and those that separate hills, called *course lines*. In general, a ridge line ascends from a pass to a peak, never reaching a pit, while a course line descends from a pass to a pit, never reaching a peak. Ridge lines are the only slope lines that never reach a pit; hence, they bound dales. Similarly, course lines are the only slope lines that never reach a peak; hence, they bound hills.

For our purposes, it is more useful to combine the division into hills and the division into dales to produce a single division into regions, which I call *slope districts*, than to consider hills and dales separately. Each slope district belongs to a single hill and to a single dale. Hence, all slope lines passing through a slope district ascend to a common peak and descend to a common pit. In general, slope district boundaries each consist of four parts: a ridge line from pass to peak, followed by another ridge line from peak to pass, followed by a course line from pass to pit, followed by a final course line from pit to pass to complete the cycle. See Figure 4.2.

We now return to the initial problem, partitioning a simplified segment on the basis of radius function properties. Once again using the surface analogy, each slope district can be thought of as a mountain face together with the valley below it. At the bottom of the valley the associated boundary surfaces are "pinched" in, close to the simplified segment. As one climbs the mountain face, the associated boundary surfaces "bulge" out, each moving away symmetrically from the simplified segment until the mountaintop is reached. In a sense, each

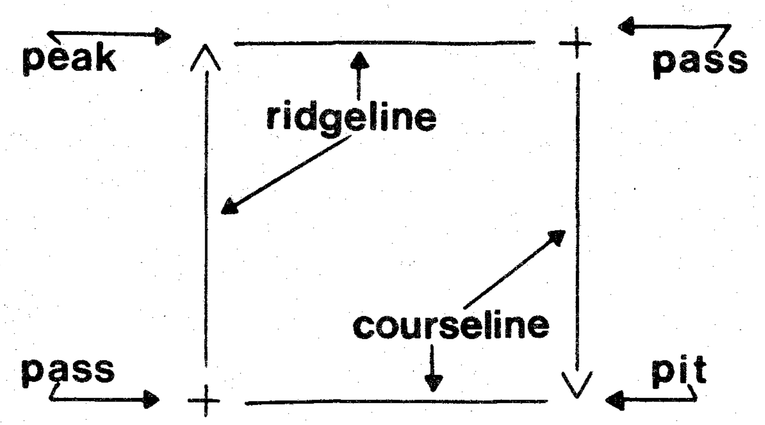


Figure 4.2.: Slope District Boundary

slope district is a region of constant first derivative behavior.

Curvature Districts. To characterize the local convexity or concavity of the mountain face we must consider second derivative behavior. This is easily accomplished by further partitioning each slope district into *curvature districts*, regions of a slope district in which the algebraic signs of the radius function Gaussian and mean curvatures are constant. Where the Gaussian curvature is positive, the mountain face is either convex or concave according to whether the mean curvature is negative or positive; where the Gaussian curvature is negative, the mountain face is neither convex nor concave, but saddle-like. Zero Gaussian curvature is intermediate between the two: in one direction the face is flat while in a perpendicular direction it is convex or concave according to the sign of the mean curvature.

4.2.2. Slope Districts

In this section, I present a more formal development of the scheme sketched in Section 4.2.1. Most likely, a rigorous development of these ideas already exists, though I have not found one. The radius function is a real-valued function on a surface, while slope lines are integral curves of the associated gradient vector field. Such situations arise frequently in physics, where the real-valued function is called a *potential function* and the vector field is called a *conservative vector field.* For example, in fluid mechanics, an incompressible fluid flow can be modeled by a two-dimensional conservative vector field. However, the emphasis is different than ours, with the result that the techniques there are slanted towards analyzing the flow itself, not the potential function associated with it.

In any case, I present here a more formal development of slope districts, striving to achieve an appropriate balance between intuition and rigor in order to expose issues not readily apparent by intuition alone; to construct a catalog of possible slope district types; to resolve difficulties arising from using slope districts on a surface with a boundary, such as the simplified segment, rather than on a surface without a boundary, such as the earth; and to provide a framework in which further work may be done.

After reviewing some notation from Chapter 3, we begin by discussing critical points of the radius function, the pits, passes, and peaks of Section 4.2.1. Then, we define slope lines as solutions of a system of differential equations and use elementary properties of such differential equations to prove simple, but useful, properties of slope lines. Under an appropriate non-degeneracy assumption, these properties, when combined with properties of the radius function near critical points, then yield an understanding of slope line behavior near critical points and a definition of ridge and course lines in terms of slope line behavior near passes. With this understanding in hand, we define slope districts as regions of the simplified segment each bounded by a cycle of alternating critical points and ridge/course lines, and then enumerate all possible slope districts.

As in Chapter 3, let S be a simplified segment in $\mathbf{R}^3$, let U be an open subset of $\mathbf{R}^2$ with coordinates u^1 and u^2, and let $\mathbf{s}(u^1, u^2) : U \rightarrow S$ be a coordinate patch on S. Further, assume[4] that $U = \mathbf{s}^{-1}(S)$. The radius function can be viewed either as a map from S to $\mathbf{R}^1$ or as a map from U to $\mathbf{R}^1$. In this chapter, as in Chapter 3, we most often take the latter view. Hence, let $r(u^1, u^2)$ denote the radius of the maximal sphere centered at $\mathbf{s}(u^1, u^2)$, let $r_i(u^1, u^2)$ denote $\frac{\partial r}{\partial u^i}(u^1, u^2)$, and let $r_{ij}(u^1, u^2)$ denote $\frac{\partial^2 r}{\partial u^i \partial u^j}(u^1, u^2)$. Where the meaning is clear from context, I drop explicit mention of (u^1, u^2).

4.2.2.1. Critical Points.

The pits, passes, and peaks of Section 4.2.1 yield important qualitative information about radius function behavior; hence, they play an essential role in the definition of slope districts. Here, we define such points in terms of properties of the first two derivatives of the radius function r and examine the geometry of the simplified segment and associated boundary surfaces near them.

Let P' denote a point in U and let P = $\mathbf{s}$(P') denote the corresponding point on S. Recall from calculus that the total derivative (see e.g. Ch. 12 of [Apostol74a]) of r at P' $\in U$ is a linear mapping from U to $\mathbf{R}^1$, $D_r(\mathrm{P}') : U \rightarrow \mathbf{R}^1$. This mapping is represented by the *gradient* vector, the 1×2 matrix $[r_1\ r_2]$. Since it will be necessary to examine the relationship between the gradient at P' and the geometry at the corresponding point P, it is useful to express $D_r(\mathrm{P}')$ with respect to a local coordinate system on S about P. Assume for the moment that the principal curvatures of the radius function, γ_1 and γ_2 (Section 3.4.1), are distinct. Then, the unit vectors in the principal directions of the radius function, $\mathbf{f}_1$

[4] Such a coordinate patch does not necessarily exist. However, it is always possible to find a collection of overlapping coordinate patches that cover S such that two overlapping patches are related by a smooth coordinate transformation. By tediously applying the chain rule or by eschewing extrinsic coordinates altogether, this assumption can be avoided. Since neither is particularly illuminating here, I make the assumption.

and $\mathbf{f}_2$, are independent (in fact, they are orthonormal) and therefore can be used as a basis for the tangent plane of S at P, $\mathrm{T_P S}$. If the principal curvatures are not distinct, choose any pair of orthonormal vectors in $\mathrm{T_P S}$ whose cross product $\mathbf{f}_1 \times \mathbf{f}_2$ is the unit normal to S at P. We would like to represent $D_r(\mathrm{P})$ with respect to the $\{\mathbf{f}_1, \mathbf{f}_2\}$ basis of $\mathrm{T_P S}$.[5] $D_r(\mathrm{P})$ is a linear mapping that maps a unit vector in the tangent plane to the directional derivative in the direction of the vector. The representation of $D_r(\mathrm{P})$ with respect to a basis is determined by the values of the directional derivatives in the direction of the basis vectors. Therefore, with respect to the $\{\mathbf{f}_1, \mathbf{f}_2\}$ basis, $D_r(\mathrm{P})$ is represented by the 1×2 matrix $[r_{\mathbf{f}_1}\ r_{\mathbf{f}_2}]$, which, by Theorem 3.1, is $-[<\mathbf{n}_b, \mathbf{f}_1>\ <\mathbf{n}_b, \mathbf{f}_2>]$.

Similarly, the second derivative of r at P′ is a bilinear mapping from $U \times U$ to $\mathbf{R}^1$, $D_r^2(\mathrm{P'}) : U \times U \rightarrow \mathbf{R}^1$. Letting $r_{ij} = \dfrac{\partial^2 r}{\partial u^i \partial u^j}$, $D_r^2(\mathrm{P'})$ is represented by the 2×2 matrix $[\, r_{ij}\,]$, often called the *Hessian* of r at P′. The Hessian too, can be expressed with respect to the $\{\mathbf{f}_1, \mathbf{f}_2\}$ basis. However, since we shall need to do so only at special points, where the form is particularly simple, we do not change bases now.

We can now make use of a well-known result[6] regarding extrema of functions defined over an open neighborhood of $\mathbf{R}^2$.

Definition 4.1: A point P is a *critical point* (or *stationary point*) of r if $D_r(\mathrm{P})$ is the zero map. A point that is not a critical point is a *regular point*. ■

Definition 4.2: A critical point of r is *non-degenerate* if the determinant of the Hessian matrix at the critical point is non-zero. Otherwise the critical point is *degenerate*. ■

[5] The derivative at a point in the domain of a map from a smooth surface to $\mathbf{R}^n$ is a linear mapping from the tangent plane of the surface to $\mathbf{R}^n$. The usual definition of the derivative of a map from $\mathbf{R}^2$ to $\mathbf{R}^n$ is a special case in which the tangent space of the surface just happens to be the surface itself. See e.g. Chapter 1, Section 2, of [Guillemin74a].

[6] Throughout Section 4.2, I state results from various sources, often changing the notation from the original and restricting the result to apply only to the situation at hand.

Using the chain rule, it is not difficult to show (see e.g. Chapter 1, Section 7 of [Guillemin74a]) that P′ is a non-degenerate critical point if and only if P is a non-degenerate critical point.

Lemma 4.1: Let P′ be a non-degenerate critical point of r. Then we have

(1) If $r_{11} > 0$ and the determinant of the Hessian is positive, r has a relative minimum at P′.

(2) If $r_{11} < 0$ and the determinant of the Hessian is positive, r has a relative maximum at P′.

(3) If the determinant of the Hessian is negative, r has a saddle point at P′.

(Theorem 13.11, [Apostol74a]) ■

After expressing the Hessian of r with respect to the $\{\mathbf{f}_1, \mathbf{f}_2\}$ basis, we can easily translate the results of Lemma 4.1 into more geometric terms.

Lemma 4.2: Let P be a critical point of r. Then, with respect to the basis $\{\mathbf{f}_1, \mathbf{f}_2\}$, the Hessian of r at P is $\begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix}$.

Proof: Since the gradient at a critical point is the zero map, $r_1 = r_2 = 0$. Then, by equation (3.9) and the definition of the Hessian, the Hessian at P is Q. Since $\mathbf{f}_1$ and $\mathbf{f}_2$ are eigenvectors of Q and γ_1 and γ_2 are the respective eigenvalues, Q is represented by $\begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix}$ with respect to $\{\mathbf{f}_1, \mathbf{f}_2\}$. ■

Theorem 4.3: Let P be a point on S, let $\mathbf{n}_b$ and $\mathbf{n}_c$ be the unit normal vectors at the associated points on the two boundary surfaces, and let K_R and H_R denote the Gaussian and mean curvatures of the radius function at P. Then:

(1) P is a critical point of r if and only if $\mathbf{n}_b$ and $\mathbf{n}_c$ are perpendicular to $T_P S$, the tangent plane to S at P.

(2) If P is a critical point of r, it is non-degenerate or degenerate according to whether K_R is non-zero or zero.

(3) If P is a non-degenerate critical point of r, then P is a local minimum (pit) if $K_R > 0$ and $H_R > 0$, a local maximum (peak) if $K_R > 0$ and $H_R < 0$, and a saddle (pass) otherwise.

Proof: Since $\mathbf{n}_s$ (the simplified segment normal at P), $\mathbf{f}_1$, and $\mathbf{f}_2$ are orthonormal, $<\mathbf{n}_s, \mathbf{n}_b>^2 + <\mathbf{n}_b, \mathbf{f}_1>^2 + <\mathbf{n}_b, \mathbf{f}_2>^2 = 1$. Therefore, $D_r(\mathrm{P})$ is the zero map whenever $<\mathbf{n}_s, \mathbf{n}_b>^2 = 1$, that is when the boundary surface normal, $\mathbf{n}_b$, is collinear with $\mathbf{n}_s$. Since $\mathbf{n}_b$ and $\mathbf{n}_c$ are reflections of each other through $T_P S$ (see Section 3.4.2), $\mathbf{n}_b$ and $\mathbf{n}_c$ are both collinear with $\mathbf{n}_s$. Thus, the first claim is proved.

Since the determinant is invariant under change of basis, by Lemma 4.2 the determinant of the Hessian is $\gamma_1\gamma_2$, which is the Gaussian curvature K_R. Thus, the second claim is proved.

Again using the invariance of the determinant, $K_R > 0$ implies $r_{11}r_{22} > r_{12}^2$, which further implies that r_{11} and r_{22} have the same sign. The trace is invariant as well. Therefore, since $2H_R = \mathrm{tr}(Q) = r_{11} + r_{22}$, the sign of H_R determines the sign of r_{11}. The last claim follows directly from the invariance of the determinant and Lemma 4.1. ■

Theorem 4.3, though not at all surprising, will be useful to us in several ways. First, it provides a simple, coordinate-system independent, geometric characterization of critical points. Second, it gives a simple test for critical point degeneracy, and, for non-degenerate critical points, it also gives simple criteria for determining whether the critical point is a pit, pass, or peak. The theorem provides little information at degenerate critical points.

Henceforth, I shall assume that all critical points of the radius function are non-degenerate.[7] This assumption vastly simplifies further analysis. Moreover, it can be shown (Chapter 1, Section 7, [Guillemin74a]) that almost any smooth perturbation of a function with degenerate critical points changes the function into one without degenerate critical points. Poston and Stewart[Poston78a] discuss the sense in which degenerate critical points are atypical.

4.2.2.2. Slope Lines

Intuitively, an ascending slope line on the simplified segment S is constructed by starting at some point on S and then taking small steps in the direction of the greatest increase in r, the direction of the gradient. Similarly, a descending slope line is constructed by taking steps in the direction opposite the gradient. These constructions can be embodied in two systems of first order nonlinear differential equations, which we then take as definitions.

Definition 4.3: Let $\mathbf{u}(t)$ denote the function $(u^1(t), u^2(t))$, let $\dot{u}^i(t)$ denote $\frac{du^i}{dt}$, and let $\dot{\mathbf{u}}(t)$ denote $(\dot{u}^1(t), \dot{u}^2(t))$. An *ascending slope line* from the point $\mathbf{u}_0 = (u_0^1, u_0^2)$ is a curve on S defined by $\mathbf{s}(\mathbf{u}(t))$, where $\mathbf{u}(t)$ is a solution to the initial value problem

$$\dot{\mathbf{u}}(t) = [r_1(\mathbf{u}(t)) \;\; r_2(\mathbf{u}(t))], \;\; \mathbf{u}(0) = \mathbf{u}_0, \; 0 \le t < \infty. \qquad (4.1)$$

[7] I have studied cursorily the effects of eliminating this assumption. However, since the results are still preliminary, I do not present them here.

A *descending slope line* from the point $\mathbf{u}_0 = (u_0^1, u_0^2)$ is a curve on S defined by $\mathbf{s}(\mathbf{u}(t))$, where $\mathbf{u}(t)$ is a solution to the initial value problem

$$\dot{\mathbf{u}}(t) = -[r_1(\mathbf{u}(t)) \;\; r_2(\mathbf{u}(t))] \,, \;\; \mathbf{u}(0) = \mathbf{u}_0 \,, \; 0 \leq t < \infty \,. \qquad (4.2)$$

A *slope line* through the point $\mathbf{u}_0$ is the intersection of U with the union of the ascending and descending slope lines from $\mathbf{u}_0$. ■

The curve in the (u^1, u^2) plane defined by a solution of an initial value problem of this type is called a *trajectory* (or an *orbit*) of the solution. For brevity I often use the phrase "slope line" rather than the more precise "trajectory that determines a slope line."

Two intuitively obvious, but nevertheless important properties of ascending and descending slope lines, set forth below in Theorems 4.4 and 4.7, follow directly from elementary properties of first order systems of differential equations. We shall use them repeatedly in the following sections. To avoid repetition, I present these properties for ascending slope lines only; they apply equally to descending slope lines.

Theorem 4.4: There is exactly one ascending slope line through each point of S.

Proof: By the existence-uniqueness theorem for solutions of initial value problems (Theorem 3, Section 4.6, [Braun75a]), there is a unique solution to (4.1) for any choice of initial point $\mathbf{u}_0$. This implies that each point of U defines a unique solution of (4.1), but not that there is only one trajectory through each point. However, the existence-uniqueness theorem of trajectories (Property 1, Section 4.6, [Braun75a]) implies that if the trajectories of two solutions pass through a single point, then the solutions are identical. Hence, there is a single trajectory through each point (u^1, u^2) in U. Since $\mathbf{s}$ is a coordinate patch, it is one-to-one. Hence, $\mathbf{s}$ maps each trajectory to a unique ascending slope line on S. ■

Definition 4.4: A point (u^1, u^2) is a *critical point* of the systems of equations (4.1) if $\dot{\mathbf{u}}(u^1, u^2) = 0$. ■

Obviously, a critical point of (4.1) is also a critical point of r.

Definition 4.5: A trajectory *reaches* a critical point P′ if there exists some $t \geq 0$ such that $\mathbf{u}(t) = \text{P}'$ or if $\lim_{t \to \infty} \mathbf{u}(t) = \text{P}'$. ■

Though a slope line may reach a critical point (in the sense of Definition 4.5), it does not necessarily contain that critical point: the slope line only approaches the critical point in the limit. Indeed, if a slope line contains a critical point, the slope line consists of that point alone, since, by (4.1), the trajectory is a constant. We now show that every ascending slope line reaches a critical point unless it first meets the simplified segment boundary.

Lemma 4.5: (Poincaré-Bendixson Theorem) Suppose that a solution $\mathbf{u}(t)$ of the system of differential equations (4.1) remains in a bounded region of the (u^1, u^2) plane that contains no critical points of (4.1). Then, its trajectory must spiral into a simple closed curve, which is itself the trajectory of a periodic solution of (4.1).[8] (Theorem 5, Section 4.8, [Braun75a]). ■

Lemma 4.6: Let $\mathbf{u}(t)$ be the trajectory of an ascending slope line. If $\mathbf{u}(t)$ is not a single point then $\mathbf{u}(t)$ is not periodic.

Proof: Assume the contrary. Then there exists t_1 and t_2, $t_1 < t_2$, such that $\mathbf{u}(t_1) = \mathbf{u}(t_2)$. Thus, $r(\mathbf{u}(t_1)) = r(\mathbf{u}(t_2))$. Hence, by Rolle's theorem (Theorem 5.1, [Apostol74a]) there is a point t_3, $t_1 < t_3 < t_2$, such that $\left.\frac{dr}{dt}\right|_{t=t_3} = 0$. By the chain rule, $\frac{dr}{dt} = r_1\dot{u}^1 + r_2\dot{u}^2$. Since $\mathbf{u}(t)$ is a solution to (4.1), $\dot{u}^i = r_i$, which implies that $\frac{dr}{dt} = r_1{}^2 + r_2{}^2$. Therefore $r_1 = r_2 = 0$ and $\mathbf{u}(t_3)$ is a critical point of r. Since the trajectory through any critical point is the point itself, either $t_1 = t_2 = t_3$ or $\mathbf{u}(t_3)$ is not on the trajectory. Both possibilities contradict the hypothesis. ■

Theorem 4.7: Every ascending slope line must either reach a critical point or must intersect the boundary of S.

Proof: Consider the trajectory $\mathbf{u}(t)$ that defines an ascending slope line. Let $\mathbf{u}_0 \in U$ be its initial point. If $\mathbf{u}_0$ is a critical point, the trajectory is the single point $\mathbf{u}_0$ and the theorem holds. Similarly, if $\mathbf{u}(t)$ is unbounded it must intersect the boundary of U since U is bounded. Therefore $\mathbf{s}(\mathbf{u}(t))$ would intersect the boundary of S, and the theorem holds. Assume neither case is true. By Lemmas 4.5 and 4.6, there is no region of U containing $\mathbf{u}(t)$, $0 \leq t < \infty$, not also containing a critical point. About each point of $\mathbf{u}(t)$ define a neighborhood of radius δ. The union of these neighborhoods contains $\mathbf{u}(t)$ and, hence, must contain a critical point. As $\delta \to 0$,the trajectory becomes arbitrarily close to a critical point. Therefore, $\mathbf{u}(t)$ reaches a critical point. ■

Thus, in this section, we have shown that through every point of S there is a unique ascending (descending) slope line. If the point is itself a critical point, then the slope line consists of that point alone. Otherwise, the slope line travels toward a critical point, reaching it in the limit as $t \to \infty$, unless it first reaches

[8] In other words, the trajectory asymptotically approaches another trajectory that is a simple closed curve.

the boundary of S.

4.2.2.3. Slope Line Behavior Near Non-degenerate Critical Points

Recall that our goal is to partition S into slope districts, regions of S bounded by an alternating "chain" of critical points and special slope lines, wherein all ascending slope lines reach a single peak and all descending slope lines reach a single pit. The next step in our programme, and the subject of this section, is to determine the behavior of slope lines near non-degenerate critical points.

In a neighborhood about a non-degenerate critical point, a local coordinate system can be defined.

Lemma[9] **4.8**: Let P′ = (0, 0) be a non-degenerate critical point of r, and let $[r_{ij}]$ be the Hessian of r at P′. Then,

$$r(u^1, u^2) = r(0,0) + \tfrac{1}{2}\sum_{i,j} r_{ij}u^i u^j, \tag{4.3}$$

near P′, where the r_{ij} are evaluated at P′ = (0, 0). (Theorem 4.2, [Poston78a]) ■

Using (4.3), we have immediately a well-known result:

Lemma 4.9: Non-degenerate critical points of r are isolated.

Proof: Taking first partial derivatives of (4.3) and setting both equal to zero, we see that in the neighborhood of P′ for which (4.3) is valid, a critical point occurs for those (u^1, u^2) that are solutions to the system of equations $[r_{ij}]\begin{bmatrix} u^1 \\ u^2 \end{bmatrix} = 0$. Since the determinant of $[r_{ij}]$ is non-zero, P′ = (0, 0) is the only solution of that system. Hence, P′ is an isolated critical point. ■

In a neighborhood of a non-degenerate critical point, we can use Lemma 4.8 to convert the nonlinear system of equations (4.1) into a linear system, thus making available the qualitative theory of linear systems of differential equations (e.g. Section 4.7 of [Braun75a]). To formulate (4.1) as a linear system near a critical point P′, first translate the coordinate system so P′ = (0, 0). Then, take

[9] A more powerful version of this result is known as the Morse Lemma. This result appears as an intermediate step in Poston and Stewart's[Poston78a] proof of the Morse Lemma.

first partial derivatives of (4.3), obtaining

$$[r_1(u^1, u^2) \;\; r_2(u^1, u^2)] = \begin{bmatrix} r_{11} & r_{12} \\ r_{12} & r_{22} \end{bmatrix} \begin{bmatrix} u^1 \\ u^2 \end{bmatrix}.$$

Therefore (4.1) becomes the linear system

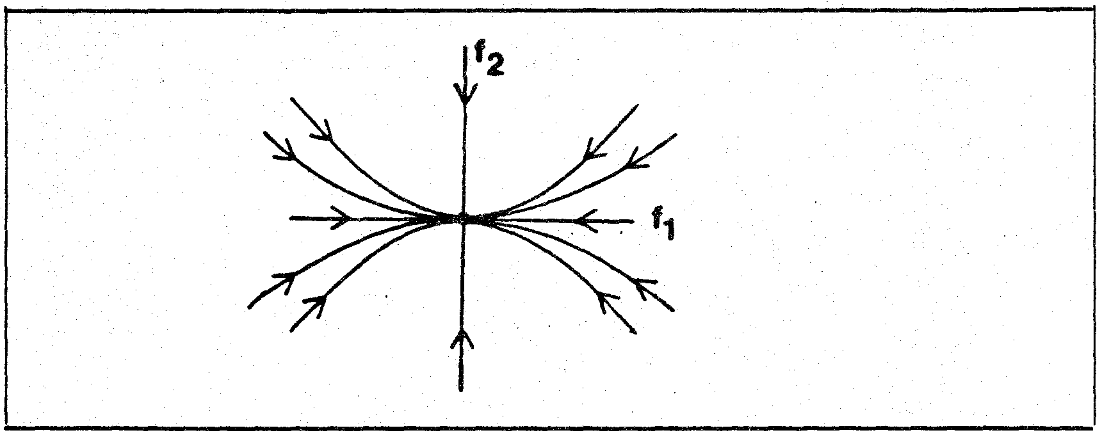
$$\dot{\mathbf{u}}(t) = [r_{ij}] \begin{bmatrix} u^1 \\ u^2 \end{bmatrix}, \; \mathbf{u}(0) = \mathbf{u}_0, \; 0 \le t < \infty .$$

Similarly, the nonlinear system (4.2) can be reformulated as a linear system near non-degenerate critical points.

The qualitative behavior of the linear system is completely determined by the signs of the eigenvalues of its matrix. Since the eigenvalue signs are invariant under change of basis, by Lemma 4.2 the behavior near a critical point is determined by the radius function principal curvatures γ_1 and γ_2, where $\gamma_1 \le \gamma_2$ (Section 3.4.1). I summarize the relevant results, illustrating the behavior of ascending slope lines.

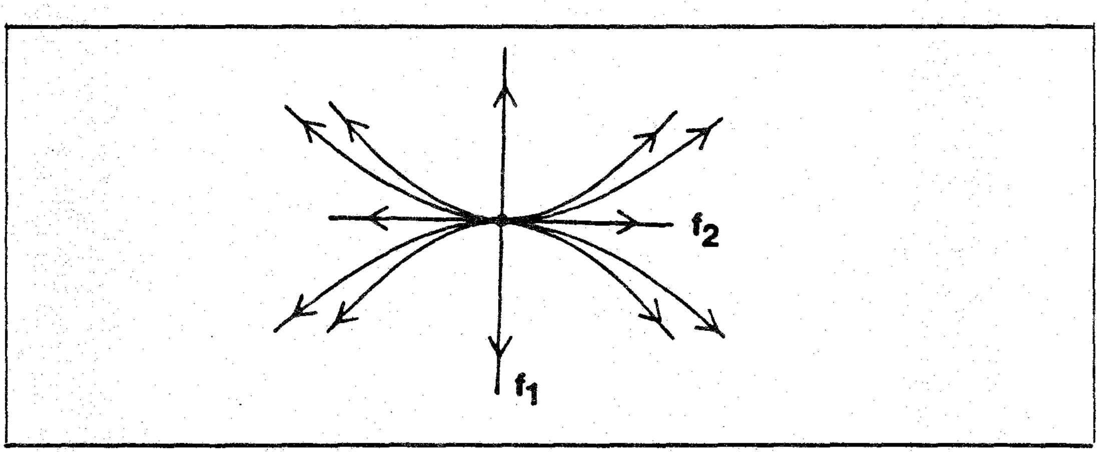
(1) $\gamma_2 < \gamma_1 < 0$: peak

All slope lines approach the critical point as $t \to \infty$. For all but two slope lines, the slope line tangent approaches the direction of $\pm\mathbf{f}_1$ as $t \to \infty$. The other two slope lines are tangent to $\pm\mathbf{f}_2$.



(2) $0 < \gamma_2 < \gamma_1$: pit

All slope lines move away from the critical point as t approaches infinity. For all but two slope lines, the slope line tangent near the critical point is in the direction of ±$\mathbf{f}_2$. The other two slope lines are tangent to ±$\mathbf{f}_1$.



(3) $\gamma_1 = \gamma_2 < 0$: peak

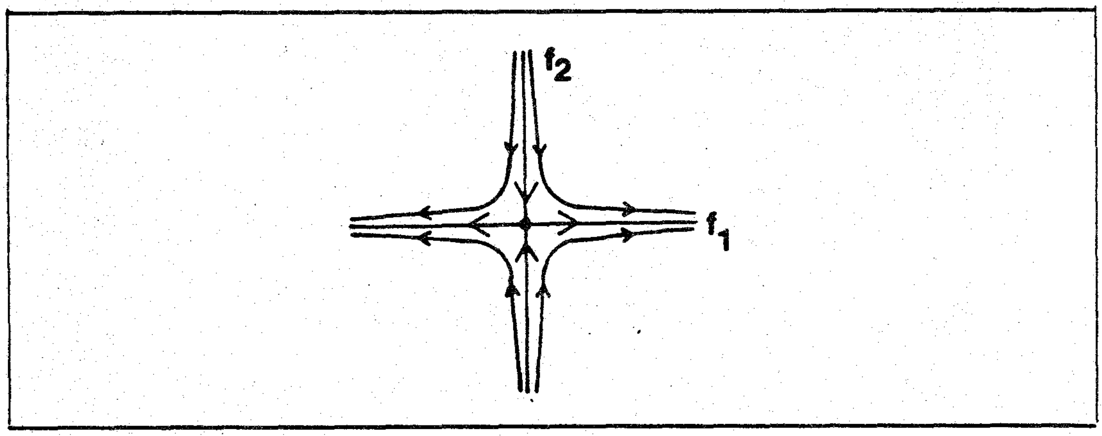
Slope lines approach the critical point from all directions.[10]

(4) $\gamma_1 = \gamma_2 > 0$: pit

Slope lines move away from the critical point in all directions.

(5) $\gamma_2 < 0 < \gamma_1$: pass

All slope lines but the two in the ±$\mathbf{f}_2$ directions move away from the critical point approaching the slope lines in the ±$\mathbf{f}_1$ directions as $t \to \infty$. The slope lines in the ±$\mathbf{f}_2$ directions approach the critical point as $t \to \infty$.



[10] When the eigenvalues of a first order linear system are equal, the system can also exhibit a different type of behavior, the so-called improper node. An improper node occurs only when the matrix of the system has exactly one linearly independent eigenvector. Here, the matrix of the system is symmetric with non-zero determinant and thus has two independent eigenvectors. Hence, there are no improper nodes.

Descending slope lines behave identically but with all traversal directions reversed.

4.2.2.4. Ridge and Course Lines

Using properties of critical points and slope lines developed in preceding sections, we now discuss ridge and course lines, those special slope lines that form the boundaries of slope districts. In Section 4.2.1, ridge lines were characterized as those slope lines that ascend from a pass to a peak, never reaching a pit, while course lines were characterized as those slope lines that descend from a pass to a pit, never reaching a peak. Though these characterizations could, with suitable added precision, be taken as definitions, they do not provide local criteria for determining whether a slope line is a ridge or course line. Instead, we define ridge and course lines locally, in terms of the behavior of slope lines in neighborhoods of passes, as discussed in Section 4.2.2.3. The local definition has the advantage that it is not necessary to traverse an entire slope line to ascertain whether it is a ridge or course line; one need only confirm that it satisfies the definition near a pass.

Definition 4.6: Let P' be a pass and let $\mathbf{u}_0$ be a point in a neighborhood of P' such that the descending slope line from $\mathbf{u}_0$ reaches P'. Then, the slope line through $\mathbf{u}_0$ is called a *ridge line.* ■

Definition 4.7: Let P' be a pass and let $\mathbf{u}_0$ be a point in a neighborhood of P' such that the ascending slope line from $\mathbf{u}_0$ reaches P'. Then, the slope line through $\mathbf{u}_0$ is called a *course line.* ■

It is easy to see from the diagram of slope line behavior near passes (page 71) that exactly two ridge lines and two course lines reach every pass, and furthermore, that ridge lines and course lines emanate from a pass in the principal directions of the radius function. This behavior is illustrated in Figure 4.3 for a surface defined by a height function with a pass at the point marked.

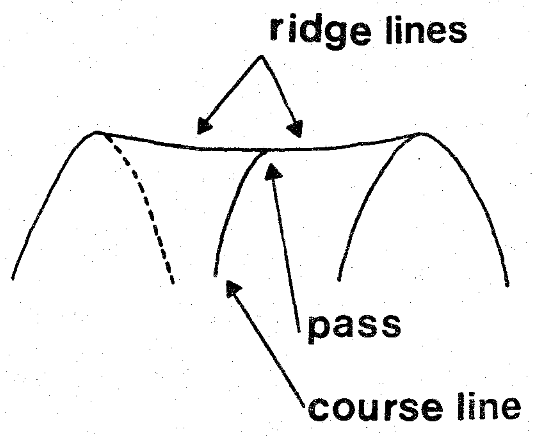


Figure 4.3.: Ridge and Course Lines at a Pass

Using properties of slope line behavior near pits and peaks, we determine some aspects of the global behavior of ridge and course lines.

Theorem 4.10: Let P′ be a pass. Then, each ridge line emanating from P′ either

(1) intersects the boundary of U;

(2) reaches a peak; or

(3) reaches a pass other than P′ along the $\pm\mathbf{f}_2$ direction of the pass that is reached.

Proof: The ridge line consists of two parts, the ascending and descending slope lines. By definition, the descending slope line reaches P′. By Theorem 4.7, either case (1) obtains or the ascending slope line reaches a critical point. Further, since all ascending slope lines diverge from pits, the critical point must be a peak or a pass. If the former, we are done; assume the latter. The ascending slope line must reach the pass along the $\pm\mathbf{f}_2$ direction, for all ascending slope lines along other directions diverge from the pass. Moreover, by Lemma 4.6[11] the ascending slope line cannot reach P′. ■

By similar reasoning we obtain analogous results for course lines:

Theorem 4.11: Let P′ be a pass. Then, each course line emanating from P′ either

(1) intersects the boundary of U;

(2) reaches a pit; or

(3) reaches a pass other than P′ along the $\pm\mathbf{f}_1$ direction of the pass that is reached. □

[11] Strictly, the proof of Lemma 4.6 does not apply here since the hypothesis of Rolle's theorem is not met. However, the mean value theorem can be used instead to show that as the ridge line approaches P′ from both directions (i.e., ascending and descending), $\frac{dr}{dt} \to 0$ for some point on the ridge line.

4.2.2.5. Non-degenerate Critical Point Configurations

We come now to the heart of our discussion of slope districts. We represent the configuration of pits, passes, peaks, course lines, and ridge lines on the simplified segment as a graph, called the *critical point configuration graph*. Pits, passes, and peaks comprise the graph vertex set; course and ridge lines comprise the edge set. By convention, we also add to the vertex set points of intersection between a ridge or course line and the simplified segment boundary. Since ridge and course lines meet only at critical points and since critical points are isolated, the critical point configuration graph is a plane graph.[12] It partitions the simplified segment into regions, some bounded by a cycle of the graph and some bounded by a path in the graph together with a portion of the simplified segment boundary. We shall show that these regions are the slope districts we seek.

We adopt the following conventions in our illustrations of critical point configuration graphs:

(1) Pits, passes, and peaks are denoted by ∇, $+$, and Δ respectively. A subscript is occasionally used for easy reference in the text.

(2) Arrows on edges indicate the ascending direction.

Let G_S denote the critical point configuration graph of the simplified segment S.

Definition 4.8: A *slope district* is a maximal subset, D, of S, such that any two points in D may be joined by a curve in D not intersecting any edge or vertex of G_S. ■

A slope district bounded by a cycle of G_S is called an *interior slope district*, while a slope district bounded by a path in G_S together with a portion of the boundary of S is called an *exterior slope district*. In the remainder of Section 4.2.2.5 we investigate properties of both interior and exterior slope districts.

[12] A plane graph is a graph embedded in a surface such that edges intersect only at vertices.

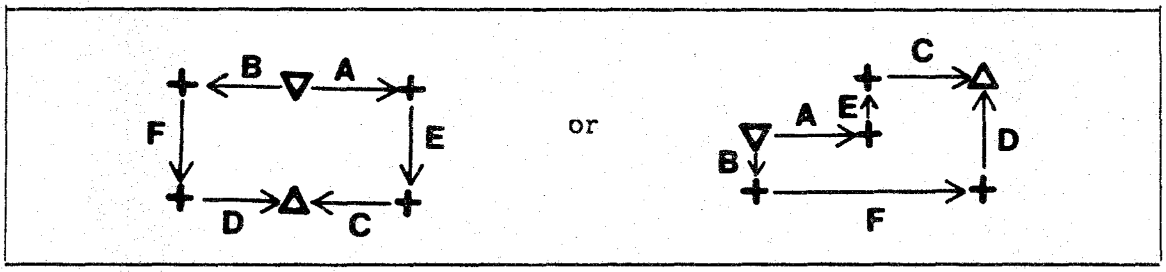
The key result of the section yields a catalog of possible interior slope districts. From the catalog, it is trivial to deduce that all slope lines through points in an interior slope district do indeed ascend to a common peak and descend to a common pit. Furthermore, the catalog yields simple constraints on adjacency relationships between slope districts.

Interior Slope Districts. Let $B(D)$ denote the cycle of G_S that bounds an interior slope district D. Since $B(D)$ is a cycle of G_S, we can traverse it in some direction, say clockwise. As we do so, the vertex–edge–vertex triples encountered are limited, by Theorems 4.10 and 4.11, to the following:

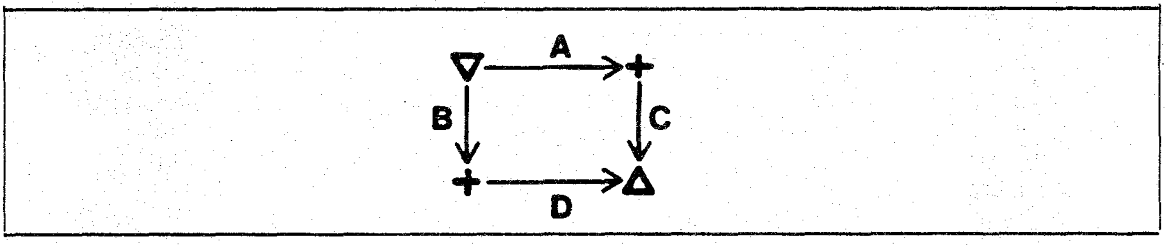
A) $\nabla \rightarrow +$
B) $+ \leftarrow \nabla$
C) $+ \rightarrow \Delta$
D) $\Delta \leftarrow +$
E) $+ \rightarrow +$
F) $+ \leftarrow +$

Therefore, any cycle of G_S can be constructed by juxtaposing these triples: the clockwise-most vertex of one triple and the counterclockwise-most vertex of the next triple must be identical. Thus, for example, the sequence ACDB might exist; ADDB cannot.

Another constraint exists as well. Before we discuss it, some additional notation is needed. Recall that exactly two ridge lines and two course lines emanate from each pass. When a pass is reached along a ridge line while traversing a cycle of the critical point configuration graph, the next edge of the cycle is either the one remaining ridge line or one of the two course lines. Similarly, when a pass is reached along a course line, the next edge is either the one remaining course line or one of the two ridge lines. Thus, to specify a cycle, more information is required than just a sequence of triples. For example, if we represent schematically the choice of the next edge to be traversed at a pass as either proceeding forward or making a left or right turn, AECDFB could represent either



Resolving the ambiguity by using the subscripts L and R to denote left and right turns, the cycles in the preceding example can be specified by $A_RE_RCD_RF_RB$ and $A_LE_RCD_RF_RB$, respectively. Obviously, two strings denote identical cycles when one string can be transformed into the other by one or more "rotates." For example, the strings A_RCD_RB, CD_RBA_R, D_RBA_RC, and BA_RCD_R all represent the same cycle:



Since course lines are always perpendicular to ridge lines, certain triples cannot be juxtaposed unless a "turn" is interposed between them. For example, the sequence AC cannot occur because the course line leading to the pass (A) is perpendicular to both of the ridge lines emanating from the pass (C). Therefore, A can be juxtaposed with C only if one of the subscripts L or R is interposed between them: A_LC or A_RC.

Cayley[Cayley59a], Maxwell[Maxwell70a], and Pfaltz[Pfaltz76a, Pfaltz78a] each make the assumption that triples of forms E and F do not occur. Now, I too make this assumption since it simplifies further analysis. I claim that any configuration containing either triple is unstable and therefore not likely to occur in practice. Consider a slope line that is both a ridge line and a course line, one that ascends to one pass and descends to another. From our discussion of the behavior of slope lines near passes, we know that there are exactly two slope lines that ascend to a pass and two that ascend to the pass; all others come near to the pass, never reaching it, and then move away. If either of the triples E or F occurs, the slope line between the two passes must be one of the two slope lines that ascends from one pass and one of the two slope lines that descends to second pass. Hence, a small perturbation of the radius function near either pass eliminates the triple from the configuration.

Figure 4.4 shows all possible juxtapositions of two triples assuming that triples E and F do not occur. As shown, a clockwise traversal of a triple begins with

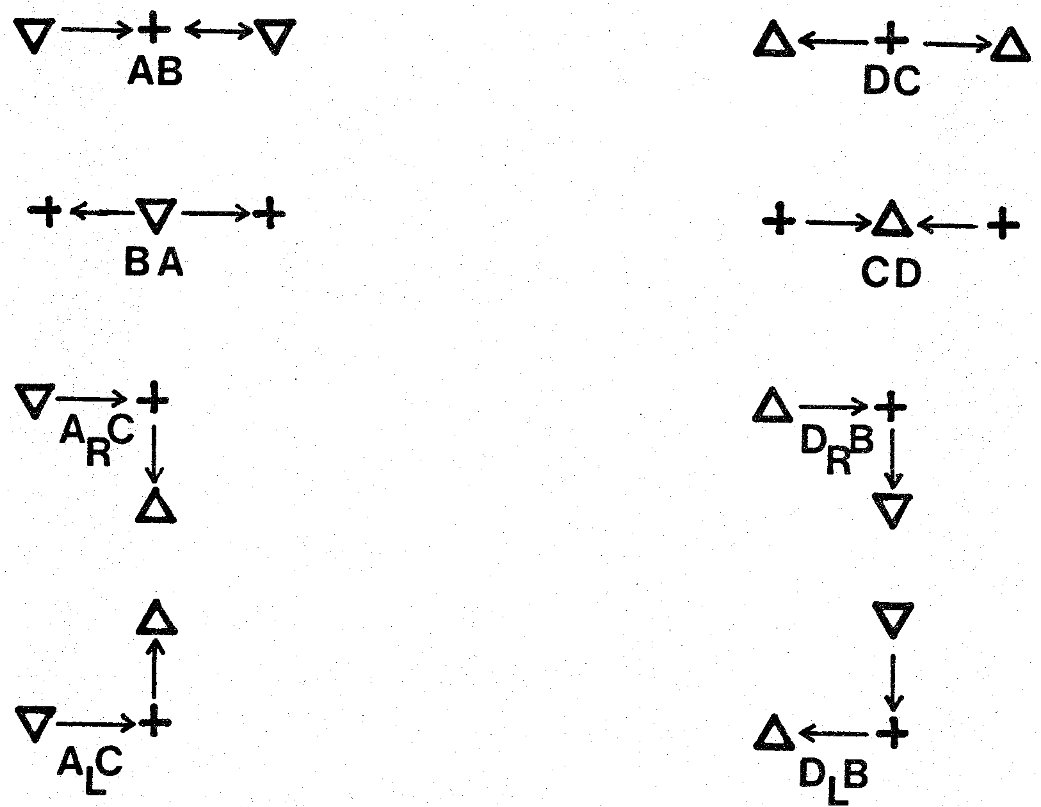


Figure 4.4.: Valid Triple Juxtapositions

the leftmost critical point.

Under this assumption, I build a catalog of interior slope districts by investigating cycles of the critical point configuration graph. My strategy is to specify concisely all possible cycles and then to show that all but three cycles contain at least one smaller subcycle. Since, by definition, any two points inside a slope district can be joined by a curve not intersecting any edge or vertex, cycles that contain subcycles cannot be slope districts. Therefore, there are only three cycles that bound interior slope districts.

I shall use regular expressions to provide a concise notation for specifying sets of strings of triples denoting paths in critical point configuration graphs. I use the following notation: parentheses denote grouping, a vertical bar (|) denotes **or**, a superscript asterisk (*) denotes zero or more repetitions of the previous symbol, and a superscript denotes a specific number of repetitions of the previous symbol. Thus $A(B|C)D^*E^2$ denotes an A followed by either a B or a C followed by any number (including zero) of D's followed by two E's. During the course of the discussion, I shall also use a question mark (?) as a symbol to denote an as yet unknown triple.

To use this regular expression notation to specify an arbitrary cycle, consider any of the equivalent sequences of triples specifying the cycle that begin with a pit. (If there is no pit, an analogous argument can be made using a peak instead.) Since the path represented by the sequence is a cycle it must also end with a pit. Therefore, the sequence must be of the form $A?^*B$ (cf. page 75). Referring to Figure 4.4, the first two triples in the sequence must therefore be either AB, A_RC, or A_LC. Similarly, the last two triples must be either AB, D_RB, or D_LB. Thus, any sequence that represents a cycle must be one of the following:

(1) AB;

(2) $(A_L C \mid A_R C)\ ?^*\ (D_L B \mid D_R B)$; or

(3) one or more of the above sequences concatenated together.

Letting *pitends* be a symbol representing any sequence both beginning and ending with a pit, we have

$$\begin{aligned} \textit{pitends} = {} & AB \\ & \mid (A_L C \mid A_R C)\ ?^*\ (D_L B \mid D_R B) \\ & \mid \textit{pitends}^*. \end{aligned}$$

Similarly, all paths beginning and ending in a peak are given by

$$\begin{aligned} \textit{peakends} = {} & DC \\ & \mid (D_L B \mid D_R B)\ ?^*\ (A_L C \mid A_R C) \\ & \mid \textit{peakends}^*. \end{aligned}$$

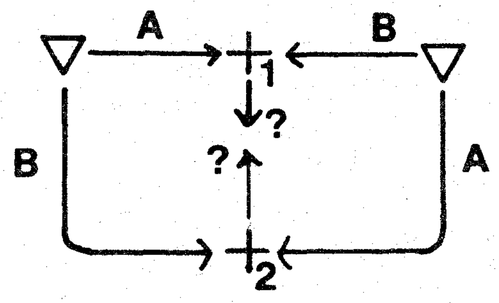
In both cases, the unknown sequence ($?^*$) is easily determined. In the former case, the sequence specified by $?^*$ must both begin and end with a peak, for else it would not "mesh" with the peaks specified by the C on its left and the D on its right. Similarly, in the latter case the corresponding sequence must both begin and end with a pit. Therefore,

$$\textit{cycle} = \textit{pitends} \mid \textit{peakends}$$

$$\begin{aligned} \textit{pitends} = {} & AB \\ & \mid (A_L C \mid A_R C)\ \textit{peakends}\ (D_L B \mid D_R B) \\ & \mid \textit{pitends}^*. \end{aligned}$$

$$\begin{aligned} \textit{peakends} = {} & DC \\ & \mid (D_L B \mid D_R B)\ \textit{pitends}\ (A_L C \mid A_R C) \\ & \mid \textit{peakends}^*. \end{aligned}$$

We now have a concise specification for all possible cycles. Using this specification I shall prove that the only cycles that are the boundaries of slope districts are AB, CD, and A_RCD_RB; all others contain one or more subcycles. First, to motivate the proof let us consider informally an example, the cycle ABAB shown below:



Since we are assuming that there are no adjacent passes, Theorem 4.10 requires that the ridge lines that emanate into the cycle, one from pass $+_1$ and one from pass $+_2$, each reach a peak. There are two alternatives. First, as illustrated on the left side of Figure 4.5, each of the ridge lines can reach a separate peak that is part of a subgraph connected to the cycle only via the ridge lines.

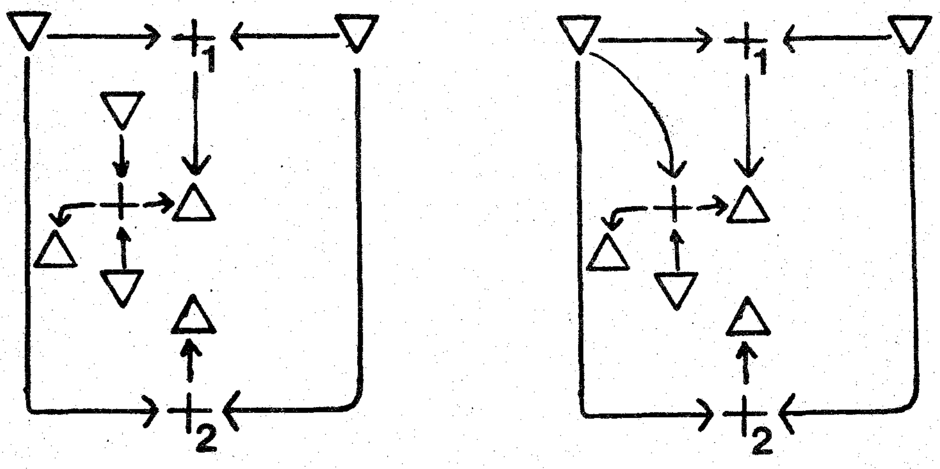


Figure 4.5.: Ridge Lines Reaching Islands

Alternatively, one or both of the ridge lines can reach a peak on a subgraph connected to the cycle via some other edge as well, as shown on the right side of Figure 4.5. In the latter case, a subcycle would be formed, thus dividing ABAB into two subcycles. Therefore, each ridge line must reach a peak on a subgraph connected to the cycle via the ridge line only. But, I shall show below that such behavior is precluded by a topological constraint, thus implying that ABAB is divided into subcycles.

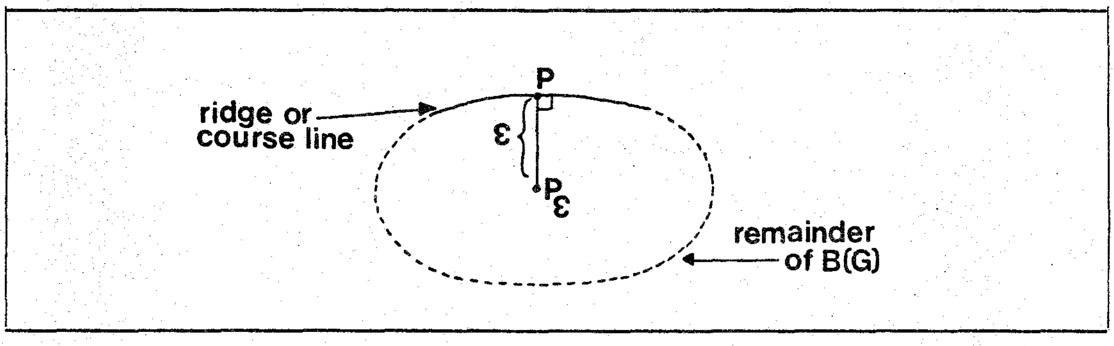
Lemma 4.13, below, provides the aforementioned topological constraint, constraining the number and type of critical points that can be contained within a cycle. After proving the lemma, I shall use the constraint it provides to prove that no cycles other than AB, CD, and A_RCD_RB can bound a slope district.

Lemma 4.12: Let there be a smooth closed curve in $\mathbf{R}^2$ having no critical points of r on it. The set of points on the curve at which the directional derivative of r along the outward directed normal to the curve is negative is called the *negative boundary* of the region enclosed by the curve. Let n_∇, n_+, and n_Δ denote respectively the number of pits, passes, and peaks of r in the region enclosed by the curve, and let $n_\downarrow$ and $n_\uparrow$ denote respectively the number of minima and maxima of r along the curve that occur on the negative boundary. Then, $n_\nabla + n_\downarrow - n_+ - n_\uparrow + n_\Delta = 1$. (Theorem 10, [Morse34a]) ■

Definition 4.9: Let G be a critical point configuration graph that contains no adjacent passes. The *Morse number* of G, denoted M(G), is the sum of the number of pits and peaks in the vertex set of G less the number of passes in the vertex set of G. ■

Lemma 4.13: Let G be a critical point configuration graph that contains no adjacent passes and is bounded by a cycle B(G). Further, let n_{AB}, n_{BA}, n_{A_LC}, and n_{D_LB} denote respectively the number of occurrences in B(G) of juxtaposed triples AB, BA, A_LC, and D_LB. Then, $M(G) = 1 - n_{BA} + n_{AB} + n_{A_LC} + n_{D_LB}$.

Proof: We prove this result by applying Lemma 4.12 to a curve "just inside" the cycle B(G). We construct a smooth closed curve inside B(G) satisfying the hypothesis of Lemma 4.12 by constructing a curve parallel to each individual ridge/course line and then joining those curves near critical points. At each regular (non-critical) point p on a ridge/course line there is a vector normal to the ridge/course line at p that points into the region bounded by B(G). We define a point p_ε on α corresponding to p as the point at distance ε from p along the inward directed normal at p:

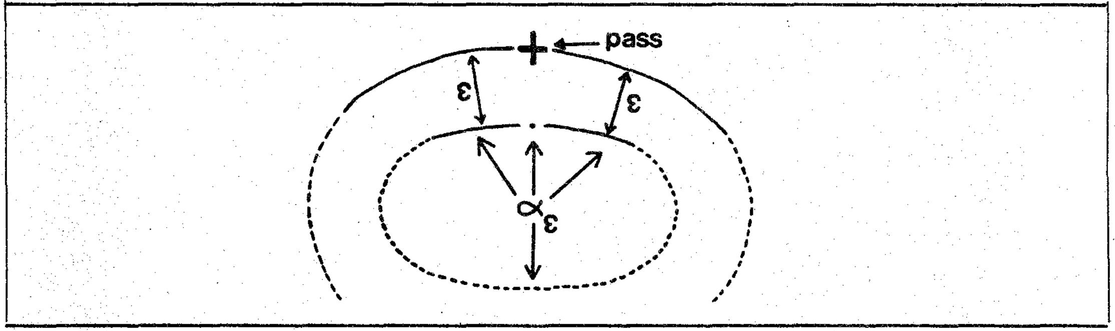


Thus, for any given value of ε there is a curve parallel to each ridge/course line in the cycle B(G). The curve α_ε is constructed by smoothly joining these curves near critical points. There are two cases:

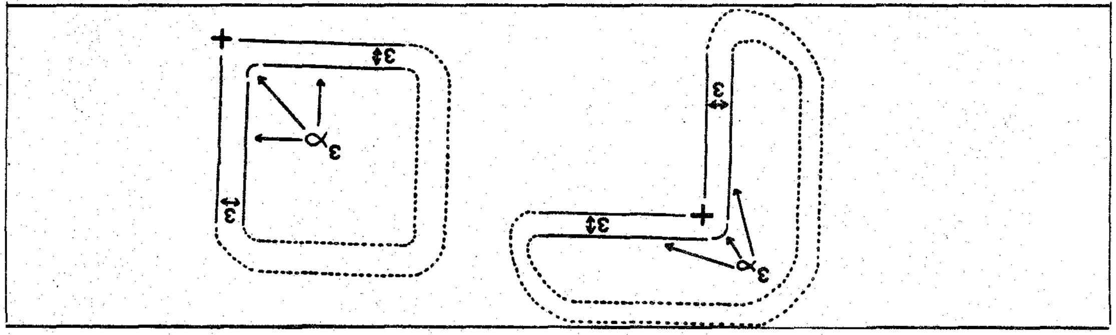
(1) Two ridge/course lines meet at a critical point so that each of the two ridge/course lines has the same tangent line at the critical point.

(2) Two ridge/course lines meet at a critical point so that each of the two ridge/course lines have different tangent lines at the critical point.

In the first case, since there is a well-defined tangent at the critical point, the two parallel curves can be joined smoothly by adding the point at distance ε along the inward directed normal at the critical point:



In the second case, since there is no well-defined tangent at the critical point, we must terminate each of the two parallel curves before they reach the critical point and then smoothly join them with some "splicing" curve. Other than smoothness, the only requirement placed on the "splicing" curve is that it can be made arbitrarily near the critical point. We shall use a circular arc, as illustrated in the two examples below:



To apply Lemma 4.12, we must determine where the extrema of r along α_ε occur. Points where r achieves a local extremum are called *extreme points*, while the values assumed by r at extreme points are called *extreme values*. Recall from two-dimensional calculus that a function along a smooth curve can have a local minimum or maximum at a point on the curve only if the directional derivative of the function in the direction of the curve at that point is zero.

Consider traversing both B(G) and α_ε clockwise. As ε is made to approach zero, α_ε approaches the cycle B(G) and the directional derivative along α_ε of r at a point p_ε on α_ε approaches the the directional derivative of r at p in the direction of traversal. Since ridge/course lines are slope lines, the directional derivative of r along them is non-zero except at critical points. Therefore, for small enough ε, the derivative of r along α_ε can be zero only near critical points.

So far we have argued that no extreme points occur on α_ε except near critical points. We now investigate at which critical points such extreme points occur and whether or not they occur on the negative boundary of the region bounded by α_ε. Again, traverse B(G) and α_ε clockwise. As B(G) is traversed, a peak is crossed by climbing a ridge line, reaching the peak, and then descending another ridge line. Hence, as the corresponding portion of α_ε is traversed, the derivative of r along α_ε is positive at points corresponding to the first ridge line and negative at points corresponding to the second ridge line. Therefore, if the two pieces of α_ε that correspond to the two ridge lines are joined by a single point, the derivative must be zero at that point; if the two pieces are joined by a circular arc, the derivative must be zero somewhere along the arc. In either case, the behavior of the radius function near the peak guarantees that r achieves a local maximum, rather than a minimum or inflection, at the point where the derivative is zero.

By similar arguments, there is a minimum near pits, a maximum near passes where two ridge lines of the cycle meet, a minimum near passes where two course lines of the cycle meet, and either no extremum or two extrema near passes where a ridge line and a course line of the cycle meet, depending upon whether the situation is as shown on the left or the right of the previous figure. Recall from the statement of Lemma 4.12 that the set of points of α_ε at which the directional derivative of r along the outward directed normal to the curve is negative is called the negative boundary of the region enclosed by α_ε. Similarly, the set of points for which the directional derivative is positive is called the *positive boundary* of α_ε. The boundary type at each of the extreme points on α_ε can be determined easily by examining the behavior of the radius function near critical points (Section 4.2.2.3). For each of the triple juxtapositions shown in Figure 4.4, Table 4.1 gives the number and type of extrema and the boundary type(s) of α_ε near the second of the three critical points.

The result now follows directly from Lemma 4.12 and the observation that since B(G) is a cycle, it consists of alternating passes and pit/peaks, and thus makes no contribution to M(G). ■

I now use Lemma 4.13 to show that no cycles other than AB, CD, and A_RCD_RB bound slope districts. I shall use the following two definitions.

Definition 4.10: A *tree-island* is a subtree of a critical point configuration graph having the property that every pass has four incident edges. ■

Informally, a tree-island is a subgraph of a critical point configuration graph that has no cycles and no "dangling edges." Figure 4.6 illustrates several

Table 4.1
Extrema of r on α_ε

Juxtaposed Triples	Extremum Type	Boundary Type
AB	maximum	negative
$A_L C$	(maximum, minimum)	(negative, positive)
$A_R C$	none	
BA	minimum	negative
CD	maximum	positive
DC	minimum	positive
$D_L B$	(minimum, maximum)	(positive, negative)
$D_R B$	none	

graphs, some of which are tree-islands and some not, as marked.

Lemma 4.14: Let G be a non-empty tree-island. Then, M(G), the Morse number of G, is $(2n_\nabla + 2n_\Delta + 1)/3$.

Proof: In any tree, the number of edges is one less than the number of vertices (Theorem 4.1, [Harary69a]). By definition, the number of edges in a tree-island is four times the number of passes. Therefore, $n_\nabla + n_\Delta + n_+ = 4n_+ + 1$. The result then follows by using this relationship to eliminate n_+ from $M(G) = n_\Delta - n_+ + n_\nabla$. ∎

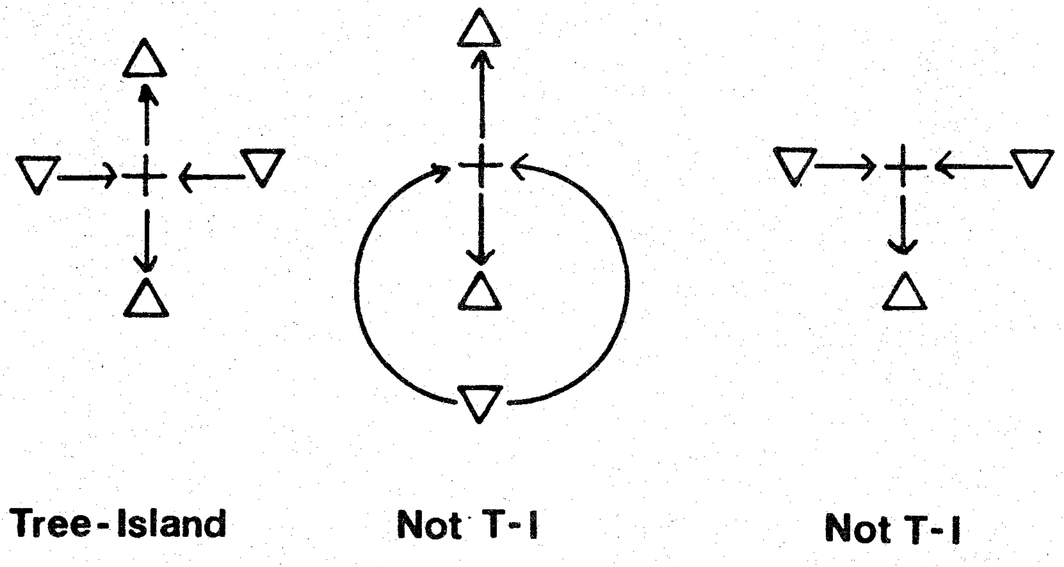


Figure 4.6.: Tree-Islands

Definition 4.11: A *tree-peninsula* is a subtree of a critical point configuration graph having the property that one pass has three incident edges and all other passes have four incident edges. ■

Informally, a tree-peninsula is a subgraph of a critical point configuration graph that has no cycles and one "dangling edge."

Lemma 4.15: Let G be a non-empty tree-peninsula. Then, M(G), the Morse number of G, is $(2n_\nabla + 2n_\Delta)/3$.

Proof: In any tree, the number of edges is one less than the number of vertices (Theorem 4.1, [Harary69a]). By definition, the number of edges in a tree-peninsula is one less than four times the number of passes. Therefore, $n_\nabla + n_\Delta + n_+ = 4n_+$. The result then follows by using this relationship to eliminate n_+ from $M(G) = n_\Delta - n_+ + n_\nabla$. ■

Definition 4.12: Let B(G) be a cycle of a critical point configuration graph and let P be a pass in B(G). The pass P is said to *emanate inward* i edges if i of the four edges defined by P are inside the cycle B(G). Let I be the sum over all passes in the cycle B(G) of the number of edges emanated inward by each pass. Then, the cycle B(G) is said to *emanate inward* I edges. ■

Lemma 4.16: Let G be a subgraph of a critical point configuration graph such that G is bounded by a cycle, B(G), that emanates inward I edges. If B(G) is the boundary of a slope district, then $M(G) \geq I$.

Proof: Consider one of the I edges that emanates into the cycle B(G). Let E denote that edge. By Theorems 4.10 and 4.11 and the assumption that there are no adjacent passes, edge E must reach either a peak or a pit. Without loss of generality, assume it reaches a peak, denoted by P; the argument below is identical for a pit. Since any edge of a graph is not on a cycle if and only if removing that edge increases the number of components of the graph (Theorem 3.2, [Harary69a]), any path in G between P and B(G) must contain E. Otherwise, G would contain a subcycle. It would then be possible to find two points in the district that are separated by the subcycle, and therefore cannot be connected without crossing G, thus contradicting the hypothesis that B(G) is the boundary of a slope district. Therefore, upon deleting edge E, G is split into two components, one connected to B(G) and not containing P and one not connected to B(G) but containing P. The latter component must be a tree for otherwise G would contain a subcycle, again contradicting the hypothesis that B(G) is the boundary of a slope district. Furthermore, since the component is not connected to B(G), it must be a tree-island. Thus, each of the I edges that emanates from B(G) into G has a tree-island "attached" to it.

Now, let us consider the Morse number of G, M(G). M(G) is the sum of the Morse numbers of B(G) and of the subgraph inside of B(G). Since B(G) consists of alternating passes and pit/peaks, it makes no contribution to M(G). By Lemma 4.14, the tree-island reached by each of the I edges emanating from B(G) into G contributes one or more to M(G). Therefore, unless some other critical points contained inside of B(G) subtract from M(G), that is, unless ignoring the aforementioned tree-islands there are more passes inside B(G) that pit/peaks, $M(G) \geq I$. But any pass contained inside B(G) must also be part of a tree. Moreover, any such tree is either not connected to B(G) or is connected by only one edge, for otherwise a cycle would exist. Thus, any pass contained inside B(G) must also be part of a tree-island or of a tree-peninsula, implying by Lemmas 4.14 and 4.15 that all the "extra" critical points contained inside B(G) make a non-negative contribution to M(G). Therefore, $M(G) \geq I$. ∎

Building upon the constraints imposed by Lemmas 4.13 and 4.16, Theorem 4.17 gives a catalog of interior slope districts and shows that slope lines through all points in an interior slope district ascend to a common peak and descend to a common pit.

Theorem 4.17: Let G be a subgraph of a critical point configuration graph and let B(G) be a cycle of G that is the boundary of an interior slope district. If there are no adjacent passes in G, i.e., if neither of the sequences E or F appear, then B(G) is equivalent to one of the following cycles:

(1) CD

(2) BA

(3) A_RCD_RB

Further, all ascending slope lines from points in the slope district reach a common peak and all descending slope lines from points in the slope district reach a common pit.

Proof: By Lemma 4.16, the Morse number of G, M(G), must be at least as large as the number of edges that emanate inward from B(G). On the other hand, Lemma 4.13 gives M(G) in terms of the number of occurrences on B(G) of AB, BA, A_LC, and D_LB. To prove the theorem, I show that for all but cycles AB, BA, and A_RCD_RB, Lemmas 4.13 and 4.16 place conflicting constraints on M(G).

Table 4.2 gives for each of the possible triple pairs (cf. Figure 4.4) the number of edges that emanate inward from that pair and its contribution to M(G) as determined by Lemma 4.13. By Lemma 4.13, M(G) is one more than the sum of the contributions given in Table 4.2. Observe from the table that no pair makes a larger contribution to M(G) than the number of edges emanated inward by that pair. Therefore, to show that a cycle does not satisfy the constraints imposed by Lemmas 4.13 and 4.16, we need only show

Table 4.2
Triple Contributions to M(G)

Type	Edges	M(G) Contribution
A_LC	2	+1
A_RC	0	0
CD	0	0
D_LB	2	+1
D_RB	0	0
BA	0	−1
AB	1	+1
DC	1	0

that for some portion of the cycle, the number of edges emanated inward exceeds its contribution to M(G) by more than one. By inspection, this is not the case for AB, CD, or A_RCD_RB.

To show that every other cycle does not satisfy the constraints, let us examine the regular expressions for a cycle. Recall that since the sequences defined by the regular expressions represent cycles, the last triple of a sequence is juxtaposed with the first triple of the sequence.

(1) Neither A_LC nor D_LB can occur in any cycle since (a) both emanate two edges but only contribute +1 to M(G), and (b) inspection shows that both occur only in cycles that contain at least one BA. Therefore, the number of edges emanated inward exceeds the contribution to M(G) by at least two.

(2) The sequence *pitends*^{x}, $x \geq 2$, cannot occur in any cycle since BA occurs at least x times in any such sequence. This can be seen by noting that *pitends* is always of the form $(A?^*B)^x$. Therefore, the number of edges emanated inward by the sequence exceeds its contribution to M(G) by at least x.

(3) The sequence D_RB *pitends* A_RC cannot occur unless *pitends* is empty since such a sequence contains at least two BA's (D_RB juxtaposed with *pitends* and *pitends* juxtaposed with A_RC). Therefore, the number of edges emanated inward by the sequence exceeds its contribution to M(G) by at least two.

(4) The sequence *peakends*^{x}, $x \geq 2$, cannot occur in any cycle. By items 1 and 3, *peakends*^{x} must be of the form (DC | *DrBArC*)^x. Therefore, there are x occurrences of DC and/or BA, implying that the number of edges emanated inward by the sequence exceeds its contribution to M(G) by at least x.

(5) The sequence A_RC *peakends* D_RB cannot occur unless *peakends* is empty. By items 1, 3, and 4, the sequence must be either A_RCDCD_RB or $A_RCD_RBA_RCD_RB$. In the former case, there is one DC and one BA, while in the latter case there are two BA's. Therefore, the number of edges emanated inward by either sequence exceeds its contribution to M(G) by at least two.[13]

Since we have examined all cases other than AB, DC, and A_RCD_RB, no cycle other than those three can bound a slope district.

That all slope lines through points in side one of these three possible slope districts ascend to a single peak and descend to a single pit is obvious from inspection. See Figure 4.7. ■

We have thus shown that all interior slope districts have one of the configurations shown in Figure 4.7.

[13] Though my treatment of these cases is asymmetric, it is not difficult to prove that a relationship similar to that of Lemma 4.13 obtains by replacing n_{BA} with n_{CD} and n_{AB} with n_{DC}. Using that relationship, the proof of Theorem 4.17 can be made symmetric.

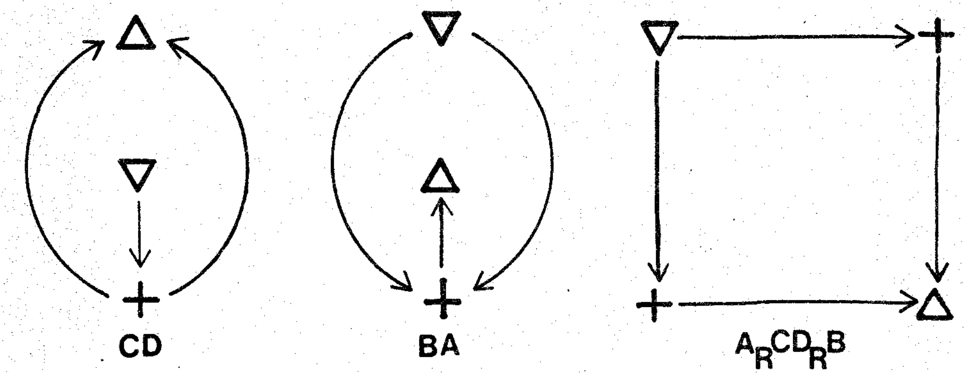


Figure 4.7.: Catalog of Interior Slope Districts

Exterior Slope Districts. We must still examine exterior slope districts. However, let us first introduce an additional problem which we shall see is closely related to exterior slope districts. Consider the height function of the surface shown in Figure 4.8. There are no critical points. Yet, it seems intuitively clear

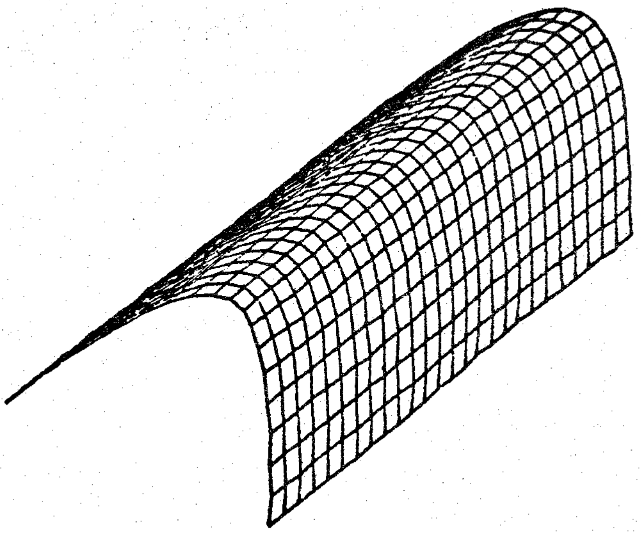


Figure 4.8.: Height Function With No Critical Points

that there should be two slope districts separated by a ridge line, not a single slope district consisting of the entire simplified segment. Even when a critical point is introduced, such as by stretching the surface, as shown in Figure 4.9, there is still but one slope district.

Both this problem and exterior slope districts have the same origin. Were the radius function defined on a closed surface, such as a sphere, rather than on a surface with boundary, such as the simplified segment, certain properties of the radius function (discussed below) would demand the existence of a pass and its associated ridge and course lines. Thus, one can consider the aforementioned problem to be caused by the intervention of the simplified segment boundary between actual critical points, if any, and critical points that would otherwise occur. Figure 4.10 illustrates an exterior slope district in which the simplified segment boundary intervenes between one of the passes in a A_RCD_RB slope district and the remainder of the district.

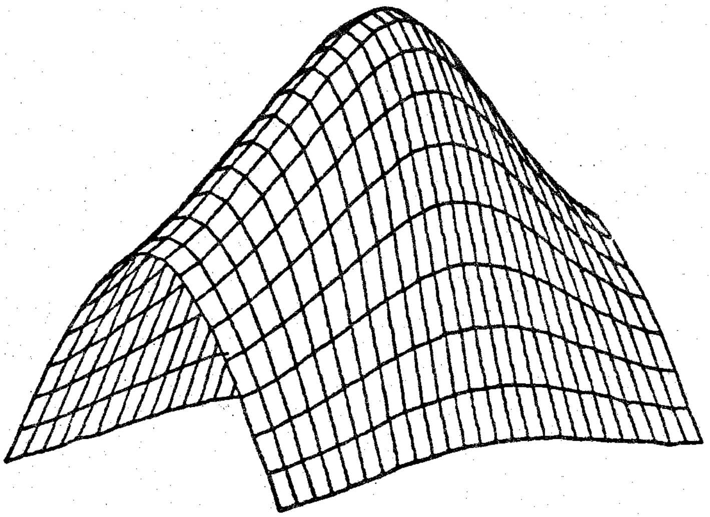


Figure 4.9.: Height Function With One Critical Point

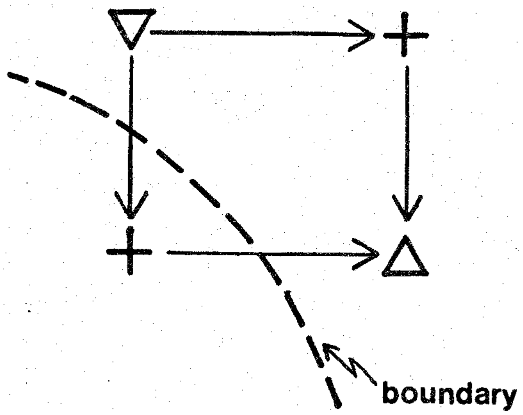


Figure 4.10.: Boundary Intervention

The solution to the problem is simple: find points where ridge and course lines emanating from a "missing" pass would have crossed the simplified segment boundary. Once such points are found, the "missing" ridge and course lines are determined; they are the ascending or descending slope lines from the boundary points. To find these points, we must characterize ridge and course lines independently of the pass from which they emanate. Here, I sketch such a characterization of ridge lines; analogous arguments apply to course lines.

A *curvature line* is a curve on the simplified segment whose tangent vector at each point is a principal direction of the radius function at that point. We have seen (page 71) that a ridge line leaves a pass in a principal direction, that is, the ridge line "starts out" as a curvature line as well as a slope line. Similarly, a ridge line reaches a peak tangent to one of the principal directions at the peak. I claim that a ridge line is also a curvature line along the maximum principal direction; the converse is not always true.

By definition, the ridge line proceeds in the direction of the gradient, the direction in which the first directional derivative of the radius function (Section 3.4.1) is largest. The second directional derivative of the radius function in some direction measures the rate of change of the first directional derivative of the radius function in the same direction. Recall that the principal directions of the radius function are the directions of the minimum and maximum radius function second directional derivatives. Assume that the direction of the gradient and of the maximum principal direction are different. Then, since the second directional derivative is smaller in the gradient direction than in the maximum principal direction, the first directional derivative increases more rapidly (or decreases less rapidly) in the principal direction than in the gradient direction. Therefore, the gradient direction approaches the maximum principal direction as the ridge line is traversed in the ascending direction. By the same argument, once the gradient direction coincides with the maximum principal direction, the two directions never part. From our discussion of slope line behavior near critical points, we know that near passes, ridge lines coincide with curvature lines. Therefore, ridge lines are also curvature lines along the direction of the maximum principal direction; analogous arguments show that course lines are also curvature lines along the direction of the minimum principal direction.

Unfortunately, the converse is not true: a slope line that is simultaneously a curvature line in the maximum principal direction is not necessarily a ridge line. However, it is still useful to examine the behavior of such a slope line. As the slope line is traversed in the descending direction, it reaches a critical point in the maximum principal direction. By examining the diagrams of slope line behavior near critical points (Section 4.2.2.3), it can be seen that such behavior only occurs in two cases: either the slope line descends to a pass, in which case it is a ridge line, or it is one of the two slope lines that reaches a pit along the maximum principal direction. I know of no way to disambiguate the two cases.

Even so, we introduce a "fake" pass, called a *boundary pass*, at each point on the simplified segment boundary where a principal direction coincides with the gradient direction, but no true ridge or course line crosses. When the principal direction is a maximum principal direction, the slope/curvature line through the boundary pass can be treated as a ridge line; otherwise, it can be treated as a course line. Using this criterion, a boundary pass and its associated ridge/course line may be introduced where none belongs. Such an error results in falsely partitioning a slope district into several slope districts. This error of commission is preferable to the corresponding error of omission, for omitting a ridge or course line where one belongs can cause significant "features" of the radius function behavior to be ignored.

Consider, for example, the height functions illustrated in Figures 4.8 and 4.9. In the first case, a single boundary pass and associated ridge line would be introduced as shown in Figure 4.11, thus creating two slope districts, as intuition demands. Similarly, in the second case, two boundary passes are introduced, each of which emanates a ridge line to the single peak as shown in Figure 4.12. Failing to introduce these boundary passes would, in each case, ignore the most striking qualitative property of the height function.

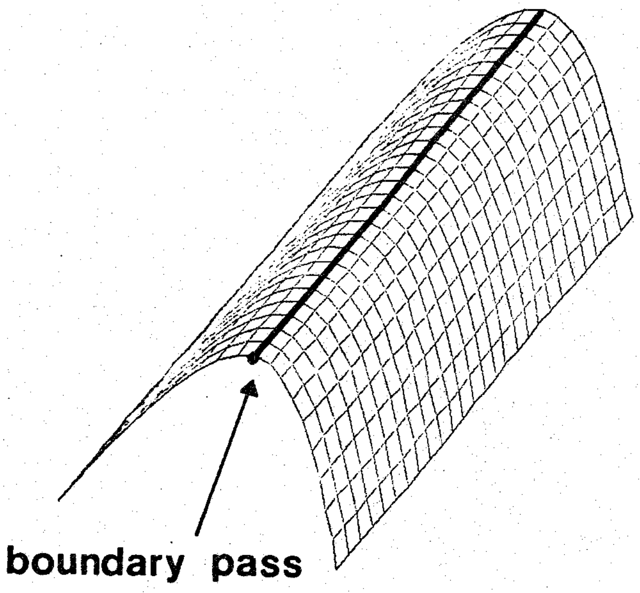


Figure 4.11.: Introduction of a Boundary Pass

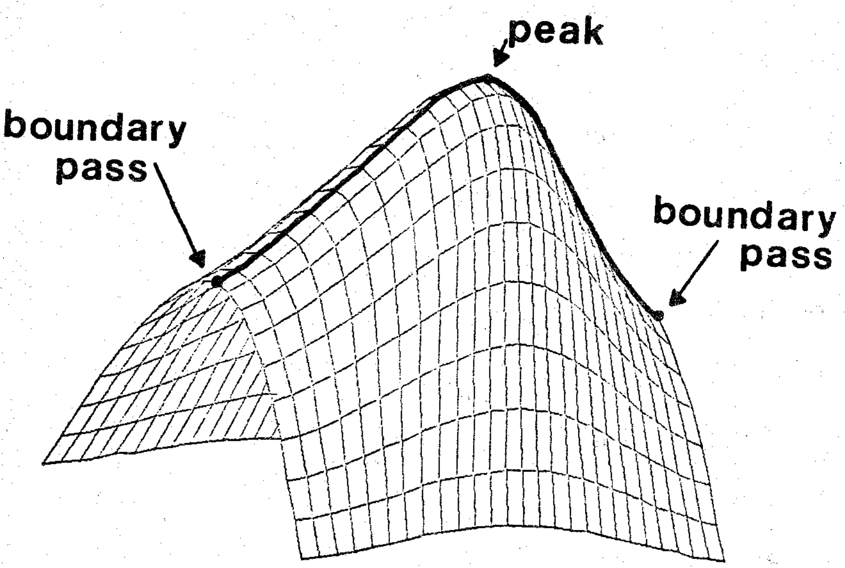


Figure 4.12.: Introduction of Two Boundary Passes

It does not seem possible to enumerate straightforwardly all possible exterior slope districts because the simplified segment boundary can cut out an arbitrarily complicated portion of a slope district configuration. This is illustrated in Figure 4.13.

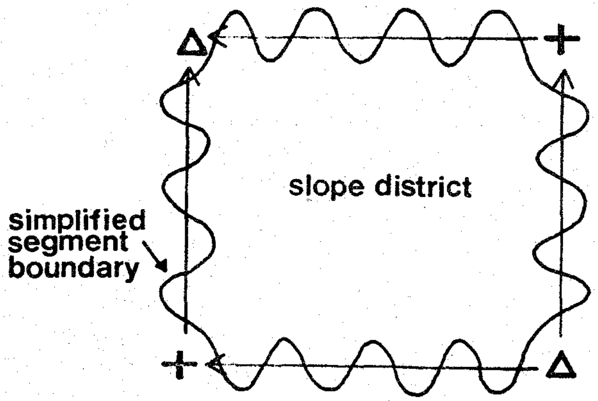


Figure 4.13.: An Arbitrarily Complicated Exterior Slope District

4.2.3. Curvature Districts

The slope districts described in Section 4.2.2 are derived primarily from first derivative properties of the radius function. In this section, we partition the simplified segment into regions, called *curvature districts*, derived from second derivative behavior as characterized by radius function Gaussian and mean curvatures. The two partitions are not independent, for as we have seen, the Gaussian and mean curvatures play an important role in defining slope districts as well. Depending upon the application, it might be appropriate to partition the simplified segment into slope districts alone, curvature districts alone, or both simultaneously. Since my intuition (not confirmed by any evidence) is that for most applications it is appropriate to use slope and curvature districts simultaneously, I shall discuss their interdependence below.

The algebraic signs of the Gaussian and mean curvatures qualitatively characterize the second derivative behavior of the radius function. We might therefore partition a simplified segment into regions wherein the signs of both the Gaussian and mean curvatures remain constant. However when the Gaussian curvature is negative, the sign of the mean curvature has little meaning, indicating only the relative magnitudes of the two principal curvatures. Coalescing the three cases of negative Gaussian curvature into one, we obtain the six curvature district types shown in Table 4.3.

Table 4.3
Curvature District Types

Name	Gaussian Curvature	Mean Curvature
Flat	0	0
Parabolic convex	0	−
Parabolic concave	0	+
Saddle	−	−, 0, +
Convex	+	−
Concave	+	+

Definition 4.13: A *curvature district* is a maximal open subset of a simplified segment such that the signs of the Gaussian and mean curvatures at all points in the subset are of the same type, as determined by Table 4.3. ■

As noted above, the partition into slope districts and the partition into curvature districts are not independent. Each pit must lie within a concave curvature district, each peak within a convex curvature district, and each pass within a saddle curvature district. Thus, for example, the slope district configuration $A_R CD_R B$ would be split into at least four curvature districts as shown in Figure 4.14. The boundary between the concave and convex curvature districts might also be a combination of flat, parabolic convex, and parabolic concave curvature districts.

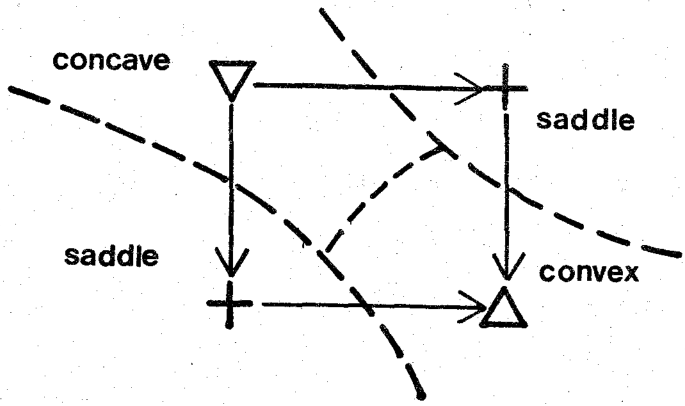


Figure 4.14.: Curvature District Partition of Slope District

4.3. Axis Primitives

Using simplified segment Gaussian and mean curvatures, the simplified segment and its associated boundary surfaces can be partitioned into a collection of two-sided *axis primitives*. Axis primitives should be defined so that the partitioning of the simplified segment is invariant under rigid motions of the figure in space. This can readily be accomplished by partitioning the simplified segment into curvature districts as defined in Section 4.2.3, using simplified segment curvatures in place of the corresponding radius curvatures. Table 4.3 defines the six possible axis primitives as well as the possible radius curvature districts. However, there is an ambiguity here.

The sign of the mean curvature is determined by the signs of the principal curvatures. For the radius function, the signs of the principal curvatures are well determined: positive means increasing, negative decreasing. On the other hand, the signs of the simplified segment principal curvatures depend upon the direction of the unit normal vector at each point of the simplified segment: reversing the direction of the normal changes the sign. Since the direction of the unit normal is arbitrary, so is the sign of the mean curvature. This is to be expected, as considering the sign of the curvature of a plane curve shows. That sign too is arbitrary, for if the curve is traversed in one direction the curvature is positive, while if traversed in the other direction it is negative. Thus, in three dimensions, arbitrarily choosing one of the two possible unit normal directions is equivalent to choosing a "traversal" direction in two dimensions. Therefore, at some point on the simplified segment, choose one of the two possible unit normal directions and apply that choice consistently to the whole simplified segment, so that the unit normal changes continuously.

4.4. Boundary Primitives

In Sections 4.2 and 4.3, I have defined two primitive sets, one based upon radius function properties alone and one upon simplified segment curvature alone. In this section, I define another set of primitives, *boundary primitives*, that are based on boundary surface curvatures and show that these primitives are determined by a combination of simplified segment curvatures and radius function properties. The approach is simple. The simplified segment and associated boundary surfaces are partitioned into primitives, each with the property that the algebraic signs of the Gaussian and mean curvatures are constant over each of the two boundary surfaces associated with the primitive. As with axis primitives, since the sign of the mean curvature has little meaning when the Gaussian curvature is negative, we coalesce the three cases of negative Gaussian curvature into one, yielding the same six curvature labels shown in Table 4.3. Hence, since there are two boundary surface pieces associated with each primitive, there are 36 boundary primitives, one for each possible pair of labels. These primitives are listed in Tables 4.4 and 4.5, which we shall discuss below.

In the remainder of Section 4.4, I use the curvature relationships derived in Chapter 3 to examine in two ways the relationship between boundary primitives and the curvatures of the simplified segment and radius function. I first use the curvature relationships to define four properties of the simplified segment and radius function that uniquely determine the boundary primitives. I then use those same curvature relationships to develop further intuition about the geometry of the three-dimensional symmetric axis transform.

We now examine the relationship between the 36 boundary primitives and the curvatures of the simplified segment and radius function. Let B and C be the boundary surfaces associated with the simplified segment $\mathcal{S}$, and let P be a point on $\mathcal{S}$. Recall (Section 3.4.2) that there is a surface B' parallel to B that passes through P (see Figure 3.3), and moreover, that the algebraic signs of the Gaussian and mean curvatures at the point on B associated with P, are the same as the Gaussian and mean curvatures at the corresponding point of B'. The mean and Gaussian curvatures of B' at the point associated with P are given by equations (3.3) and (3.4) respectively, repeated here for convenience:

$$h = \frac{\gamma_1(1 - r_{\mathbf{f}_2}^2) + \gamma_2(1 - r_{\mathbf{f}_1}^2)}{2<\mathbf{n}_s,\mathbf{n}_b>^2} + \frac{\lambda_1(1 - r_{\mathbf{e}_2}^2) + \lambda_2(1 - r_{\mathbf{e}_1}^2)}{2<\mathbf{n}_s,\mathbf{n}_b>} \tag{3.3}$$

$$k = \lambda_1\lambda_2 + \frac{\gamma_1\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>^2} + \frac{\lambda_1 r_{\mathbf{e}_2\mathbf{e}_2} + \lambda_2 r_{\mathbf{e}_1\mathbf{e}_1}}{<\mathbf{n}_s,\mathbf{n}_b>}. \tag{3.4}$$

Since the signs of the curvatures at corresponding points on B on B' are identical, we can use (3.3) and (3.4) to determine the signs of the curvatures at points on B. Likewise, when the sign of $<\mathbf{n}_s,\mathbf{n}_b>$ is changed, these same equations determine the signs of the curvatures at the corresponding point on the other boundary surface, C.

We split each of these equations into the sum of two terms, one that is the same for both boundary surfaces and one that has the same magnitude for both boundary surfaces but opposite sign. Using the subscript i to denote the "invariant" part, and the subscript v to denote the "variant" part, we let

$$\begin{aligned} h_i &= \frac{\gamma_1(1 - r_{\mathbf{f}_2}^2) + \gamma_2(1 - r_{\mathbf{f}_1}^2)}{2<\mathbf{n}_s,\mathbf{n}_b>^2}, \\ h_v &= \frac{\lambda_1(1 - r_{\mathbf{e}_2}^2) + \lambda_2(1 - r_{\mathbf{e}_1}^2)}{2<\mathbf{n}_s,\mathbf{n}_b>}, \\ k_i &= \lambda_1\lambda_2 + \frac{\gamma_1\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>^2}, \text{ and} \\ k_v &= \frac{\lambda_1 r_{\mathbf{e}_2\mathbf{e}_2} + \lambda_2 r_{\mathbf{e}_1\mathbf{e}_1}}{<\mathbf{n}_s,\mathbf{n}_b>}. \end{aligned} \tag{4.4}$$

Equations (3.3) and (3.4) then become

$$h = h_i + h_v \text{ , and} \\ k = k_i + k_v \text{ ,} \tag{4.5}$$

respectively.

Each combination of the signs of k_i, k_v, h_i, and h_v, together with the relative magnitudes of $|k_i|$ and $|k_v|$ and of $|h_i|$ and $|h_v|$ determines a single boundary surface curvature pair, that is, a single boundary primitive. Tables 4.4 and 4.5 show all possible combinations of the signs and relative magnitudes of the variables mentioned above, together with the boundary primitive determined by each combination. The columns labeled "Boundary 1" and "Boundary 2" give the boundary piece labels of the two boundary surfaces associated with the simplified segment. "Boundary 1" is the boundary surface "pointed to" by the simplified segment unit normal, that is, the boundary surface for which $<\mathbf{n}_s, \mathbf{n}_b>$ is positive; "Boundary 2" is the other boundary surface. The columns labeled k_d and h_d give the relationship (<, =, or >) between $|k_i|$ and $|k_v|$ and between $|h_i|$ and $|h_v|$ respectively. A question mark (?) entry indicates that the sign of that quantity is irrelevant as long as it is consistent with the other entries within the same row and triple of columns (separated by double vertical rules). On the other hand, an asterisk (*), optionally preceded by a minus sign (−), indicates that the sign of that quantity is irrelevant as long as all asterisks within the same row and triple of columns are given the same sign. If present, the minus sign indicates that the sign of that quantity must be opposite the sign of other quantities marked by an asterisk alone.

Tables 4.4 and 4.5 make explicit the relationship between symmetric surface and radius curvatures on the one hand, and boundary primitives on the other. I have divided the primitives among these tables to reflect their stability, that is, their behavior under slight perturbations. For those primitives in Table 4.4, a small enough change in any of k_i, k_v, h_i, or h_v, does not change the bound-

Table 4.4
Boundary Primitives (Part 1)

Boundary 1	Boundary 2	k_i	k_v	k_d	h_i	h_v	h_d
Concave	Concave	+	?	>	+	?	>
Concave	Convex	+	?	>	?	+	<
Convex	Concave	+	?	>	?	−	<
Convex	Convex	+	?	>	−	?	>
Concave	Saddle	?	+	<	+	?	>
					?	+	<
					+	+	=
Convex	Saddle	?	+	<	?	−	<
					−	?	>
					−	−	=
Saddle	Concave	?	−	<	+	?	>
					?	−	<
					+	−	=
Saddle	Convex	?	−	<	?	+	<
					−	?	>
					−	+	=
Saddle	Saddle	−	?	>	?	?	?

Table 4.5
Boundary Primitives (Part 2)

Boundary 1	Boundary 2	k_i	k_v	k_d	h_i	h_v	h_d
Concave	Parabolic concave	+	+	=	+	?	>
Concave	Parabolic convex	+	+	=	?	+	<
Concave	Flat	+	+	=	+	+	=
Convex	Parabolic concave	+	+	=	?	−	<
Convex	Parabolic convex	+	+	=	−	?	>
Convex	Flat	+	+	=	−	−	=
Saddle	Parabolic concave	−	−	=	+	?	>
					?	−	<
					+	−	=
Saddle	Parabolic convex	−	−	=	?	+	<
					−	?	>
					−	+	=
Saddle	Flat	−	−	=	*	*	=
Parabolic concave	Concave	+	−	=	+	?	>
Parabolic concave	Convex	+	−	=	?	+	<
Parabolic convex	Concave	+	−	=	?	−	<
Parabolic convex	Convex	+	−	=	−	?	>
Flat	Concave	+	−	=	+	−	=
Flat	Convex	+	−	=	−	+	=
Parabolic concave	Saddle	−	+	=	+	?	>
					?	+	<
					+	+	=
Parabolic convex	Saddle	−	+	=	?	−	<
					−	?	>
					−	−	=
Flat	Saddle	−	+	=	*	−*	=
Parabolic concave	Parabolic concave	0	0	=	+	?	>
Parabolic concave	Parabolic convex	0	0	=	?	+	<
Parabolic concave	Flat	0	0	=	+	+	=
Parabolic convex	Parabolic concave	0	0	=	?	−	<
Parabolic convex	Parabolic convex	0	0	=	−	?	>
Parabolic convex	Flat	0	0	=	−	−	=
Flat	Parabolic concave	0	0	=	+	−	=
Flat	Parabolic convex	0	0	=	−	+	=
Flat	Flat	0	0	=	0	0	=

ary primitive, whereas for those primitives in Table 4.5, any change whatsoever in k_i or k_v changes the boundary primitive. I believe that the primitives in Table 4.5 will be less useful in practice than those in Table 4.6, for in numerical computing it is not possible to compare two quantities for exact equality.

We now turn from an explicit discussion of the relationship between boundary primitives and symmetric surface and radius curvatures, to a more general effort to build intuition about the geometry of the three-dimensional symmetric axis transform. When the simplified segment is flat, intuition suggests that the boundary piece labels for both boundary surfaces of a boundary primitive are identical, and that the signs of the boundary surface curvatures are the same as the signs of the radius function curvatures. Hence, a flat simplified segment is partitioned into identical regions by radius function curvature districts and by boundary primitives. Each induces the same simplified segment partition with the same labels. We can confirm our intuition by noting that when the simplified segment is flat, k_v and h_v are both zero and that k_i is the Gaussian curvature of the radius function. Then, inspecting either equations (4.4) and (4.5) or Tables 4.4 and 4.5 provides the desired confirmation.

The relationships between boundary primitives and curvatures of the simplified segment and radius function are much more complex when the simplified segment is curved rather than flat. We can improve our understanding by trying to analyze the situation in two orthogonal directions, that is, by splitting the three-dimensional geometry into two independent two-dimensional cases. Though we shall see that this is not generally possible, the exercise will illuminate the three-dimensional geometry and will also show, as stated in Chapter 3, that the three-dimensional curvature relationships subsume the two-dimensional relationships given by Blum and Nagel[Blum78a].

Let us begin by rewriting equations (3.3) and (3.4) in another form. Recall (Section 3.5.3) that $\mathbf{e}_1$ and $\mathbf{e}_2$ are unit vectors in the symmetric surface principal directions, $\mathbf{f}_1$ and $\mathbf{f}_2$ are unit vectors in the radius function principal directions, and θ is the counterclockwise angle from $\mathbf{e}_1$ to $\mathbf{f}_1$. We also have from Section 3.5.3 that $r_{\mathbf{e}_1\mathbf{e}_1} = \gamma_1\cos^2\theta + \gamma_2\sin^2\theta$ and $r_{\mathbf{e}_2\mathbf{e}_2} = \gamma_1\sin^2\theta + \gamma_2\cos^2\theta$. Substituting into (3.4) and rearranging terms,

$$k = \lambda_1\lambda_2 + \frac{\gamma_1\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>^2} + \frac{\lambda_1\gamma_2 + \lambda_2\gamma_1 + (\lambda_1 - \lambda_2)(\gamma_1 - \gamma_2)\sin^2\theta}{<\mathbf{n}_s,\mathbf{n}_b>}. \tag{4.6}$$

Assume for the moment that the radius function principal directions and the symmetric surface principal directions coincide, that is, $\theta = 0$. Then, we can rewrite (4.6) to obtain

$$k = (\lambda_1 + \frac{\gamma_1}{<\mathbf{n}_s,\mathbf{n}_b>})(\lambda_2 + \frac{\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>}) .$$

Similarly, by setting $r_{\mathbf{f}_1}^2 = r_{\mathbf{e}_1}^2$ and $r_{\mathbf{f}_2}^2 = r_{\mathbf{e}_2}^2$, recalling that since $\mathbf{n}_s$, $\mathbf{e}_1$, and $\mathbf{e}_2$ are orthonormal, $<\mathbf{n}_s,\mathbf{n}_b>^2 + <\mathbf{n}_b,\mathbf{e}_1>^2 + <\mathbf{n}_b,\mathbf{e}_2>^2 = 1$, and that $r_{\mathbf{e}_1} = <\mathbf{n}_b,\mathbf{e}_1>$ and $r_{\mathbf{e}_2} = <\mathbf{n}_b,\mathbf{e}_2>$, we can rewrite (3.3) as

$$h = ½[(\lambda_1 + \frac{\gamma_1}{<\mathbf{n}_s,\mathbf{n}_b>}) + (\lambda_2 + \frac{\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>})] . \tag{4.7}$$

Hence, we see that when the principal directions of the radius function and of the symmetric surface coincide, $\lambda_1 + \frac{\gamma_1}{<\mathbf{n}_s,\mathbf{n}_b>}$ and $\lambda_2 + \frac{\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>}$ are the principal curvatures of the surface parallel to the boundary surfaces.

After adjusting for somewhat different notation, these expressions for the principal curvatures are each identical to the two-dimensional curvature relation given by Blum and Nagel[Blum78a]. Hence, in this case the three-dimensional curvature relation is determined locally by two orthogonal two-dimensional slices.

Now consider the effect of rotating the radius function principal directions with respect to the simplified segment principal directions. Rewrite (4.6) as

$$k = (\lambda_1 + \frac{\gamma_1}{<\mathbf{n}_s,\mathbf{n}_b>})(\lambda_2 + \frac{\gamma_2}{<\mathbf{n}_s,\mathbf{n}_b>}) + \frac{(\lambda_1 - \lambda_2)(\gamma_1 - \gamma_2)}{<\mathbf{n}_s,\mathbf{n}_b>}\sin^2\theta . \tag{4.8}$$

By definition, $\lambda_1 \geq \lambda_2$ and $\gamma_1 \geq \gamma_2$. Therefore, for the boundary surface determined by $<\mathbf{n}_s,\mathbf{n}_b>$ positive, the effect of rotational displacement is to increase the boundary surface Gaussian curvature. As the radius principal directions are rotated with respect to the simplified segment principal directions, the effect of the displacement increases, reaching a maximum when the two sets of directions are orthogonal and then diminishing as θ approaches 180 degrees.

One expects the magnitude of the effect of the rotational displacement to depend on the range of normal section curvatures (Section 3.2) and radius function second directional derivatives. For if either range is small, the geometry is almost rotationally symmetric, implying that rotation of the principal axes with respect to each other makes little difference unless the other range is correspondingly large. Equation (4.8) confirms that this is indeed the case. Since λ_1 and λ_2 are the maximum and minimum curvatures of all normal sections, their difference determines the range of normal section curvatures. Similarly, the difference between γ_1 and γ_2 determines the range of radius function second directional derivatives.

4.5. Primitive Adjacency Graphs

In the preceding sections we have developed several different ways to partition a simplified segment into primitives, but have ignored the important question of how to maintain information about the spatial relationships among those primitives. Fortunately, this question has been addressed extensively in the picture processing literature in relation to an almost identical problem, image segmentation. There the goal is to subdivide an image into maximal disjoint regions each satisfying some *uniformity predicate*. Ideally, the region defined by each predicate would correspond directly to an object potentially present in the image. For example, when analyzing images of street scenes one would like the regions to correspond to houses, cars, etc. This is rarely possible in practice. Consequently, image segmentation is usually followed by processing called scene analysis, in which global information is used to merge regions and to assign region "interpretations." Both processes require the ability to examine neighboring regions and to determine the spatial relationships among them. This need has led to the development of a number of data structures for maintaining region adjacency information. Pavlidis[Pavlidis77a] discusses several of these and provides references to others.

Perhaps the most useful of these data structures is the *region adjacency graph*, a graph in which each vertex corresponds to a region and two vertices are connected by an edge if their corresponding regions have overlapping boundaries. We can use the same data structure, which we call a *primitive adjacency graph*, to maintain information about the spatial relationships among primitives. Properties of the graph translate directly into relationships among the primitives. For example, a vertex of degree one corresponds to a primitive completely surrounded by another primitive. More generally, a cut-vertex[14] corresponds to a primitive that completely surrounds other primitives. An example is illustrated in Figure 4.15.

The edges of the primitive adjacency graph capture inter-region relationships but carry no information about properties of the primitive(s) themselves. A more complete data structure is the *labeled primitive adjacency graph*, a primitive adjacency graph in which each vertex is labeled with information about the primitive it represents. Consider, as a very simple example, the simplified segment and associated boundary surfaces shown in Figure 4.1 on page 54. The primitive adjacency graph is trivial: two vertices connected by a single edge. If all three primitive sets are used to form a larger cartesian-product primitive set, each vertex could be labeled to indicate which member of each of the three

[14] A vertex v_1 is a cut-vertex if there are two other vertices, v_2 and v_3, such that all paths between v_2 and v_3 contain v_1. Using depth-first search, all of the cut-vertices of a graph can be found in time linear in the sum of the number of vertices and the number of edges in the

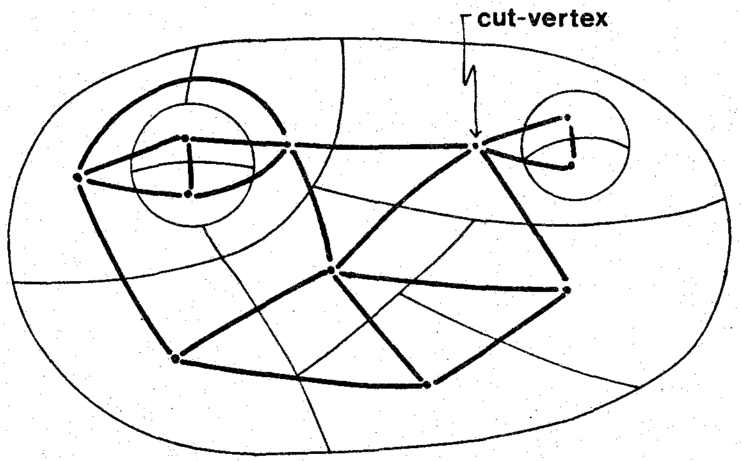


Figure 4.15.: Example of a Primitive Adjacency Graph

primitive sets applies. In this simple example, due to symmetry, both vertices receive the same labels: axis = parabolic convex; boundary = (parabolic concave, parabolic concave); radius = (curvature district: parabolic concave, slope district: external). Of course, the vertex labels need not be restricted to primitive names. Other possibly useful labels include properties of primitives such as maximum and minimum radius function values, region areas, maximum and minimum principal curvatures, etc.

Once a labeled primitive adjacency graph is constructed, the scene analysis techniques reviewed by Pavlidis ([Pavlidis77a], Chapter 6) may be useful for further processing. In more recent work, Shapiro and Haralick[Shapiro81a] have developed an approach to such processing that may prove useful for matching inexactly two labeled primitive adjacency graphs, such as might be derived in shape description using the prototype paradigm of Chapter 1.

graph[Reingold77a].

4.6. Summary

Beginning with the result of the unique figure decomposition induced by Blum's symmetric axis transform, the simplified segment, I have proposed a further decomposition into primitives drawn from three separate, but not completely independent, primitive sets. Each captures different qualitative properties of the two-sided piece associated with a simplified segment. They can either be used separately or combined together to form cartesian-product primitive sets. In the latter case, each primitive becomes an ordered 2- or 3-tuple of primitives drawn from two or three separate primitive sets. I have also proposed a simple data structure, the labeled primitive adjacency graph, to be used to maintain information about the spatial relationships among primitives. These proposals have yet to be tested in practice.

4.7. Unsolved Problems and Research Directions

It is almost superfluous to say that the techniques described here need to be evaluated by applying them, in concert with the other work described in this dissertation, to realistic applications. There are also a number of theoretical issues that should be addressed, preferably in combination with the necessary applied work.

(1) It appears straightforward, but tedious, to remove the assumption that a critical point configuration graph has no adjacent passes.

(2) Pfaltz[Pfaltz76a] has defined a graph of critical points, called a *surface network*, much like the critical point configuration graph of Section 4.2.2.5. As part of his investigation of using surface networks in spatial data bases, he proposed simplifying surface networks by replacing certain subgraphs with a single vertex, thus discarding "irrelevant" detail. Similar techniques might prove useful here to discard minor slope districts, such as those caused by a small "bump" on the side of a large "mountain".

(3) I have proposed three schemes for partitioning a simplified segment into regions, each with its own boundary. Though some properties of these region boundaries are defined, for example Gaussian curvature along boundaries of axis primitives, I have ignored the problem of describing the shape of those boundaries. As mentioned in Chapter 1, it appears that the two-dimensional SAT, together with Blum's shape description methodology, can be generalized easily to apply to outlines on surfaces.

(4) Blum and Nagel[Blum78a] have defined several measures used to characterize properties of the boundaries of two-dimensional simplified segments, branch and end points. For example, they define the "busyness" of a branch point as the number of other branch points contained within the maximal disc centered at the first branch point. I have not defined any similar measures on the boundaries of three-dimensional simplified segments, branch and end curves.

(5) I have used the notion of structural stability to justify an incomplete discussion of radius function behavior near degenerate critical points. In practice, what is meant by a degenerate critical point? Indeed, is it reasonable to expect to be able to use any of the unstable primitives, those, such as the boundary primitives in Table 4.5, that are defined in terms of the equality of two quantities? I think not—some sort of tolerance is necessary. Blum and Nagel[Blum78a] completely ignore the issue in their two-dimensional work. Such numerical issues are rarely addressed in the shape description literature.

CHAPTER 5

APPROXIMATING THE THREE-DIMENSIONAL SYMMETRIC SURFACE

5.1. Introduction

Many algorithms for computing the symmetric axis of a two-dimensional figure, or an approximation thereto, have been developed. With but one exception, which we discuss below, each algorithm is a variation on one of two themes. In the first, the outline is approximated by a simple polygon. Then, an algorithm that computes the true symmetric axis of the polygon[Montanari69a, Lee77a, Preparata77a, Kirkpatrick79a], without regard to any smooth underlying outline, is applied. Unfortunately, the resulting symmetric axis, which consists of line segments and parabolic arcs, differs from the axis of the smooth underlying outline lying near the polygon by the inclusion of simplified segments making contact with each non-reentrant ("convex") polygon vertex. Various thresholding techniques have been devised to delete such superfluous segments[Blum78a, Montanari69a].

In the other common approach, points on the symmetric axis in the digital plane are computed from a digitized outline, either by collapsing the outline into the figure until "opposite sides" of the outline meet on the symmetric axis[Rosenfeld66a, Philbrick68a, Montanari68a, DeSouza77a] or by finding circles that fit just inside the figure[Badler79a]. The latter approach has also been used in three dimensions[O'Rourke79a]. The distance metric used is the primary distinction among these algorithms. All compute only a sampling of points on the symmetric axis, thus losing symmetric axis connectivity information which must then be reconstructed by heuristic means.

Bookstein[Bookstein79a] takes a much different approach. Beginning with a polygonal approximation to a smooth underlying outline wherein each polygon edge is tangent to the outline, his algorithm yields a connected graph of line segments, which he calls the *line-skeleton*, that approximates the symmetric axis of the outline. The resulting line-skeleton is not the symmetric axis of the approximating polygon. It has neither parabolic arcs nor segments contacting non-reentrant vertices. Instead, each element of the line-skeleton lies tangent to the true symmetric axis of the underlying outline.

Both the two-dimensional shape description methodology proposed by Blum and Nagel[Blum78a] and the three-dimensional generalization set forth in this dissertation depend heavily on curvature. This is hardly surprising since the importance of curvature in human shape perception has been widely recognized for years. Yet Bookstein presents the only algorithm of which I am aware that explicitly deals with outline and symmetric axis tangents and that maintains symmetric axis continuity. In my view, any symmetric axis algorithm must have these characteristics if it is to be useful for shape description. My work in three dimensions therefore builds upon Bookstein's work in two dimensions.

In the next section, I describe Bookstein's two-dimensional algorithm. Then, I present a three-dimensional generalization of the key concept on which Bookstein's work is built and sketch an algorithm that utilizes that generalization.

5.2. Bookstein's Line-Skeleton

Bookstein's algorithm is best described in two parts: (1) a procedure that, were it possible in continuous space to examine all points near another point, could find an outline's true symmetric axis, and (2) a discrete approximation of that procedure. The continuous space procedure applicable in two dimensions is also applicable in three dimensions with but minor modification.

5.2.1. The Medial Involution and Continuous Extension

Let C be a smooth outline, let SA(C) be the symmetric axis (surface) of C, and let τ be the mapping from C onto SA(C) that maps a point P_C in C to the center of the maximal disc that touches C at P_C. See Figure 2.4 (page 24). Further, let C_2 be the set of all points P_C in C for which $\tau(P_C)$ is a point contact normal point. For P_C in C_2, there is by definition a single point P_C' in C_2, called the ***medial involute*** of P_C, such that $\tau(P_C') = \tau(P_C)$. The function that maps a point in C_2 to its medial involute, called the ***medial involution***, is continuous on C_2. Let $T(P_C)$ denote the line (plane in three dimensions) tangent to C at P_C and let $N(P_C)$ denote the line normal to C at P_C. As a consequence of the definition of the SA, $\tau(P_C) = \tau(P_C')$ must lie at the intersection of $N(P_C)$, $N(P_C')$, and the bisector of $T(P_C)$ and $T(P_C')$. Further, the bisector is tangent to SA(C) at $\tau(P_C)$.

Given any P_C in C_2, its medial involute is easily found. At each point P of C, construct N(P), T(P), and the bisector of T(P) and $T(P_C)$. Only points P for which the two normals and the bisector coincide are candidates for the medial involute of P_C. Of all candidates, the medial involute is the point P_C' for which the distance between P_C and the point of coincidence is least. See Figure 5.1.

Because the medial involution is continuous, the search for medial involutes of points on C_2 near to P_C can be constrained to a neighborhood of P_C'. The new pair of medial involutes determines a point on SA(C) which, by the continuity of τ, is near to $\tau(P_C)$, thus extending SA(C). A simplified segment can be constructed in two steps:

(1) Pick some point on the outline and search for its medial involute. The resulting pair of medial involutes determines a point on SA(C).

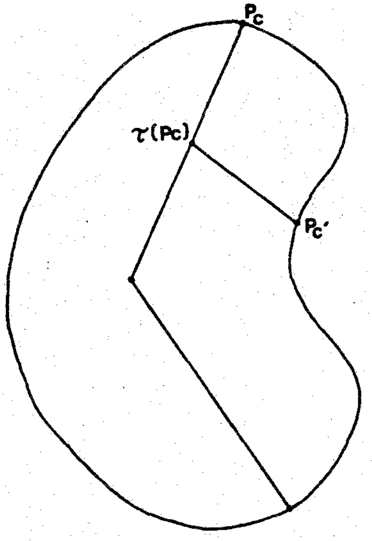


Figure 5.1.: Medial Involutes

(2) Using the continuity of the medial involution, grow SA(C) until a point P is reached where the medial involution fails to be continuous. That point is not in C_2; τ(P) is either a branch point or an end point of SA(C).

Of course, in continous space, where each neighborhood contains an infinite number of points, this procedure never terminates. That is not a problem in discrete space.

5.2.2. Discrete Approximation of the Two-dimensional Medial Involution

To apply the aforementioned procedure to discrete data, a discrete approximation of the continuous medial involution is required. Recall that Bookstein begins with a polygonal approximation to the underlying outline wherein each polygon edge lies tangent to the outline at some point along its length. This provides a sampling of the outline tangent assumed to be sufficiently fine to capture the outline curvature. Bisectors of adjacent polygon edges, called ***pseudonormals***, serve as approximations of outline normals. Consider two non-adjacent edges, e_i and e_j, and let B be their bisector line, as shown in Figure 5.2.

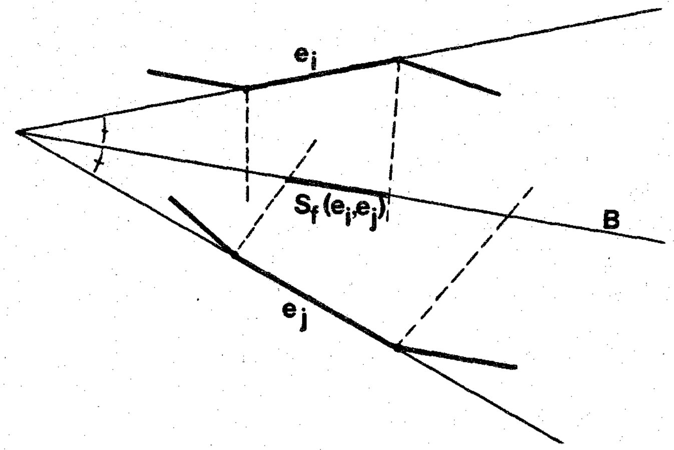


Figure 5.2.: Finite Skeletal Line Elements (after [Bookstein79a])

Edges e_i and e_j each determine a pair of pseudonormals, one through each endpoint, which demarcate a (possibly empty) interval on B. The two edges are called ***discrete medial involutes*** if the two intervals so defined on B overlap. In that case, the overlap on B is called the ***finite skeletal line element*** (fsle) of e_i and e_j, denoted $S_f(e_i, e_j)$.[1]

In continuous space coincidence of the normals at each of two outline points with the bisector of the tangent planes at those points is necessary, but not sufficient, to ensure that those points are medial involutes. Similarly, each edge of the approximating polygon may have more than one discrete medial involute. A ***true discrete involute*** (tdi) of a polygon edge e_i is a discrete medial involute of e_i for which the corresponding fsle is closest to e_i, distance between an fsle and e_i being defined as the smaller of the distances from the endpoints of the fsle to the line containing e_i. Whenever two edges are true discrete involutes, the fsle between them is presumed to approximate a (one-dimensional)

[1] Figure 5.2 and all subsequent figures in Section 5.2.2 are closely modeled after figures in [Bookstein79a]. Polygon edges and fsle's are drawn bold with endpoints shown as black dots. Pseudonormals are drawn dashed.

neighborhood of the true symmetric axis. Moreover, the two fsle's defined by two contiguous edges and a third, non-contiguous edge are connected, as shown in Figure 5.3.

Bookstein's algorithm, as he describes it[Bookstein79a], is a "tree-structured assembly" of several operations on edges and fsle's: finding initial finite skeletal line elements, extending fsle's into fsle chains, and determining branch and end points. The algorithm begins by picking an arbitrary edge of the approximating polygon and finding one of its tdi's[2] and corresponding fsle. From this "seed" fsle, the algorithm constructs two connected fsle chains, one left and one right, by "marching" along the polygon edges as illustrated in Figure 5.4. (In essence, the transition from continuous to discrete space replaces extension of the SA by neighborhood search as described in Section 5.2.1, with simple extension of a chain of fsle's.) The left and right chains are constructed

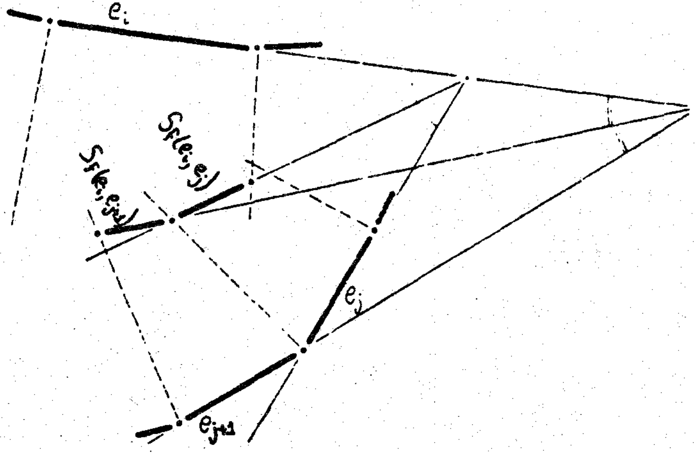


Figure 5.3.: Connected Fsle's (after [Bookstein79a])

[2] If the approximating polygon does not adequately capture the curvature of the underlying outline, the edge may not have a tdi. In that case, another starting edge must be chosen. In general, a small "gap" appears in the symmetric axis approximation whenever an edge has no corresponding fsle. Bookstein describes an *ad hoc* procedure for patching such "gaps."

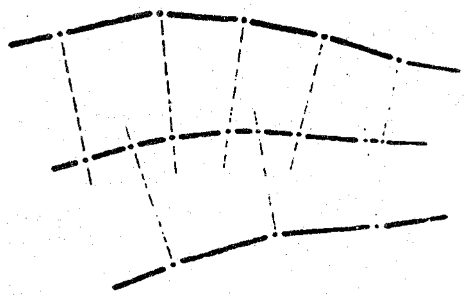


Figure 5.4.: Fsle Chain Extension (from [Bookstein79a])

independently and identically; consider the extension right. Eventually, the extension fails in one of two ways, corresponding either to an end or to a branch point of the true symmetric axis. In the first failure mode, which Bookstein calls "failure by mode A," the extension terminates when the two edges that determine the rightmost fsle are separated by but one edge, as shown in Figure 5.5. Extension failure by mode A corresponds to reaching an end point.

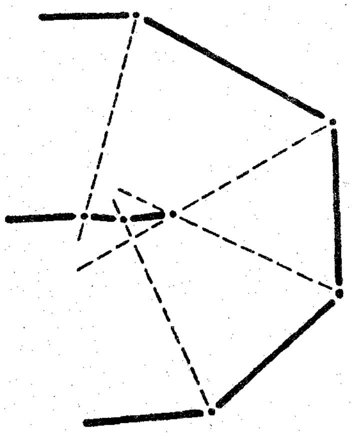


Figure 5.5.: Mode A Chain Termination (after [Bookstein79a])

In the second failure mode, illustrated in Figure 5.6, the pseudonormals of an edge cross before intersecting the bisector line that contains what should be the next chain fsle, thus terminating the chain. This failure mode, which Bookstein calls "failure by mode B," occurs either when an edge has no tdi because the outline curvature is sampled inadequately or when the fsle chain is extended *past* a branch point. I ignore the former case. The latter situation is illustrated in Figure 5.7. As chain extension proceeds rightward from $S_f(e_i, e_j)$, crossing the true branch point (shown solid with its branches), the algorithm must eventually encounter an edge on the "upper" boundary arc, here edge e_k, whose tdi, here edge e_{k-2}, lies between e_i and e_k. Since the pseudonormals of e_k must intersect beyond the upward branch from the branch point, they must also intersect above the chain being extended right from $S_f(e_i, e_j)$. But this implies failure of the extension right by mode B. Therefore, extension past a branch point implies eventual failure by mode B.

We must still approximately locate the branch point. Upon failure of the extension after fsle $S_f(e_{k-3}, e_m)$, the algorithm finds the tdi e_k of e_{k-2} by

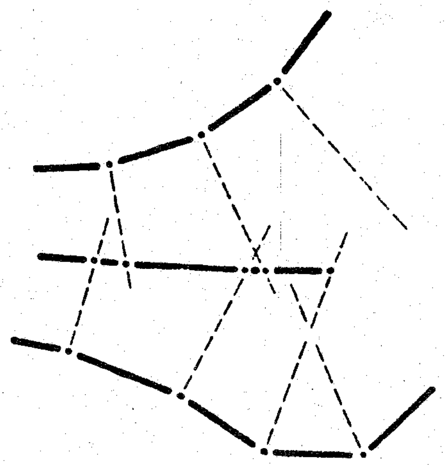


Figure 5.6.: Mode B Chain Termination (after [Bookstein79a])

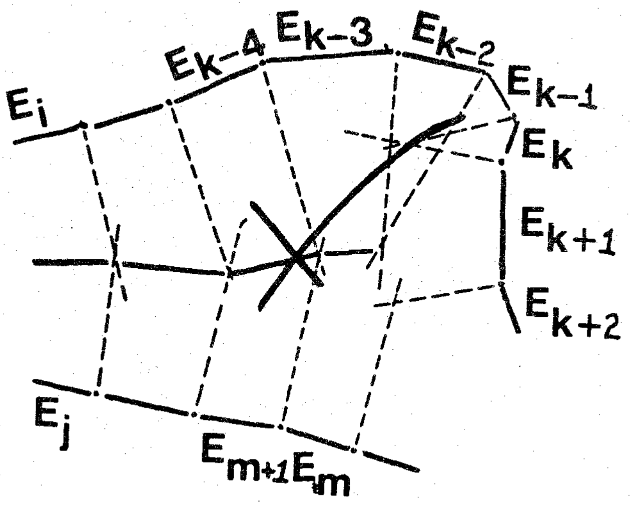


Figure 5.7.: Extension Failure Past Branch Points (after Bookstein79a])

exhaustive search and then extends left a new chain from $S_f(e_k, e_{k-2})$. Either the extension left fails by passing a second branch point or an fsle of the new chain intersects an fsle of the original chain. In the former case, illustrated schematically in Figure 5.8, a new tdi is found and yet another new chain is extended left by this same procedure. In the latter case, the intersecting fsle's, here $S_f(e_{k-4}, e_{k+1})$ and $S_f(e_{k-4}, e_{m+1})$ are determined by the common edge e_{k-4}. Furthermore, a third fsle, $S_f(e_{m+1}, e_{k+1})$ intersects at the same point. Two new chains out of this point of intersection are constructed by recursively invoking the extension procedure twice, once using $S_f(e_{k-4}, e_{k+1})$ as the "seed" fsle and once using $S_f(e_{m+1}, e_{k+1})$.

Upon failure of all extension procedures at end points, that is, by mode A, the algorithm terminates, yielding a connected chain of fsle's each lying tangent to the true symmetric axis of the underlying outline. Of course, this description of Bookstein's algorithm is a simplification; his exposition[Bookstein79a] is more

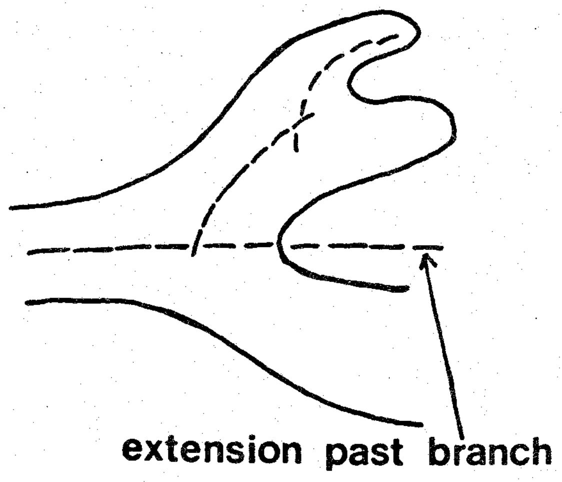


Figure 5.8.: Extension Left to Branch Point (after [Bookstein79a])

complete.

5.3. Overview of the Three-dimensional Algorithm

In three dimensions, we seek an algorithm that takes a polyhedral approximation to a smooth outline and yields a polyhedral surface approximating the symmetric surface of the outline. Since the two- and three-dimensional continuous medial involutions are defined identically, our approach is to generalize Bookstein's algorithm. The principal task is to define three-dimensional analogs of pseudonormals and finite skeletal line elements, the two components of the two-dimensional discrete medial involution. In three dimensions, we approximate outline normals by polyhedral regions of space, called *pseudonormal pencils*,[3] rather than by pseudonormal lines, and symmetric surface neighborhoods by planar polygons, called *symmetric surface planar elements* (sspe's), rather than by line segments. There corresponds to each pair of non-adjacent faces of the approximating polyhedron an sspe, possibly empty, defined by the overlap of

[3] I shall use the term pencil in its informal sense—something long and thin like a pencil—rather than in the sense used in projective geometry.

two pseudonormal pencils upon the bisector of the faces. Two such faces having a non-empty symmetric surface planar element are *discrete medial involutes*, the discrete analog of continuous medial involutes.

Like Bookstein's two-dimensional algorithm, the three-dimensional algorithm I propose below consists of three basic operations: finding a "seed" sspe, extending sspe's into polyhedral surfaces, and determining branch and end point curves. Given a polyhedral approximation to a smooth outline, the algorithm begins by arbitrarily choosing a face of the approximation and then finding, by exhaustive search, a true discrete involute of the starting face, and hence, a "seed" sspe. Then, using this "seed," the simplified segment extension procedure constructs, without further searching, the entire simplified segment containing the "seed" sspe. The extension procedure fails at end curves and past branch curves, much as in the analogous two-dimensional situation. However, once the branch is detected, a new "seed" sspe is found, again by exhaustive search, and the extension procedure is invoked to construct another simplified segment. By intersecting the new simplified segment with the original, the actual location of the branch curve can be found and other simplified segments constructed.

In the balance of this chapter, I describe a the three-dimensional generalization of Bookstein's algorithm outlined above. In the next section, I define pseudonormal pencils and symmetric surface planar elements, and investigate their individual properties. Then, in the following section, after showing that sspe's "fit together" to form a polyhedral surface, I present a three-dimensional sspe extension procedure. Finally, I outline a procedure for intersecting simplified segments to find the actual location of branch curves.

5.4. Three-dimensional Discrete Medial Involutes

This section addresses the principal task of the chapter, defining the components of the three-dimensional discrete medial involution: pseudonormal pencils and symmetric surface planar elements. I first introduce the terminology and the mathematical concepts we shall need and discuss required properties of the approximating polyhedral surface. Then, I define pseudonormal pencils and show intuitively in what sense they approximate normals to the underlying outline. Using this understanding, I then define symmetric surface planar elements and investigate their individual properties.

5.4.1. Background

Since different sources use the same terminology for slightly different notions, the terminology we shall use is defined here.

Definition 5.1: A *closed polygonal curve* is a finite set of line segments such that

(1) Two distinct closed line segments are either disjoint or intersect at a common endpoint.

(2) Each endpoint is an endpoint of exactly two line segments. ■

Definition 5.2: A *closed planar polygonal curve* is a polygonal curve contained in a plane. ■

Definition 5.3: A *closed polygon* is the union of a closed planar polygonal curve and its inside. A *vertex* is a point at which two non-collinear line segments in the polygonal curve intersect. An *edge* is a closed line segment in the polygonal curve with vertices as endpoints. ■

Definition 5.4: A *polyhedral surface (without boundary)* is a finite set of closed polygons called *faces*, such that

(1) Two faces are either disjoint or intersect in an entire edge common to both faces or in a vertex common to both faces.

(2) Each edge of each face is also an edge of exactly one other face.

(3) The set of faces that share a common vertex can be labeled in cyclic order, $F_0, \ldots, F_{n-1}, F_n = F_0$, such that F_i and F_{i+1} share a common edge. ■

Figure 5.9 (a) illustrates several permissible face intersections, while (b) illustrates several illegal intersections.

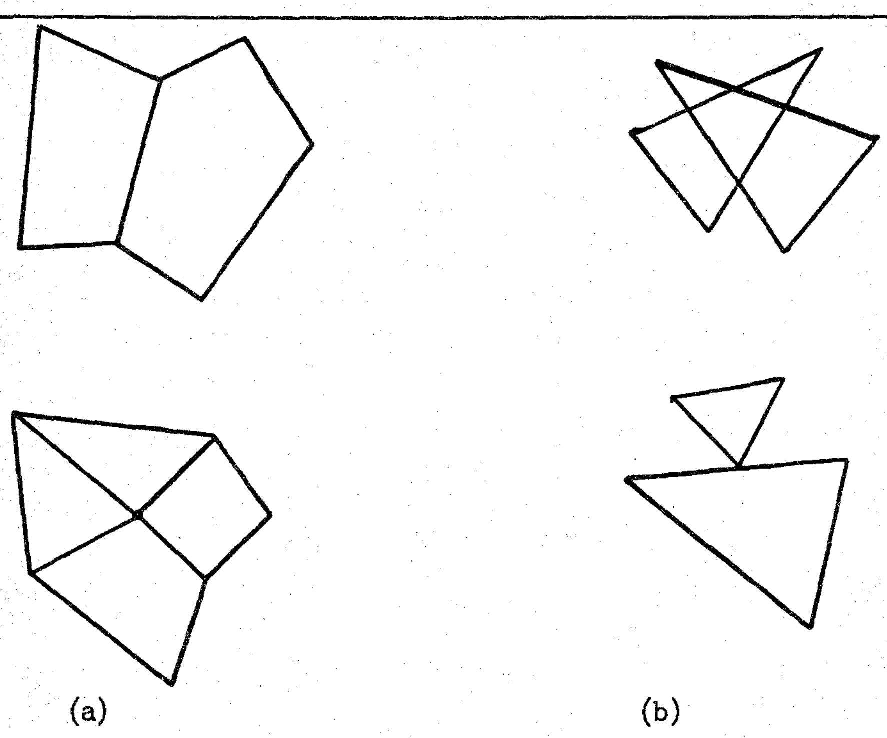


Figure 5.9.: Face Intersections

Definition 5.5: A *polyhedral surface with boundary* is a finite set of closed polygons, called *faces*, such that

(1) Two faces are either disjoint or intersect in an entire edge common to both faces or in a vertex common to both faces.

(2) Each edge of each face is also an edge of at most one other face. An edge contained in exactly one face is called a *boundary edge*.

(3) The set of faces that share a common vertex can be labeled $F_0, \ldots, F_{n-1}, F_n$, such that for $0 \leq i < n$, F_i and F_{i+1} share a common edge. If, in addition, $F_0 = F_n$, the vertex is called an *interior vertex*; otherwise it is called a *boundary vertex*. ■

The term *polyhedral surface*, used without qualification, refers to a polyhedral surface without boundary.

We shall also use some elementary concepts of elementary point set topology in Euclidean spaces. Concise, yet readable treatments of these concepts appear in Sections 1.1 through 1.7 of [Kelly79a] and in Section 2 of [Requicha78a], as well as in many elementary topology texts.

Definition 5.6: Let $d(p,q)$ denote the Euclidean distance between two points p and q. The set $N(p,\delta) = \{q \mid d(p,q) < \delta\}$ is the *neighborhood* of p with *center* p and *radius* δ. ■

Definition 5.7: A point p is an *interior point* of a set A if there exists a neighborhood of p that is contained in A. A point p is an *exterior point* of a set A if there exists a neighborhood of p that is contained in the complement of A. A point p is a *boundary point* of a set A if every neighborhood of p intersects both A and the complement of A. The *interior* of A, denoted int$A$, is the set of all interior points of A. The *exterior* of A, denoted ext A, is the set of all exterior points of A. The *boundary* of A, denoted bd$A$, is the set of all boundary points of A. ■

Definition 5.8: A set A is *open* if it consists entirely of interior points. A set A is *closed* if it contains its boundary. ■

Note that a set can be both open and closed. For example, the empty set is both open and closed.

Definition 5.9: The *closure* of a set A, denoted cl$A$, is the union of the set and its boundary. ■

Definitions 5.6 through 5.9 depend crucially on the *universe*, the set characterized by its complement being empty. In other words, the universe contains all points considered. Different universes can yield different results for the same notion. For example, if A is a closed line segment in the universe $\mathbf{R}^1$, bd$A$ consists of the two endpoints. If, however, the same line segment is considered in the universe $\mathbf{R}^2$, bd$A$ is A itself. Unless specified, the universe will be clear from context.

Where it is necessary to work simultaneously with two universes, one a subset of the other, say $U' \subset U$, we denote the interior, boundary, exterior, and closure of a set A in the universe U' by int'A, bd'A, ext' A, and cl'A respectively. Additionally, if with respect to the universe U' A is open, it will be said to be *open in* U'. If with respect to the universe U' it is closed, it will be said to be *closed in* U'. The following result shows how some of these notions are related for different universes.

Lemma 5.1: Let U be the universe and let $U' \subset U$. Then

(1) A neighborhood in U' is the intersection of U' with the neighborhood of U that has the same center and radius.

(2) A subset of U' is open in U' if and only if it is the intersection of U' with an open subset of U.

(3) A subset of U' is closed in U' if and only if it is the intersection of U' with a closed subset of U.

(4) If A is a subset of U', then the closure of A in U' is the intersection of U' with the closure of A in U.

(Theorem 7, Section 1.3, [Kelly79a]). ■

5.4.2. The Approximating Polyhedral Surface

The algorithm is to take as its input a polyhedral surface that approximates an outline. To be acceptable, the approximating polyhedral surface must satisfy certain conditions which, after appropriate notation is introduced, are discussed in this section.

Each face, F_i, of the polyhedral surface is bounded by n_i edges denoted e_i^j, $j=0, \ldots, n_i-1$. See Figure 5.10. Beginning with an arbitrary edge, the edges are numbered in counterclockwise order as seen from outside the face. There are n_i vertices $v_i^j = e_i^j \cap e_i^{CC_i(j)}$, where $CC_i(j) = (j+1) \bmod n_i$ denotes the index

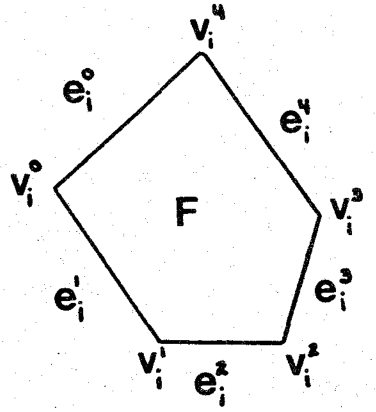


Figure 5.10.: Face and Vertex Notation

of the edge on F_i counterclockwise adjacent to e_i^j. Henceforth, we use subscripts to index faces and superscripts to index edges and vertices. Two distinct faces are called *edge-adjacent* if they share a common edge and *vertex-adjacent* if they share a common vertex. By convention, a face is neither edge-adjacent nor vertex-adjacent to itself. Note, however, that part (1) of Definition 5.4 implies that edge-adjacent faces are also vertex-adjacent. Two vertex-adjacent faces are also called *neighbors*. Finally, let V_i be the set of indices of all faces vertex adjacent to F_i and let E_i be the set of indices of all faces edge adjacent to F_i.

In addition to the conditions imposed by Definition 5.4, the approximating polyhedral surface must satisfy the following conditions:

(1) each face must be a convex closed polygon;

(2) at some point within its extent, each face of the approximating polyhedral surface must lie tangent to the underlying smooth outline; and,

(3) the approximating polyhedral surface must "adequately" capture the curvature of the outline.

Note that condition (1) does not require that the approximating polyhedral surface bound a convex polyhedron, but only that each face be convex. I shall elaborate condition (3) below.

Throughout this chapter, we shall also assume that vertex-adjacent faces are not coplanar. Though this condition is not essential, it results in a substantial simplification of the algorithm description with little loss of generality.

5.4.3. Pseudonormal Planes and Pencils

At some point within its extent, each face of the approximating polyhedral surface lies tangent to the underlying outline. Unfortunately, discrete samples of normals to the outline cannot be obtained by constructing a perpendicular to the face at the point of tangency, for the point of tangency is not known. Instead, a suitable approximation is required. Therefore, in this section I define for each face of the approximating polyhedral surface a set of *pseudonormal planes* which, taken together, determine a *pseudonormal pencil* associated with that same face. I then argue intuitively that pseudonormal planes and pseudonormal pencils together play the role of outline normals.

Let F_i be a face of the approximating polyhedral surface and let π_i denote the plane containing F_i. Assign to each face, the inward directed unit normal vector $\mathbf{n}_i$, and denote by π_i^+ the open half-space into which the normal vector points, by π_i^- the opposite open half-space, and by π_i^{0+} and π_i^{0-} the corresponding closed half-spaces. We shall consider a plane to be defined not only by its point set but also by the direction of its unit normal. For any point in space p, the signed distance from p to π_i, denoted $d_{\pi_i}(p)$, is defined as

$$d_{\pi_i}(p) = \begin{cases} d(p,\pi_i) & \text{if } p \in \pi_i^{0+} \\ -d(p,\pi_i) & \text{if } p \in \pi_i^- , \end{cases}$$

where $d(p,\pi_i)$ denotes the shortest distance between the point p and the plane π_i.

Each pair of faces F_i and F_j determines a set $\sigma_{ij} = \{p \mid d_{\pi_i}(p) = d_{\pi_j}(p)\}$, called the ***bisector*** set. For all but parallel faces, σ_{ij} is one of the two planes that bisect the angle formed by the intersection of π_i and π_j. When π_i and π_j are parallel and distinct, σ_{ij} is either the plane midway between them ($\mathbf{n}_i = -\mathbf{n}_j$), or is empty ($\mathbf{n}_i = \mathbf{n}_j$). When $\pi_i = \pi_j$, σ_{ij} is all of space.

Definition 5.10: Let F_i and F_m be vertex-adjacent faces. The *pseudonormal plane* N_{im} is σ_{im}, the bisector of π_i and π_m. ▪

See Figure 5.11. Note that the pseudonormal plane N_{im} determined by faces F_i and F_m may intersect the interiors of either or both of the faces, as illustrated in Figure 5.12 (the vertex shared by F_i and F_m is not in the plane of the paper).

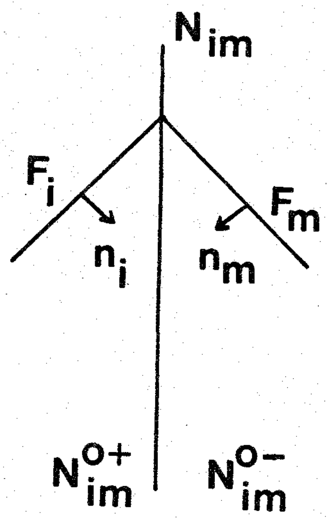


Figure 5.11.: Pseudonormal Planes (side view)

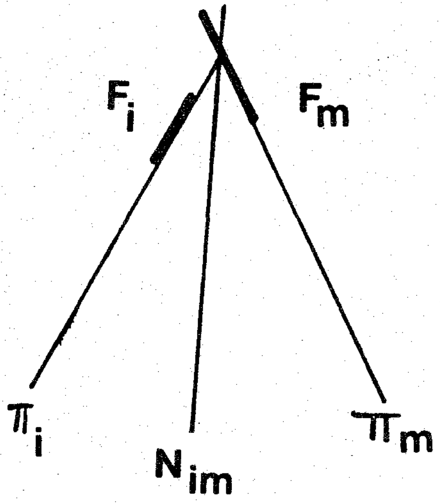


Figure 5.12.: Faces "Cut" by a Pseudonormal Plane (side view)

Assume for now that F_i is contained in one of the two closed half-spaces bounded by N_{im}. Denote by N_{im}^{0+} the closed half-space bounded by N_{im} containing F_i, by N_{im}^{0-} the other closed half-space (which need not contain F_m), and by

N_{im}^{+} and N_{im}^{-} the corresponding open half-spaces. N_{im}^{0+} and N_{im}^{0-} are called respectively the *closed positive* and *closed negative pseudonormal half-spaces* of N_{im}. Similarly, N_{im}^{+} and N_{im}^{-} are called respectively the *open positive* and *open negative pseudonormal half-spaces* of N_{im}. We shall see below that the case where F_i is not contained in one of the two closed half-spaces bounded by N_{im} is irrelevant.

Definition 5.11: Let F_i be a face. The *closed pseudonormal pencil* of F_i, $P_i^{0+} = \bigcap_{k \in V_i} N_{ik}^{0+}$, is the intersection of the closed positive pseudonormal half-spaces of the pseudonormal planes determined by F_i and its neighbors. The *open pseudonormal pencil* of F_i, $P_i^{+} = \bigcap_{k \in V_i} N_{ik}^{+}$, is the intersection of the corresponding open positive pseudonormal half-spaces. ■

See Figure 5.13.

A normal pencil approximates a true outline normal in the following sense. Consider a neighborhood of radius δ about a point p on the underlying outline. There is a line normal to the outline through each point on the neighborhood boundary. Collectively, those lines sweep out a surface in space that separates

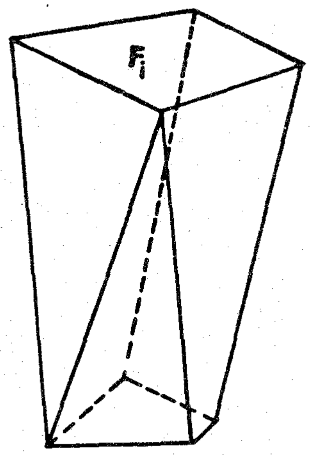


Figure 5.13.: Example of a Pseudonormal Pencil

space into regions, one of which contains the neighborhood of p and is called the ***normal pencil*** at p of ***pencil radius*** δ. The surface swept out by the normal is called the ***normal pencil boundary***. See Figure 5.14. As the pencil radius is made to approach zero, the neighborhood becomes more closely approximated by the tangent plane to the outline at p, while at the same time the normal pencil becomes more nearly cylindrical with the normal through p as its axis.

Pseudonormal pencils, in turn, approximate normal pencils. Each face of the approximating polyhedral surface defines the tangent plane to the outline at some point within the face. As the approximating polyhedral surface becomes increasingly accurate, that is, as its faces become smaller and more numerous, each face becomes a better approximation of a neighborhood about a point on the underlying outline. Moreover, the points of tangency of neighboring faces move closer together, implying that the pseudonormal planes become increasingly accurate approximations to normals on the neighborhood boundary, and hence, that pseudonormal pencils approach normal pencils. We shall therefore use pseudonormal pencils to approximate outline normals.

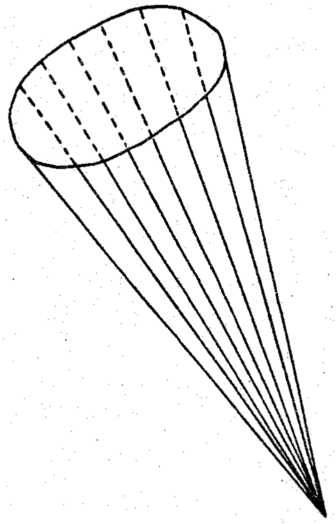


Figure 5.14.: Example of a Normal Pencil

5.4.4. Symmetric Surface Planar Elements

Recall from Section 5.2.1 that two points on the outline, p and q, are potential medial involutes only if the normals at those points intersect on the bisector of the tangent planes at p and q and, furthermore, that if p and q are indeed medial involutes, the point of intersection is on the symmetric surface. Since normals to a smooth surface (such as the outline) change continuously, for small enough pencil radii the intersection of normal pencils at p and q with the bisector of the tangent planes at p and q approximates a neighborhood of the symmetric surface. Similarly, in the discrete case, two faces are *discrete medial involute candidates* if the closed pseudonormal pencils at those faces intersect on the bisector plane between the two faces. Temporarily ignoring certain details, the neighborhood of the bisector plane so defined is called the *symmetric surface planar element candidate* (sspec) defined by the two faces. The approximate symmetric surface consists entirely of sspec's, each of which approximate a symmetric surface neighborhood; not all sspec's are part of the approximate symmetric surface. An sspec contained in the approximate symmetric surface is called a *symmetric surface planar element* (sspe). Unlike Bookstein's fsle terminology, the terminology used here distinguishes between sspec's that are *potentially* part of the approximate symmetric surface and sspe's that *are* part of the approximate symmetric surface. Pairs of faces for which the corresponding sspec is not empty are called *discrete medial involute candidates*, and those pairs of faces for which the corresponding sspec is also an sspe are called *discrete medial involutes*.

In the next section, I discuss properties of continuous medial involutes, normal pencils, and symmetric surface neighborhoods that discrete medial involutes, pseudonormal pencils, and sspe's must also possess if they are to be reasonable approximations of their continuous counterparts. Then, I give a formal definition of sspec's and prove that the conditions established therein ensure the necessary properties.

5.4.4.1. Definition

Let F_i and F_j be distinct faces of the approximating polyhedral surface. Informally, the sspec determined by F_i and F_j, denoted $S_{ij} = S_{ji}$, is the intersection of the bisector plane σ_{ij} with the closed pseudonormal pencils P_i^{0+} and P_j^{0+}. Unfortunately, such a simple definition is not adequate if we are to avoid running afoul of artifacts caused by the noninfinitesimal extent of pseudonormal pencils. If the approximating polyhedral surface does not adequately sample the curvature of the underlying smooth outline, pseudonormal pencils and sspe's need not possess certain properties of the normal pencils and symmetric surface neighborhoods they approximate. In the seven items below, I discuss these properties by comparison to the corresponding continuous behavior. Then I set forth formally conditions sufficient to ensure that non-empty sspec's possess the requisite properties.

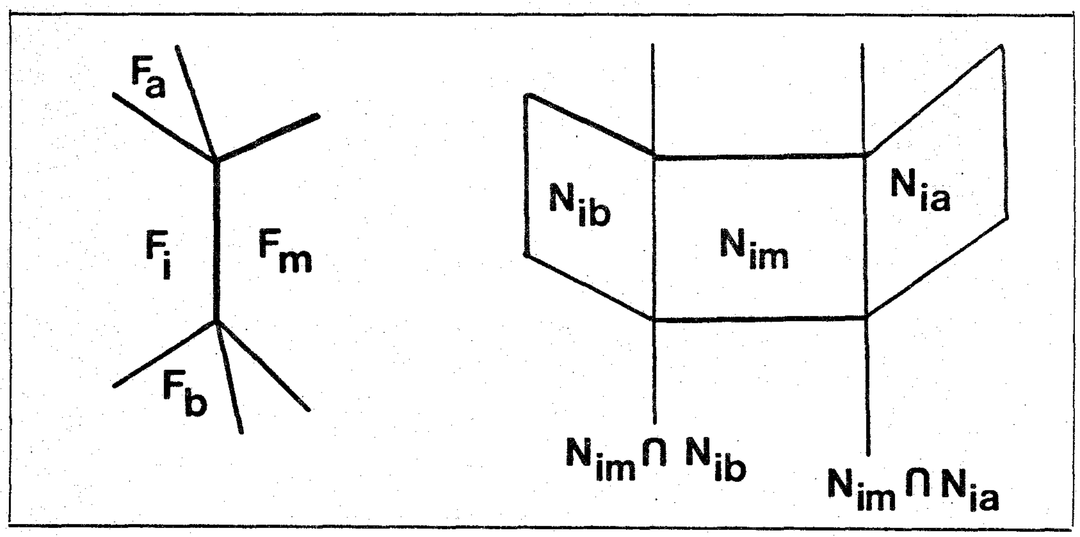
(1) By definition, a normal pencil contains its defining neighborhood. Similarly, we require (part (1) of Definition 5.14 below) that each closed pseudonormal pencil contain the face that defines it. Note that a face having an ill-defined pseudonormal half-space (page 127), has no sspec associated with it that satisfies this condition.

(2, 3) Let p and q be two continuous medial involutes. By definition, p and q also lie on a maximal sphere centered on a symmetric surface point and are strictly separated by the tangent plane to the symmetric surface at the sphere center. Moreover, the sphere is tangent to the outline at p and at q. Since no two points on a sphere have the same normal,[4] the normals at p and q are distinct. Therefore, since the normals of a

[4] Antipodal points on the sphere have parallel normals but they are directed in opposite directions.

smooth surface change continuously, there exist two neighborhoods on the outline, one about p and one about q, such that the normal at any point in the neighborhood of p is different from the normal at q, and vice versa. We shall require analogous behavior of discrete medial involutes: σ_{ij} must strictly separate F_i and F_j; the normals of the neighbors of F_j are distinct from the normal of F_i, and the normals of the neighbors of F_i are distinct from the normal of F_j. (Parts (2) and (3) of Definition 5.14 below.)

(4) For a normal pencil at p (likewise at q) to demarcate a neighborhood on the symmetric surface, the pencil radius must be sufficiently small that the pencil intersects the bisector of the tangent planes at p and q in a closed curve rather than in open curve.[5] Similarly, we shall require that for any pair of discrete medial involutes F_i and F_j, the pseudonormal pencils defined by F_i and F_j must each intersect the bisector plane σ_{ij} in a closed polygonal curve. To develop conditions sufficient to ensure such behavior, we consider an example:

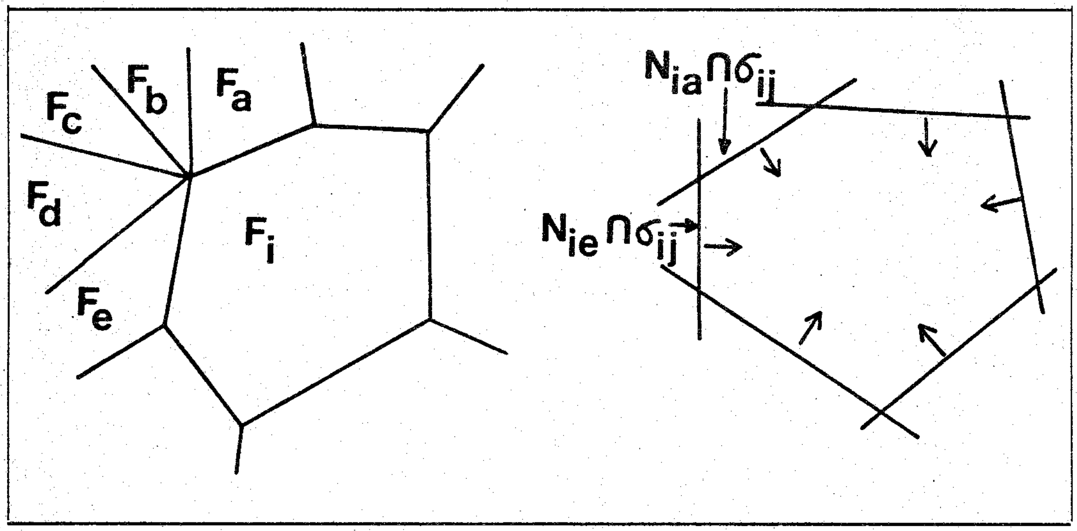


The left figure shows two edge-adjacent faces F_i and F_m viewed from

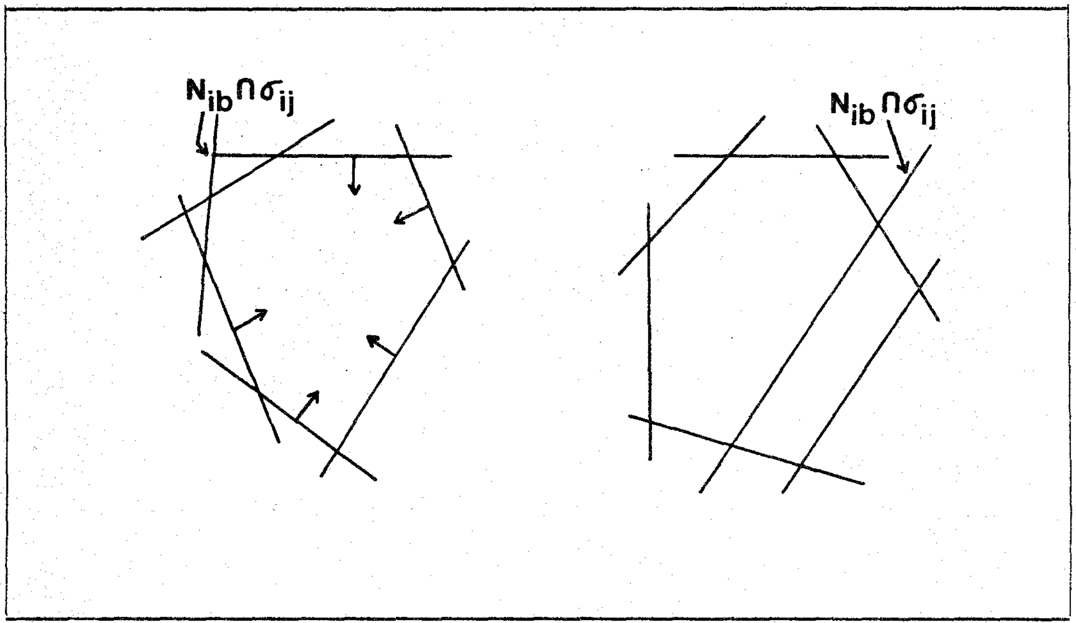
[5] This is analogous to the intersection of a plane and a cone. Depending upon the generating angle of the cone and the orientation of the plane with respect to the cone axis, the curve of intersection is either an ellipse (closed), a parabola (open), or a hyperbola (open).

outside of the approximating polyhedral surface. Faces F_a and F_b are the two faces that are both edge-adjacent to F_i and vertex-adjacent to F_m. The right figure shows the three pseudonormal planes N_{im}, N_{ia}, and N_{ib} viewed from the F_m side of N_{im}. Pseudonormal planes N_{im} and N_{ia} intersect in a line; likewise N_{im} and N_{ib} also intersect in a line. We shall prove in the following section that if all such lines (i.e., for all faces F_m edge-adjacent to F_i) are not parallel to the bisector plane σ_{ij}, then the pseudonormal pencil at F_i intersects σ_{ij} in a closed polygonal curve. (Part (4) of Definition 5.14 below.)

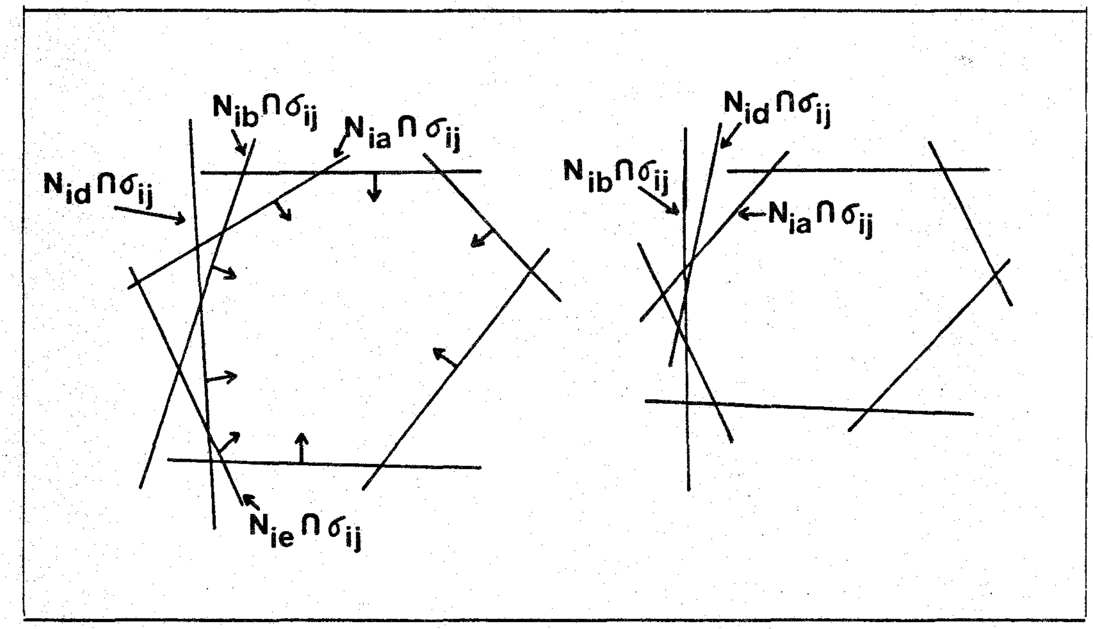
(5, 6) A pseudonormal pencil approximation of a normal pencil must be local in two senses. First, the pseudonormal planes that provide estimates of normals in one portion of the normal pencil's defining neighborhood should have no effect on estimates in other portions of the neighborhood. Second, those estimates should be ordered about the pseudonormal pencil of a face in the same way that neighboring faces are ordered about the face. Let us again consider an example:



The left figure shows a face F_i and its neighbors viewed from outside the approximating polyhedral surface. Consider the intersection of the bisector plane σ_{ij} with all of the closed positive pseudonormal half-spaces defined by faces edge-adjacent to F_i, as shown on the right. Now consider the intersection of this polygon with the positive pseudonormal half-space defined by the non-edge-adjacent neighbors of F_i, in this example F_b, F_c, and F_d. We require (part (5) of Definition 5.14 below) that if one or more of the pseudonormal planes N_{ib}, N_{ic}, or N_{id} intersects the polygon they do so only in the two edges defined by the pseudonormal planes N_{ia} and N_{ie}. This type of intersection is shown on the left below; a prohibited intersection is shown on the right:



We also require (part (6) of Definition 5.14) that as the edges of the resulting polygon are traversed in some direction, say clockwise, the neighbors of F_i that determine the pseudonormals containing the edges are traversed in clockwise order about F_i, with the possible exception that not all neighbors need be traversed. This type of ordering is shown on the left below; a prohibited ordering is shown on the right:


$N_{ib} \cap \sigma_{ij}$
$N_{ia} \cap \sigma_{ij}$
$N_{id} \cap \sigma_{ij}$
$N_{ie} \cap \sigma_{ij}$
$N_{id} \cap \sigma_{ij}$
$N_{ib} \cap \sigma_{ij}$
$N_{ia} \cap \sigma_{ij}$

(7) Thus far, we have considered only properties required independently of each of the two pseudonormal pencils that determine an sspec; we now deal with a property of their intersection. Since an sspec is to approximate a neighborhood of the simplified segment, it must be two-dimensional, neither a point nor a curve. To avoid such degeneracies, sspec's are defined in terms of the intersection of open, rather than of closed, pseudonormal pencils.[6] Taking intersections of open pencils (which are open sets) ensures that such degeneracies cannot occur because, as we shall show in the next section, the intersection of two open pencils with the bisector plane is an open set, S'_{ij}, in the plane σ_{ij}. Since open sets in a plane (other than the empty set) are, by definition, two-dimensional, S'_{ij} is two-dimensional. The sspec S_{ij} is defined as the closure in the plane σ_{ij} of S'_{ij}. The closure operator simply "wraps" a boundary around S'_{ij} so that the sspe is a closed polygon rather than just the interior of a closed polygon.

[6] This is more an issue of mathematical formulation than of practical significance, for in numerical computing of this sort the notion of a closed set is specious: numerical error precludes any test for strict equality.

The conditions introduced in the preceding informal discussion are set forth formally in the next three definitions.

Definition 5.12: Let F_i be a face of the approximating polyhedral surface. A ***partial pseudonormal pencil*** at F_i is the intersection of two or more of the closed positive pseudonormal half-spaces of the pseudonormal planes determined by F_i and its neighbors. ■

Every partial pseudonormal pencil at F_i contains the pseudonormal pencil at F_i.

Definition 5.13: Consider a plane that intersects the boundary of a partial pseudonormal pencil at face F_i in a polygonal curve. Each edge of the polygonal curve is contained in the intersection with a pseudonormal plane defined by F_i and one of its neighbors; the edge is said to be associated with the neighbor. The pseudonormal pencil at F_i is *well-ordered* with respect to a plane if, for every partial pseudonormal pencil that intersects the plane in a polygonal curve, as the edges of the polygonal curve are traversed clockwise,[7] the associated neighbors of F_i are traversed in clockwise order about F_i, with the possible exception that some neighbors may not be traversed. ■

Definition 5.14: Let F_i and F_j be faces of the approximating polyhedral surface, and let

$$S'_{ij} = S'_{ji} = \sigma_{ij} \cap P_i^+ \cap P_j^+ . \tag{5.1}$$

If

(1a) $F_i \subset P_i^{0+}$;

(1b) $F_j \subset P_j^{0+}$;

(2) the bisector plane σ_{ij} separates[8] F_i and F_j;

(3a) for $k \in V_j$, $\mathbf{n}_i \neq \mathbf{n}_k$;

(3b) for $k \in V_i$, $\mathbf{n}_j \neq \mathbf{n}_k$;

(4a) for $m \in E_i$ and $k \in (E_i \cap V_m)$, $\sigma_{ij} \cap N_{im} \cap N_{ik} \neq \phi$;

(4b) for $m \in E_j$ and $k \in (E_j \cap V_m)$, $\sigma_{ij} \cap N_{jm} \cap N_{jk} \neq \phi$;

(5a) for $m \in V_i$, $\sigma_{ij} \cap N_{im} \cap \bigcap_{\substack{k \in V_i \\ k \in V_m}} N_{ik}^{0+} \subset \bigcap_{\substack{k \in V_i \\ k \notin V_m}} N_{ik}^{0+}$;

(5b) for $m \in V_j$, $\sigma_{ij} \cap N_{jm} \cap \bigcap_{\substack{k \in V_j \\ k \in V_m}} N_{jk}^{0+} \subset \bigcap_{\substack{k \in V_j \\ k \notin V_m}} N_{jk}^{0+}$;

(6a) P_i^{0+} is well-ordered with respect to σ_{ij}; and

(6b) P_j^{0+} is well-ordered with respect to σ_{ij};

then the sspec, S_{ij}, is the closure of S'_{ij} in σ_{ij}, $\mathrm{cl}' S'_{ij}$; otherwise S_{ij} is empty (ϕ). ■

[7] The clockwise direction is determined by the usual "keep your left hand on the inside wall" rule.

[8] Two sets are separated by a plane if they are contained in opposite open half-spaces of the

I conjecture that any pseudonormal pencil satisfying conditions (1), (4), and (5), is also well-ordered with respect to σ_{ij}.

Definition 5.15: Let F_i and F_j be faces of the approximating polyhedral surface. If S_{ij} is not empty, then F_i and F_j are *discrete medial involute candidates*. We also say that F_j is a discrete medial involute candidate of F_i and vice versa. ■

Definition 5.16: Let F_i be a face of the approximating polyhedral surface. A *true discrete involute* (tdi) of F_i, if one exists, is a discrete medial involute candidate of F_i, F_j, for which the minimum distance between a point of S_{ij} and the plane π_i is smallest. ■

We say that the approximating polyhedral surface "adequately" approximates the underlying smooth outline if every face has at least one discrete medial involute candidate.

Definition 5.17: An approximating polyhedral surface is *admissible* if every face has at least one discrete medial involute candidate. ■

Henceforth, we shall assume that the approximating polyhedral surface is admissible.

5.4.4.2. Properties

In this section, I prove formally that the conditions stated in Definition 5.14 are sufficient to ensure that a non-empty sspec is a closed convex polygon:

Theorem 5.2: Let F_i and F_j be faces of the approximating polyhedral surface. Then, if S_{ij} is not empty, S_{ij} is a convex closed planar polygon. ■

Essentially, Theorem 5.2 ensures that any sspec that is not empty, and is thus eligible to approximate a symmetric surface neighborhood, is a polygon rather than an unbounded region.

The proof is in three parts. First, I show that a non-empty sspec is homogeneously two-dimensional, neither a line nor a point, and that it is convex. I then show that the sspec is the intersection of two closed pseudonormal pencils with the bisector plane, and, finally, that the sspec is bounded and therefore a closed

plane.

polygon.

The intuitive notion of homogeneity is captured by the set-theoretic concept of a *regular* set[Requicha77a, Requicha78a, Kuratowski76a].

Definition 5.18: A set A is *regular* if $A = \mathrm{cl}\,\mathrm{int}\, A$. ■

Informally, the operator "cl int," sometimes called *regularization*, discards portions of the set having no interior and then "wraps" a boundary around the remainder of the set.

Lemma 5.3: If A is a convex set, then $\mathrm{int}\, A = \mathrm{int}\,\mathrm{cl}\, A$. (Theorem 12, Section 3.1,[Kelly79a]). ■

Lemma 5.4: If A is a convex set with a non-empty interior, then $\mathrm{cl}\, A = \mathrm{cl}\,\mathrm{int}\, A$. (Theorem 11, Section 3.1, [Kelly79a]). ■

Lemma 5.5: S_{ij} is regular in the plane σ_{ij}.

Proof: If $\mathrm{S}_{ij} = \phi$, the result is trivial. By Definition 5.14, S_{ij} is the closure in the bisector plane σ_{ij} of S'_{ij}. We must therefore show that in the plane σ_{ij}, $\mathrm{cl}'\mathrm{int}'\mathrm{cl}'\mathrm{S}'_{ij} = \mathrm{cl}'\mathrm{S}'_{ij}$, where primes on the closure and interior operators, cl' and int', denote closure and interior in σ_{ij}. Since planes and half-spaces are convex sets and the intersection of any number of convex sets is convex (Theorem 7, Section 3.1, [Kelly79a]), by (5.1) S'_{ij} is convex. Then, by Lemma 5.3, $\mathrm{cl}'\mathrm{int}'\mathrm{cl}'\mathrm{S}'_{ij} = \mathrm{cl}'\mathrm{int}'\mathrm{S}'_{ij}$. The result now follows immediately from Lemma 5.4. ■

We also have

Lemma 5.6: The closure of a convex set is convex. (Theorem 12, Section 2.6, [Kelly79a]). ■

Lemma 5.7: S_{ij} is convex.

Proof: S'_{ij} was shown to be convex in the proof of Lemma 5.5. The result follows from Lemma 5.6. ■

Together, Lemmas 5.5 and 5.7 show that a non-empty sspec is a convex, homogeneously two-dimensional region of the bisector plane σ_{ij}. The second part of the proof of Theorem 5.2 entails showing that a non-empty sspec is the intersection of two closed pseudonormal pencils with the bisector plane, and hence, is a closed polygonal region of the bisector plane. By definition, a non-empty sspec S_{ij} is the closure in the bisector plane σ_{ij} of the intersection with

σ_{ij} of the open pseudonormal pencils at faces F_i and F_j. Essentially, what is required is to show that in the particular case when the sspec is non-empty, the closure operator distributes over the intersection operator.

Lemma 5.8: Let F_i and F_m be vertex-adjacent faces and let F_j be any other face. If $\mathbf{n}_i \neq \mathbf{n}_j$, $\mathbf{n}_m \neq \mathbf{n}_j$, $\sigma_{ij} \neq \phi$, and $\sigma_{mj} \neq \phi$, then $\sigma_{ij} \cap \mathrm{N}_{im}$ is a line. Furthermore, $\sigma_{ij} \cap \sigma_{mj} \cap \mathrm{N}_{im} = \sigma_{mj} \cap \mathrm{N}_{im} = \sigma_{ij} \cap \mathrm{N}_{im}$.

Proof: We first show that $\sigma_{ij} \cap \mathrm{N}_{im}$ is a line. Using the definition of a bisector plane, it is not difficult to see that $\mathbf{n}_i - \mathbf{n}_m$ and $\mathbf{n}_i - \mathbf{n}_j$ are vectors normal to N_{im} and σ_{ij} respectively. Hence, $\sigma_{ij} \cap \mathrm{N}_{im} = \phi$ only if $\mathbf{n}_i - \mathbf{n}_m = c\,(\mathbf{n}_i - \mathbf{n}_j)$, for some non-zero constant c. By solving for $\mathbf{n}_i$ and taking its magnitude, it is easy to see that $\sigma_{ij} \cap \mathrm{N}_{im}$ is empty only if $\mathbf{n}_j = \mathbf{n}_m$. If $\mathbf{n}_j = \mathbf{n}_m$, σ_{mj} is empty unless $\pi_j = \pi_m$. But, by hypothesis, σ_{mj} is not empty. Therefore, $\pi_j = \pi_m$. But since vertex-adjacent faces are not coplanar, $\pi_j \neq \pi_m$. Thus, σ_{ij} and N_{im} are not parallel and must intersect in a line.

By corresponding arguments, σ_{mj} and N_{im} also intersect in a line. It remains to be shown that the two lines are identical. Let $p \in \sigma_{ij} \cap \mathrm{N}_{im}$. By definition, $d_{\pi_i}(p) = d_{\pi_j}(p)$ and $d_{\pi_i}(p) = d_{\pi_m}(p)$, which implies that $p \in \sigma_{mj}$. Thus $\sigma_{ij} \cap \mathrm{N}_{im} \subset \sigma_{mj} \cap \mathrm{N}_{im}$. An identical argument yields the converse and thus equality. ■

Lemma 5.9: For any two faces F_i and F_j, if S_{ij} is not empty, then $\mathrm{S}_{ij} = \sigma_{ij} \cap \mathrm{P}_i^{0+} \cap \mathrm{P}_j^{0+}$.

Proof: Rearranging (5.1) and substituting from Definition 5.11,

$$\mathrm{S}_{ij}' = \bigcap_{k \in \mathrm{V}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{+}) \cap \bigcap_{k \in \mathrm{V}_j} (\sigma_{ij} \cap \mathrm{N}_{jk}^{+}).$$

By Lemma 5.8, N_{ik} intersects σ_{ij} in the line $\sigma_{ij} \cap \mathrm{N}_{ik}$, implying that $\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}$ is a closed half-plane. Therefore the interior of $\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}$ in σ_{ij}, $\mathrm{int}'(\sigma_{ij} \cap \mathrm{N}_{ik}^{0+})$, is the open half-plane $\sigma_{ij} \cap \mathrm{N}_{ik}^{+}$. Thus,

$$\mathrm{S}_{ij}' = \bigcap_{k \in \mathrm{V}_i} (\mathrm{int}'(\sigma_{ij} \cap \mathrm{N}_{ik}^{0+})) \cap \bigcap_{k \in \mathrm{V}_j} (\mathrm{int}'(\sigma_{ij} \cap \mathrm{N}_{jk}^{0+})).$$

Applying the distributive property of the interior operator over intersection (Property 2.6.9, [Requicha78a]), we have

$$\mathrm{S}_{ij}' = \mathrm{int}'(\bigcap_{k \in \mathrm{V}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}) \cap \bigcap_{k \in \mathrm{V}_j} (\sigma_{ij} \cap \mathrm{N}_{jk}^{0+})).$$

Let $A = \bigcap_{k \in \mathrm{V}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}) \cap \bigcap_{k \in \mathrm{V}_j} (\sigma_{ij} \cap \mathrm{N}_{jk}^{0+})$. Since S_{ij}' is not empty (else S_{ij} would be empty), A is not empty. Moreover, since A is the intersection of planes and half-spaces, which are convex sets, and the intersection of any number of convex sets is convex (Theorem 7, Section 3.1, [Kelly79a]), by Lemma 5.4,

$$\mathrm{cl}'\,\mathrm{S}'_{ij} = \mathrm{cl}'\,\mathrm{int}'\Big(\bigcap_{k\in \mathrm{V}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}) \cap \bigcap_{k\in \mathrm{V}_j} (\sigma_{ij} \cap \mathrm{N}_{jk}^{0+})\Big)$$

$$= \mathrm{cl}'\Big(\bigcap_{k\in \mathrm{V}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}) \cap \bigcap_{k\in \mathrm{V}_j} (\sigma_{ij} \cap \mathrm{N}_{jk}^{0+})\Big).$$

Since the intersection of any number of closed sets is closed (Theorem 4, Section 1.3, [Kelly79a]), $\bigcap_{k\in \mathrm{V}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}) \cap \bigcap_{k\in \mathrm{V}_j} (\sigma_{ij} \cap \mathrm{N}_{jk}^{0+})$ is closed in $\mathbf{R}^3$, and hence, by part (3) of Lemma 5.1, it is also closed in σ_{ij}. Therefore, since the closure of a closed set is the set itself (Property 2.5.6, [Requicha78a]),

$$\mathrm{cl}'\,\mathrm{S}'_{ij} = \bigcap_{k\in \mathrm{V}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{0+}) \cap \bigcap_{k\in \mathrm{V}_j} (\sigma_{ij} \cap \mathrm{N}_{jk}^{0+}).$$

The result then follows by substituting from Definition 5.11 and rearranging terms. ■

So far, we have shown that a non-empty sspe is a closed polygonal region of the bisector plane. We complete the proof of Theorem 5.2 by showing that the region is bounded.

Theorem 5.2: Let F_i and F_j be faces of the approximating polyhedral surface. Then, if S_{ij} is not empty, S_{ij} is a convex closed planar polygon.

Proof: By Lemmas 5.8 and 5.9, S_{ij} is the intersection of a finite set of closed half-planes of σ_{ij}. Moreover, by Lemma 5.5, S_{ij} is homogeneously two-dimensional. It is therefore a polygon if it is also bounded (Theorem 3.1.3, [Grunbaum67a]). Let $A = \sigma_{ij} \cap \bigcap_{k\in \mathrm{E}_i} \mathrm{N}_{ik}^{0+}$. We show that S_{ij} is bounded by showing that A, which contains S_{ij}, is bounded. Rewrite A as $\bigcap_{k\in \mathrm{E}_i} (\sigma_{ij} \cap \mathrm{N}_{ik}^{0+})$ and apply Lemma 5.8 to see that A is the intersection of a finite set of closed half-planes of σ_{ij}, each bounded by the line of intersection between σ_{ij} and the pseudonormal plane N_{ik} through the edge e_i^k of F_i. For $t=0, \ldots, n_i-1$, let N_i^t denote the pseudonormal plane N_{ik}, $k \in \mathrm{E}_i$, that contains edge e_i^t of face F_i. We need only show that for all edges e_i^t of F_i, the line $\sigma_{ij} \cap \mathrm{N}_i^t$ intersects the line $\sigma_{ij} \cap \mathrm{N}_i^{\mathrm{CC}_i(t)}$, for then A is bounded by a closed polygon in σ_{ij}. But, this is equivalent to showing that $\sigma_{ij} \cap \mathrm{N}_i^t \cap \mathrm{N}_i^{\mathrm{CC}_i(t)}$ is not empty, which is guaranteed by part (4) of Definition 5.14. ■

In this section we have examined properties of individual sspec's, showing principally that an sspec is a convex, closed, planar polygon. Along the way we have also derived several results that will be useful below as we discuss properties of sspec's defined by neighboring faces.

5.5. A Simplified Segment Extension Procedure

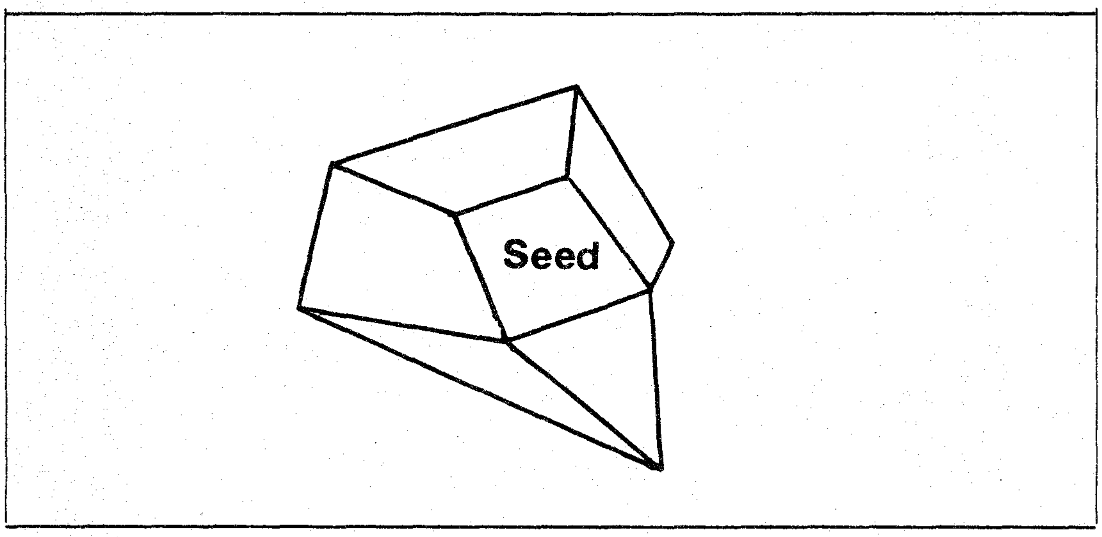
In Section 5.4, I defined and investigated some of the properties of pseudonormal pencils and symmetric surface planar element candidates, the principal components of a three-dimensional generalization of Bookstein's algorithm. Here, I use those components to construct a three-dimensional generalization of Bookstein's fsle chain extension procedure. I first give a brief overview of the three-dimensional extension procedure. Then, after proving that sspec's "fit together" into polyhedral surfaces, I give a detailed presentation of the extension procedure. In Section 5.6, I show how this extension procedure can be integrated into a complete algorithm for finding an approximate symmetric surface.

5.5.1. Overview

As described briefly in Section 5.3, the simplified segment extension procedure is to begin with a single "seed" sspe about which it grows an entire simplified segment. Not surprisingly, we face the same problem generalizing Bookstein's fsle chain extension procedure that we encountered in Chapter 3 and again in Chapter 4, namely, since a simplified segment of a three-dimensional outline is a surface, rather than a curve, there is no one-dimensional axis along which we can work. Therefore, it makes no sense to speak of extending a chain left or right. Instead, the extension procedure must extend a "seed" sspe in all directions, either depth-first or breadth-first, yielding a polyhedral surface. I present a breadth-first procedure, since I believe that it admits a more efficient implementation than the corresponding depth-first procedure. The procedure is quite simple:

(1) Initially, the polyhedral surface being constructed contains the "seed" sspe alone.

(2) About each boundary vertex of the polyhedral surface constructed thus far, attach one or more new sspe's in cyclic order, obtaining a new polyhedral surface of sspe's, as illustrated:



(3) Repeat step (2) until no new sspe's can be added due to extension failure (discussed below).

Of course, this procedure is applicable only if sspec's fit together appropriately. Therefore, before discussing this procedure in detail, we pause to prove that sspec's do indeed "fit together."

5.5.2. Sspec Intersections

Thus far, we have shown that an individual sspec is a convex, closed, planar polygon defined by the intersection of two pseudonormal pencils with the bisector plane between two faces. In this section, as a prelude to describing the extension procedure in detail, we investigate some intersection properties of two non-empty sspec's defined by a pair of vertex-adjacent faces and a third, "opposite" face. For brevity, we shall call two such sspec's *neighboring sspec's*. The principal result of the section ensures that we can construct a polyhedral surface from sspec's by yielding a solution to the following problem: Given an sspec and an edge of that sspec, is it possible to find without searching a neighboring sspec sharing that same edge? If so, how?

Let us try to achieve an intuitive understanding of the solution. By definition, an sspec, say S_{ij}, is the intersection with σ_{ij}, the bisector plane between faces F_i and F_j, of the pseudonormal pencil at face F_i and the pseudonormal pencil at face F_j. Each edge of the sspec S_{ij} must therefore be contained in the line of intersection between the bisector plane σ_{ij} and one of the pseudonormal planes at either F_i or F_j. For concreteness, pick some edge of S_{ij} and assume that it is contained in the line of intersection between σ_{ij} and the pseudonormal plane N_{im} determined by F_i and one of its neighbors, F_m. Call the line of intersection L_{ijm}.

Clearly, if some other sspec is to share that edge with S_{ij}, the bisector plane containing the other sspec must contain L_{ijm}. We have seen previously (Lemma 5.8) that the bisector plane σ_{mj} between F_m and F_j satisfies this requirement, as illustrated in Figure 5.15. Therefore, we shall argue that the sspec S_{mj} defined by faces F_m and F_j shares the edge of S_{ij} contained in N_{im}. This result, which we shall prove below, is stated formally in Theorem 5.10:

Theorem 5.10: Let F_i and F_m be vertex-adjacent faces and let F_j be a face such that S_{ij} and S_{mj} are non-empty sspec's. If $S_{ij} \cap N_{im}$ is an edge of S_{ij}, then S_{ij} and S_{mj} share the common edge $S_{ij} \cap S_{mj} = S_{ij} \cap N_{im} = S_{mj} \cap N_{im}$. ■

Thus, given an sspec and any edge of that sspec, we can find a neighboring sspec that shares the edge simply by knowing which pseudonormal plane contains the edge. This result is the basis for the extension procedure described in detail in the next section.

I now prove Theorem 5.10. I first show that if two neighboring sspec's intersect, they do so in the pseudonormal plane between the pair of neighboring faces that define the sspec's.

Lemma 5.11: If F_i and F_m are vertex-adjacent faces and F_j is another face, then $S_{ij} \cap S_{mj}$ is a subset of N_{im} and $S_{ij} \cap S_{mj} = (S_{ij} \cap N_{im}) \cap (S_{mj} \cap N_{im})$.

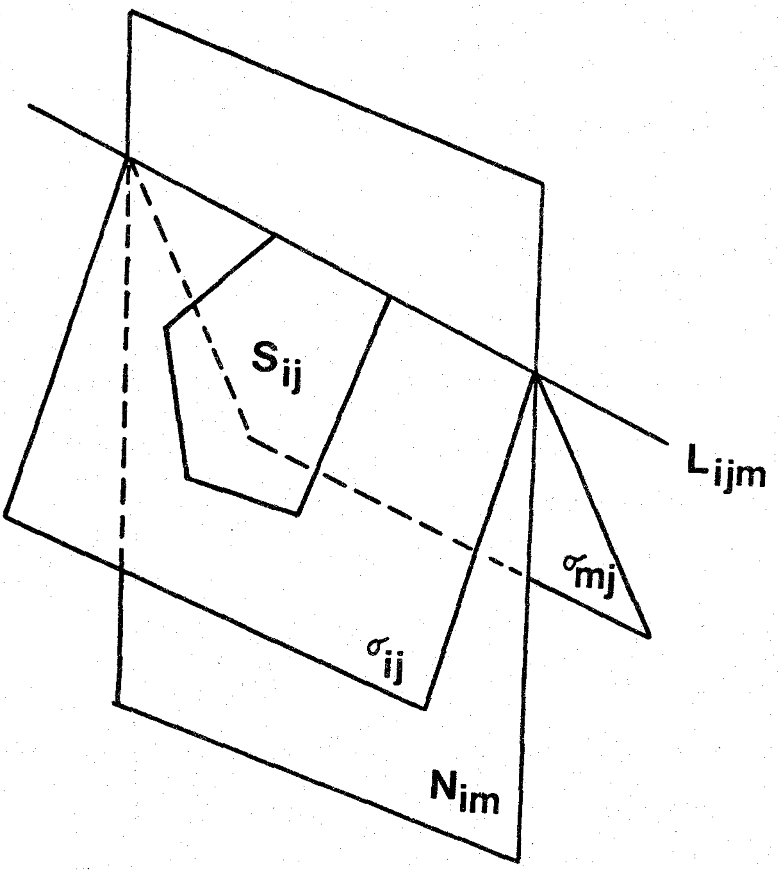


Figure 5.15.: Intersection of Bisector and Pseudonormal Planes

Proof: If either S_{ij} or S_{mj} is empty the result is trivial. Assume both are non-empty. Using Lemma 5.9,

$$S_{ij} \cap S_{mj} = \sigma_{ij} \cap \sigma_{mj} \cap P_i^{0+} \cap P_m^{0+} \cap P_j^{0+}.$$

Arguments identical to those used in the proof of Lemma 5.8 show that σ_{ij} and σ_{mj} intersect in a line. Let p be a point in $\sigma_{ij} \cap \sigma_{mj}$. Then, by definition, $d_{\pi_i}(p) = d_{\pi_j}(p)$ and $d_{\pi_m}(p) = d_{\pi_j}(p)$. Therefore, $d_{\pi_i}(p) = d_{\pi_m}(p)$, which implies that $\sigma_{ij} \cap \sigma_{mj}$ is a subset of $\sigma_{im} = N_{im}$. Hence, $S_{ij} \cap S_{mj}$ is a subset of N_{im}. Since $S_{ij} \cap S_{mj}$ is a subset of N_{im}, $S_{ij} \cap S_{mj} = (S_{ij} \cap N_{im}) \cap (S_{mj} \cap N_{im})$. ▫

Recalling that S_{ij} is a polygon in σ_{ij} and that S_{mj} is a polygon in σ_{mj}, and referring to Figure 5.15, we see that Lemma 5.11 states that if S_{ij} and S_{mj} intersect, they do so in the line L_{ijm}, where $L_{ijm} = \sigma_{ij} \cap N_{im} = \sigma_{mj} \cap N_{im}$.

Using the first of the pseudonormal pencil localness properties mentioned above, I now state and prove a condition sufficient to ensure that S_{ij} and S_{mj} have identical intersections with L_{ijm}. The pseudonormal pencils at F_i and F_m consist of the intersections of the positive pseudonormal half-spaces determined by the neighbors of F_i and F_m, respectively. The first pseudonormal pencil localness property (part (5) of Definition 5.14) implies that the intersection of L_{ijm} with the pseudonormal pencil at F_i is completely determined by the postive pseudonormal half-spaces associated with F_i and faces vertex-adjacent to both F_i and F_m. Similarly, the intersection of L_{ijm} with the pseudonormal pencil at F_m is completely determined by the positive pseudonormal half-spaces associated with F_m and faces vertex-adjacent to both F_i and F_m. Thus, to determine the intersections of S_{ij} and S_{mj} with L_{ijm}, we need only consider respectively the intersections of L_{ijm} with the positive pseudonormal half-spaces determined by F_i and faces that are vertex-adjacent to both F_i and F_m and of L_{ijm} with the positive psuedonormal half-spaces determined by F_m and faces that are vertex-adjacent to both F_i and F_m. In the following lemma, I show that S_{ij} and S_{mj} have identical intersections with L_{ijm} if for every face F_k vertex-adjacent to both F_i and F_m, the positive pseudonormal half-space of F_i and F_k intersects the same half-line of L_{ijm} as does the positive pseudonormal half-space of F_m and F_k. See Figure 5.16.

Lemma 5.12: Let F_i and F_m be vertex-adjacent faces and let F_j be a face such that S_{ij} and S_{mj} are non-empty sspec's. If for all faces F_k vertex-adjacent to both F_i and F_m, $L_{ijm} \cap N_{ik}^{0+} = L_{ijm} \cap N_{mk}^{0+}$, then $S_{ij} \cap N_{im} = S_{mj} \cap N_{im}$.

Proof: Using Lemma 5.9,

$$S_{ij} \cap N_{im} = \sigma_{ij} \cap N_{im} \cap P_i^{0+} \cap P_j^{0+}.$$

By part (5a) of Definition 5.14, $\sigma_{ij} \cap N_{im} \cap \bigcap_{\substack{k \in V_i \\ k \in V_m}} N_{ik}^{0+} \subset \bigcap_{\substack{k \in V_i \\ k \notin V_m}} N_{ik}^{0+}$. Therefore,

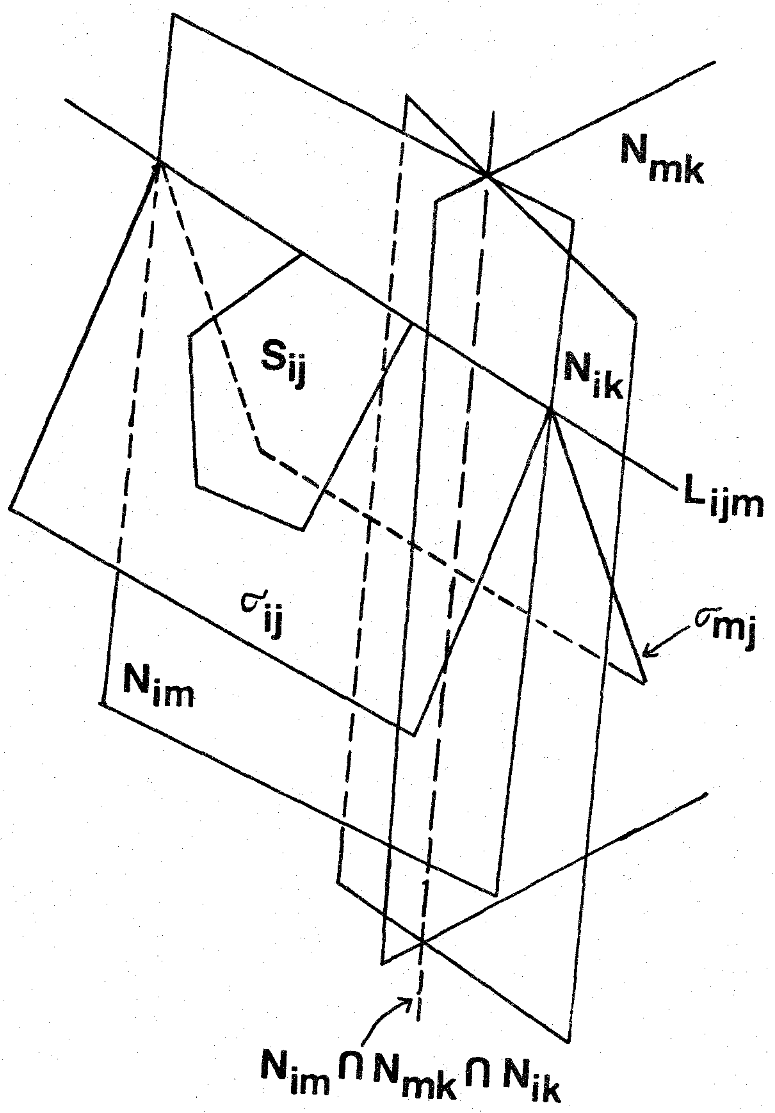


Figure 5.16.: Pseudonormal Half-space Intersections with L_{ijm}

since $P_i^{0+} = \bigcap_{k\in V_i} N_{ik}^{0+}$,

$$S_{ij} \cap N_{im} = \sigma_{ij} \cap N_{im} \cap \Big(\bigcap_{\substack{k\in V_i\\ k\in V_m}} N_{ik}^{0+} \Big) \cap P_j^{0+} .$$

Similarly,

$$S_{mj} \cap N_{im} = \sigma_{mj} \cap N_{im} \cap \Big(\bigcap_{\substack{k\in V_i\\ k\in V_m}} N_{mk}^{0+} \Big) \cap P_j^{0+} .$$

Rearranging terms,

$$S_{ij} \cap N_{im} = \bigcap_{\substack{k \in V_i \\ k \in V_m}} (\sigma_{ij} \cap N_{im} \cap N_{ik}^{0+}) \cap P_j^{0+}, \text{ and}$$

$$\begin{aligned} S_{mj} \cap N_{im} &= \bigcap_{\substack{k \in V_i \\ k \in V_m}} (\sigma_{mj} \cap N_{im} \cap N_{mk}^{0+}) \cap P_j^{0+} \\ &= \bigcap_{\substack{k \in V_i \\ k \in V_m}} (\sigma_{ij} \cap N_{im} \cap N_{mk}^{0+}) \cap P_j^{0+}, \end{aligned}$$

where the last step follows from Lemma 5.8. Comparing these expressions for $S_{ij} \cap N_{im}$ and $S_{mj} \cap N_{im}$ and substituting $L_{ijm} = \sigma_{ij} \cap N_{im}$ establishes the lemma. ■

To prove that S_{ij} and S_{mj} share a common edge, we now need only show that for every face F_k vertex-adjacent to both F_i and F_m, the positive pseudonormal half-space of F_i and F_k intersects the same half-line of L_{ijm} as does the positive pseudonormal half-space of F_m and F_k. I first establish that the pseudnormal planes N_{ik} and N_{mk} indeed intersect the pseudnormal plane N_{im} in the same line, as Figure 5.16 illustrates. Then, I shall use the second localness property of pseudonormal pencils (part (6) of Definition 5.14) to complete the proof that S_{ij} and S_{mj} share a common edge.

Lemma 5.13: If face F_k is vertex-adjacent to both F_i and F_m, then the pseudonormal planes N_{im}, N_{ik}, and N_{mk} intersect in a common line, that is, $N_{im} \cap N_{ik} = N_{im} \cap N_{mk}$.

Proof: Since F_i, F_k, and F_m all share a common vertex, both N_{im} and N_{ik} contain that vertex. Hence, $N_{im} \cap N_{ik}$ is not empty. Let p be a point in $N_{im} \cap N_{ik}$. Then, by Definition 5.10, $d_{\pi_i}(p) = d_{\pi_k}(p)$ and $d_{\pi_i}(p) = d_{\pi_m}(p)$. Therefore, $d_{\pi_m}(p) = d_{\pi_k}(p)$ as well. Thus $p \in N_{mk}$, implying that $N_{im} \cap N_{ik} \subset N_{im} \cap N_{mk}$. A similar argument yields $N_{im} \cap N_{mk} \subset N_{im} \cap N_{ik}$, which establishes the result. ■

I complete the proof that S_{ij} and S_{mj} share a common edge, by using the second localness property of pseudonormal pencils (part (6) of Definition 5.14) to show that for any face F_k vertex-adjacent to both F_i and F_m, not only do N_{ik} and N_{mk} intersect L_{ijm} at the same point, as shown in Figure 5.16, but their positive half-spaces intersect the same half-line of L_{ijm}.

Theorem 5.10: Let F_i and F_m be vertex-adjacent faces and let F_j be a face such that S_{ij} and S_{mj} are non-empty sspec's. If $S_{ij} \cap N_{im}$ is an edge of S_{ij}, then S_{ij} and S_{mj} share the common edge $S_{ij} \cap S_{mj} = S_{ij} \cap N_{im} = S_{mj} \cap N_{im}$.

Proof: To prove the theorem, we establish the hypothesis of Lemma 5.12. Let F_k be any face vertex-adjacent to both F_i and F_m. The closed half-spaces N_{ik}^{0+} and N_{mk}^{0+} each define a closed half-line of L_{ijm}. It is sufficient to show that the two half-lines are identical. By Lemma 5.13, we know that the two half-lines $N_{ik}^{0+} \cap L_{ijm}$ and $N_{mk}^{0+} \cap L_{ijm}$ have identical endpoints. The situation in the σ_{ij} and σ_{mj} planes as viewed from directly above L_{ijm} is shown in Figure 5.17 (cf. Figure 5.16). The small arrows near pseudonormal plane labels indicate the positive pseudonormal half-spaces of the corresponding pseudonormal planes.

It remains to be determined which half-space of N_{mk} is the positive half-space. We do so by using the well-ordered property (Definitions 5.12 and 5.13) of the pseudonormal pencils at F_i and F_m. Consider the partial pseudonormal pencil at F_i defined by the intersection of the positive half-spaces of N_{im} and N_{ik}. As we traverse clockwise the intersection of the partial

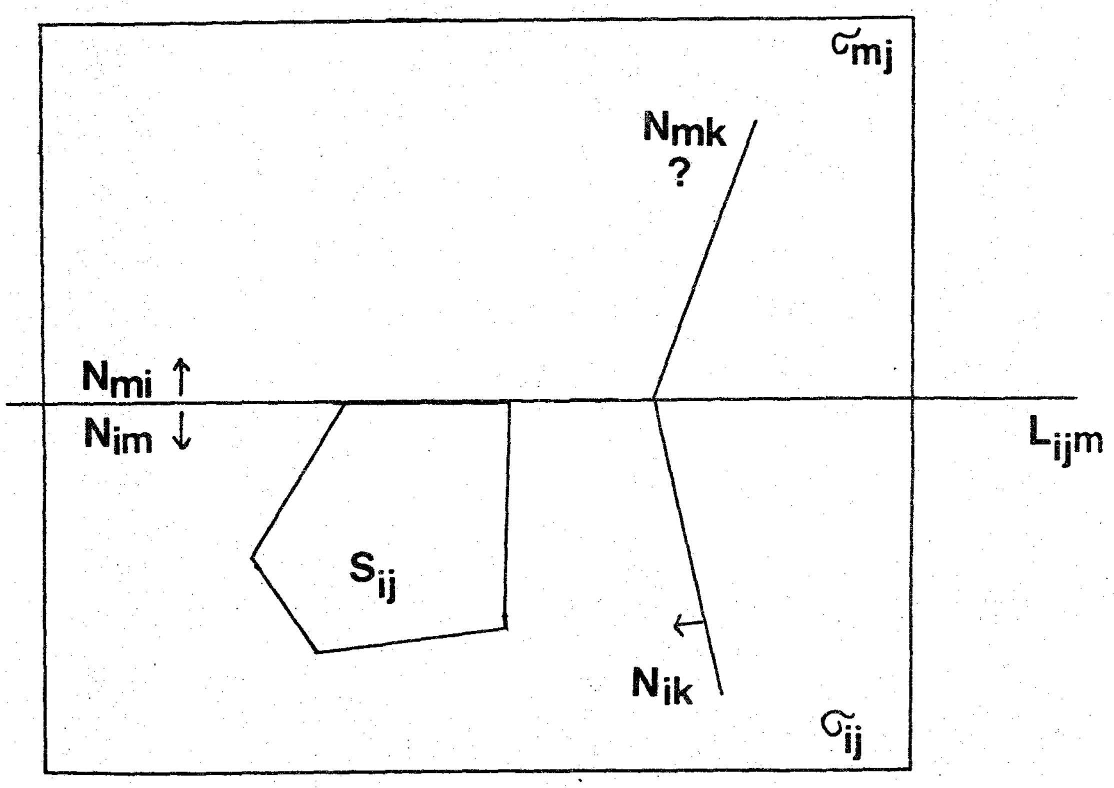
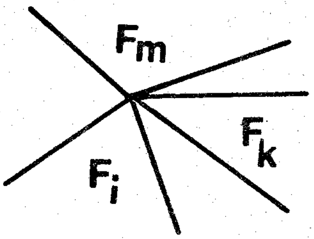


Figure 5.17.: Diagram for Proof of Theorem 5.10

pseudonormal boundary with σ_{ij}, the pseudonormal planes N_{im} and N_{ik} are encountered in order. Thus, the associated neighbors of F_i must occur in the order F_m followed by F_k. Therefore, since the pseudonormal pencil at F_i must be well-ordered with respect to σ_{ij} (part (6) of Definition 5.14), faces F_i, F_m, and F_k must be ordered about their common vertex as shown:



Now consider the partial psuedonormal pencil at F_m defined by the intersection of the positive half-spaces of N_{mi} and N_{mk}. Since the pseudonormal pencil at F_m must be well-ordered with respect to σ_{mj}, to be consistent with the ordering of F_i, F_m, and F_k shown above, as we traverse clockwise the intersection of the partial pseudonormal boundary with σ_{mj}, the associated neighbors of F_m must occur in the order F_k followed by F_i. Therefore, the positive half-space of N_{mk} must lie to the left of N_{mk} in Figure 5.17, thus confirming that the half-lines of L_{ijm} determined by N_{ik}^{0+} and N_{mk}^{0+} are identical. Analogous arguments apply when N_{ik}^{0+} lies to the right of N_{ik}.

Thus, the hypothesis of Lemma 5.12 is satisfied, implying that $S_{ij} \cap N_{im} = S_{mj} \cap N_{im}$. That $S_{ij} \cap S_{mj} = S_{ij} \cap N_{im} = S_{mj} \cap N_{im}$ follows directly from this result and Lemma 5.11. ■

Thus, given an sspec and an edge of that sspec, Theorem 5.10 tells us how to find without searching an edge-adjacent neighboring sspec. Since $S_{ij} \cap N_{im} = S_{mj} \cap N_{im}$, it follows that the shared edge is an entire edge of each.

Theorem 5.10 ensures that if we carry out the sspec extension procedure sketched in Section 5.5.1, the result will be a polyhedral surface without holes.

5.5.3. Abstract Data Types

To construct a simplified segment approximation, the extension algorithm sketched in Section 5.5.1 manipulates both the topology and the geometry of collections of convex planar polygons (sspec's). Many different data structures that maintain sufficient information to perform the extension algorithm can be devised. They differ primarily in the amount and type of redundant information maintained about the relationships among the polygons. Usually, data structures that maintain the most redundant information require the least computation. Rather than evaluate the redundancy vs. efficiency tradeoff under some arbitrary assumptions, in this section I present specifications for several abstract data types that provide the necessary capabilities; I do not discuss their implementation. However, assuming that there is an upper bound on the number of vertices in each sspe,[9] all of the operations defined here can be implemented in constant time and space using the "winged-edge" polyhedron data structure[Baumgart75a, Newell79a].

Each abstract data type is specified by a list of access functions together with the domain and range of each and a description of the semantics of each function. With but a few obvious exceptions, the functions are typical of those one would expect to find in any geometric modeling package based on polyhedra.[10] I adopt several notational conventions similar to those used in [Guttag78a]:

(1) Data type names appear in italics.

(2) Non-italicized lowercase symbols are free variables of a type that either is clear from context or is specified in a **declare** statement.

(3) Function names appear in uppercase.

(4) The domain of each function is specified by a list of data types, separated by commas, contained within matched parentheses. The range appears to the right of an arrow (→).

All program fragments are written in "pidgin-Algol," as described in Section 1.8 of [Aho74a].

[9] Such an upper bound follows immediately from an upper bound on the number of faces that can be vertex-adjacent to any face.

[10] See e.g. [Baumgart75a] or [Eastman77a].

Boundary Face Pair

A *bdyface_pair* instance denotes a pair of faces in the approximating polyhedral surface.

Sspec Vertex

An *sspec_vertex* contains the coordinates of an sspe vertex in some unspecified coordinate system. No operations other than instantiation may be performed on an *sspec_vertex* unless it is part of an *sspec* (described below).

Sspec Vertex Queue

The sspec vertex queue is a first-in-first-out queue of sspec vertices.

Syntax:

INITQ() →
ENQ(*sspec_vertex*) →
DEQ() → *sspec_vertex*
EMPTYQ() → *boolean*

Semantics:

INITQ
  INITQ() initializes the queue to an empty queue.

ENQ
  ENQ(v) places vertex v last on the queue.

DEQ
  DEQ() removes the first vertex on the queue from the queue and returns it.

EMPTYQ
  EMPTYQ() returns **true** if and only if the queue is empty.

Sspec Edge

An *sspec_edge* is the edge of an sspe defined by two sspe vertices. No operations other than instantiation may be performed on an *sspec_edge* unless it is part of an *sspec* (described below).

Sspec

An *sspec* instance denotes an oriented sspec.

Syntax:

MAKESSPEC(*bdyface_pair*) → *sspec*
OPPBF(*sspec*) → *bdyface_pair*
CCV(*sspec*, *sspec_vertex*) → *sspec_vertex*
CV(*sspec*, *sspec_vertex*) → *sspec_vertex*
FORCCV(*sspec*, *procedure*) →
FORCV(*sspec*, *procedure*) →
CCE(*sspec*, *sspec_vertex*) → *sspec_edge*
CE(*sspec*, *sspec_vertex*) → *sspec_edge*
ADJBF(*sspec*, *sspec_edge*) → *bdyface_pair*

Semantics:

MAKESSPEC

MAKESSPEC(x) returns the sspec defined by the pair of faces specified by x. If the specified sspec does not exist, **empty** is returned.

OPPBF

OPPBF(x) returns the *bdyface_pair* that determines sspec x. More precisely, if MAKESSPEC(x) ≠ **empty**, OPPBF(MAKESSPEC(x)) = x; otherwise, its value is undefined.

CCV

CCV(x, v) returns the vertex counterclockwise adjacent to vertex v on sspec x.

CV

CV(x, v) returns the vertex clockwise adjacent to vertex v on sspec x.

FORCCV

FORCCV(x, P) calls procedure P once for each vertex of sspec x, passing the vertex as a parameter to P. Successive calls to p are passed successive vertices in counterclockwise order. FORCCV(x, P) is equivalent to:

```
$v_0$ ← arbitrary vertex of x;
v ← $v_0$;
repeat
   begin
    P(v);
    v ← CCV(v);
   end
until v = $v_0$;
```

FORCV

FORCV(x, P) calls procedure P once for each vertex of sspec x, passing the vertex as a parameter to P. Successive calls to P are passed successive vertices in clockwise order. FORCV(x, P) is equivalent to:

```
$v_0 \leftarrow$ arbitrary vertex of x;
$v \leftarrow v_0$;
repeat
   begin
    P(v);
    $v \leftarrow$ CV(v);
   end
until $v = v_0$;
```

CCE

CCE(x, v) returns the edge of x defined by v and CCV(x, v).

CE

CE(x, v) returns the edge of x defined by v and CV(x, v).

ADJBF

ADJBF(x, e) returns the *bdyface_pair* that specifies the sspec that shares edge e of sspe x. The appropriate sspec is determined by applying Theorem 5.10 as follows. Say that x represents the sspec S_{ij} and that e represents an edge of S_{ij}. Then, either e is $S_{ij} \cap N_{im}$ where F_m is a neighbor of F_i, or e is $S_{ij} \cap N_{jm}$ where F_m is a neighbor of F_j. In the former case, ADJBF(e, x) returns the *bdyface_pair* that specifies S_{mj}, in the latter case it returns the *bdyface_pair* that specifies S_{im}.

Several of these functions are illustrated in Figure 5.18.

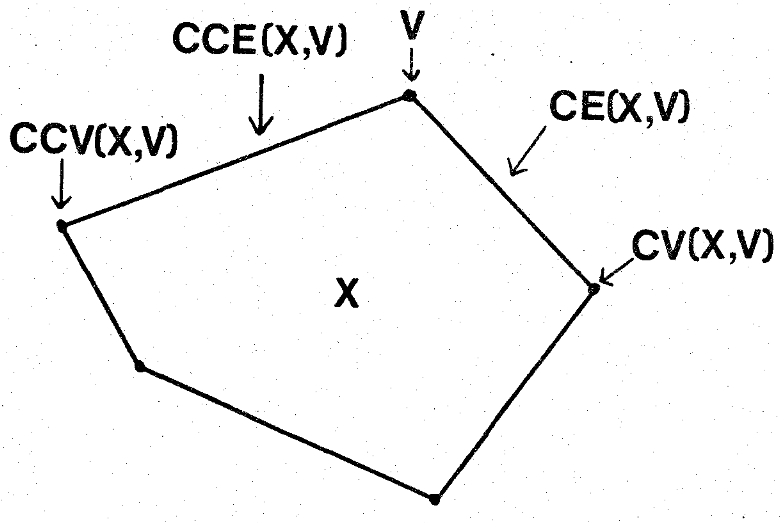


Figure 5.18.: Operations on the *sspec* Data Type

Polyhedral Surface

A *polysurf* instance denotes a collection of sspec's that form a polyhedral surface.

Syntax:

INITSURF(*sspec*) → *polysurf*
EF1(*polysurf*, *sspec_edge*) → *sspec*
EF2(*polysurf*, *sspec_edge*) → *sspec*
EADJF(*polysurf*, *sspec*, *sspec_edge*) → *sspec*
EXTCCF(*polysurf*, *sspec_vertex*) → *sspec*
EXTCF(*polysurf*, *sspec_vertex*) → *sspec*
BDYVERT(*polysurf*, *sspec_vertex*) → *boolean*
EDGEMERGE(*polysurf*, *sspec_edge*, *sspec*) → *polysurf*

Semantics:

INITSURF

INITSURF(x) returns a *polysurf* comprised of the single sspec x.

EF1, EF2

EF1(p, e) returns one of the two possible sspec's sharing edge e and EF2(p, e) returns the other sspec. If only one sspec in p contains e, then either EF1 or EF2, but not both, returns **empty**. These functions are so-called "hidden functions," used only in describing other functions.

EADJF

EADJF(p, x, e) returns the sspec in p that shares edge e of sspe x. If e is a boundary edge, **empty** is returned. Note that EADJF(p, EF1(p, e), e) = EF2(p, e) and EADJF(p, EF2(p, e), e) = EF1(p, e).

EXTCCF

EXTCCF(p, v) returns the counterclockwise most sspec in p about vertex v. More precisely, EXTCCF(p, v) returns the sspec x, if it exists, such that EADJF(p, x, CE(v, x)) = **empty**. If no such sspec exists, EXTCCF(p, v) returns an arbitrary sspec having v as a vertex.

EXTCF

EXTCF(p, v) returns the clockwise most sspec in p about vertex v. More precisely, EXTCF(p, v) returns the sspec x, if it exists, such that EADJF(p, x, CCE(v, x)) = **empty**. If no such sspec exists, EXTCF(p, v) returns an arbitrary sspec having v as a vertex.

BDYVERT

BDYVERT(p, v) returns **true** if and only if vertex v of p is a boundary vertex (Definition 5.5).

EDGEMERGE

EDGEMERGE(p, e, x) adds sspe x to polysurf p along edge e of p. This routine may be invoked only if:

(1) Either EF1(p, e) = **empty** or EF2(p, e) = **empty**, but not both; and

(2) If EF1(p, e) = x_1 ≠ **empty**, x and x_1 must share edge e. Otherwise, if EF2(p, e) = x_2 ≠ **empty**, x and x_2 must share edge e.

Invoking EDGEMERGE(p, e, x) has the following effects:

(1) The edge of sspec x that is shared by edge e of p becomes identical to e for comparison purposes. Similarly, the vertices of that edge become identical to the vertices of edge e.

(2) The values returned by subsequent calls of the functions EF1, EF2, EXTCCF, EXTCF, or EADJF are possibly changed. If before invoking EDGEMERGE, EF1(p, e) returned **empty**, then afterward EF1(p, e) returns x and EADJF(p, x, e) returns EF2(p, e). Similarly, if before invoking EDGEMERGE, EF2(p, e) returned **empty**, then afterward EF2(p, e) returns x and EADJF(p, x, e) returns EF1(p, e). Further, if before invoking EDGEMERGE, e was in EXTCF(p, v), v a vertex of e, then afterward EXTCF(p, v) = x. If e was in EXTCCF(p, v), then afterward EXTCCF(p, v) = x.

5.5.4. Extension Procedure

Using the sspec intersection properties proved in Section 5.5.2 and the abstract data types described in the previous section, in this section I give a detailed description of the extension procedure sketched above. Recall from Section 5.5.1 that the extension procedure consists of several simple steps:

(1) Initially, the polyhedral surface being constructed contains the "seed" sspe alone.

(2) About each boundary vertex of the polyhedral surface constructed thus far, attach one or more new sspe's in cyclic order, obtaining a new polyhedral surface of sspe's.

(3) Repeat step (2) until no new sspe's can be added due to extension failure (discussed below).

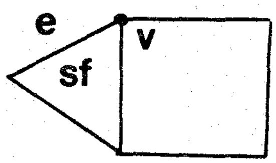
The extension procedure consists of two subroutines, MAKE_SIMP_SEG and VGROW, given as "pidgin-Algol" procedures in Figures 5.19 and 5.20 respectively. The extension procedure is invoked by calling MAKE_SIMP_SEG, passing it a "seed" sspe (obtained by search, as described in the next section) as parameter. It creates a polyhedral surface consisting of the "seed" alone (statement 1) and then inserts the vertices of the "seed" in clockwise order into a first-in, first-out queue (statement 3). Throughout the execution of the procedure, the

```
procedure MAKE_SIMP_SEG(init_sspe):
begin
    declare init_sspe sspec;

    declare simp_seg polysurf;

    comment Insert initial sspe into simplified segment;
1   simp_seg ← INITSURF(init_sspe);

    comment Place all vertices of the initial sspe on the queue of vertices to ex-
        plore;
2   INITQ();
3   FORCV(ENQ, init_sspe);

    comment Grow the segment by generating all sspe's that share each vertex
        in the queue;
4   while ¬EMPTYQ() do
5       VGROW(simp_seg, DEQ());

6   return simp_seg;
end
```

Figure 5.19.: MAKE_SIMP_SEG

vertex queue will contain all boundary vertices of the polyhedral surface yet to be processed. After initializing the vertex queue, MAKE_SIMP_SEG calls VGROW to process each vertex by attaching to the vertex all of the sspec's that are adjacent to the vertex but not already present (statements 4-5).

Upon being invoked to process a vertex v, VGROW finds (statements 2-3) the edge counterclockwise from v of the clockwise-most sspe about v, in the illustration below, edge e of sspe sf:



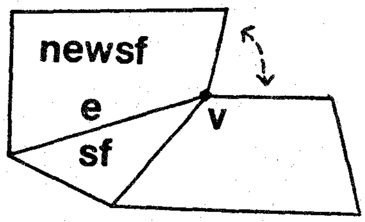
Then, using Theorem 5.10 to determine which pair of faces determine the sspec

```
procedure VGROW(s, v):
begin
    declare s polysurf;
    declare v sspec_vertex;

    declare sf sspec;
    declare newsf sspec;
    declare e sspec_edge;
    declare tempv sspec_vertex;

    comment Add all sspe's vertex−adjacent to vertex v in simplified segment s;
    while BDYVERT(s, v) do
        begin
            sf ← EXTCF(s, v);
            e ← CCE(sf, v);

            comment If the sspe to be added already exists, just merge it in;
            if ADJBF(sf, e) = OPPBF(EXTCCF(s, v)) then
                begin
                    EDGEMERGE(s, e, EXTCCF(s, v))
                    return
                end

            newsf ← MAKESSPEC(ADJBF(sf, e));
            if newsf ≠ empty then
                begin
                    comment Merge the new sspe into the simplified segment;
                    EDGEMERGE(s, e, newsf);
                    if ADJBF(newsf, CCE(newsf, v)) = OPPBF(EXTCCF(s, v)) then
                        EDGEMERGE(s, CE(EXTCCF(s, v), v), newsf);

                    comment Add vertices of the new sspe not common with the
                        original sspe to the queue of vertices to explore;
                    tempv ← CV(newsf, CV(newsf, v));
                    while EADJF(s, newsf, CE(newsf, tempv)) = empty do
                        begin
                            ENQ(tempv);
                            tempv ← CV(tempv);
                        end
                end

            else
                begin
                    comment Extension failure;
                    Mark edge e with failure mode A or B
                    return
                end
        end
end
```

Figure 5.20.: VGROW

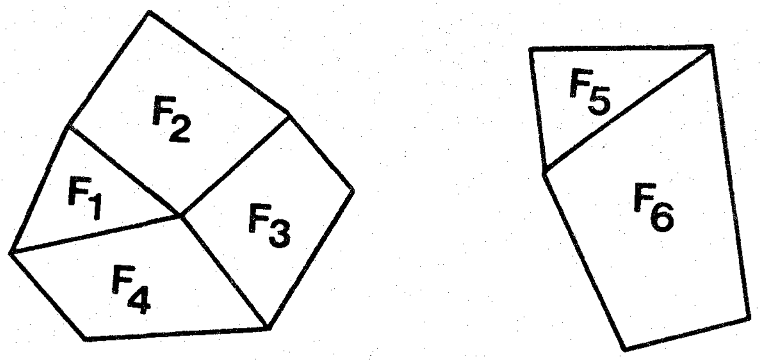
that shares edge e with sf, VGROW first determines whether that sspec is the counterclockwise most sspe about v (statement 4). If it is, then that sspec is merged with sspe sf along edge e (statement 5) and VGROW is finished. Otherwise, VGROW calls MAKESSPEC to attempt to construct the sspec newsf neighboring sf (statement 7). If the appropriate sspec is empty, extension has failed and the call to VGROW terminates. We shall discuss extension failure below.

If it is not empty, newsf must be merged into the polyhedral surface (statements 9-11) as illustrated below:

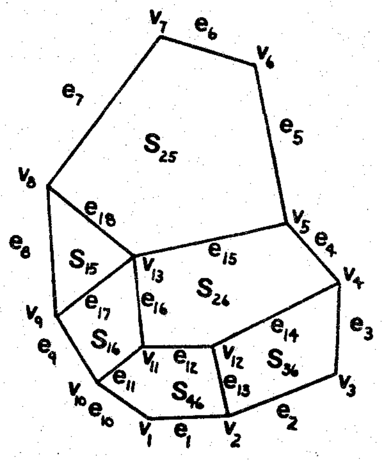


First, edge e of sspec newsf must be merged with edge e of sspe sf (statement 9). Then, a test must be made to determine whether the two edges indicated by the dotted arrows in the illustration above are identical (statement 10); if so, they too must be merged. This test requires no numerical comparison, rather, Theorem 5.10 is used to determine whether the two sspec's involved share those edges. Finally, any new boundary vertices must be inserted last into the queue of vertices yet to be processed (statements 12-15). This whole process repeats until either vertex v is completely surrounded by sspe's and therefore is no longer a boundary vertex, or until extension fails (statements 16-17). In the latter case, additional sspe's may be added about vertex v by later invocations of VGROW.

I now illustrate the major steps of extension with a simple example. Let F_1, F_2, F_3, and F_4 be vertex-adjacent faces as shown below on the left, and let F_5 and F_6 be edge-adjacent faces "opposite" the others, as shown on the right:



Further, we shall assume that the geometry is such that the portion of the simplified segment approximation produced by the opposition of F_1 through F_4 with F_5 and F_6 is as shown:



The labels placed on the edges and vertices are for ease of reference only and carry no further meaning.

The extension procedure is begun by calling MAKE_SIMP_SEG with a "seed" sspe, say S_{26}, as parameter. After statements 1 and 2 are executed, the vertex queue might contain,[11] in first to last order, v_{13}, v_5, v_4, v_{12}, and v_{11}. Then, for each vertex on the queue, statements 4 and 5 invoke VGROW to completely surround the vertex with sspe's. Figure 5.21 shows the queue contents and polyhedral surface before the first call and after the first five calls to VGROW. Subsequent calls to VGROW have no effect other than depleting the queue and marking polyhedral surface boundary edges when further extension fails.

Extension failure in three-dimensions is much like in two-dimensions. Extension fails whenever the call to MAKESSPEC in statement 7 of VGROW returns **empty**. For concreteness, let us say that the *bdyface_pair* returned by the call to ADJBF in statement 6 represents the pair of faces F_i and F_j. There are three causes of extension failure:

(1) The sspec S_{ij} is empty because condition (2) of Definition 5.14 does not obtain, that is, F_i and F_j are edge-adjacent. Following Bookstein, I call this *extension failure by mode A.*

(2) The sspec S_{ij} is empty because the intersection with the bisector plane σ_{ij} of the pseudonormal pencil at F_i or at F_j empty. Again following Bookstein, I call this *extension failure by mode B.* Extension past a branch curve implies eventual failure by mode B, for Bookstein's analogous two-dimensional argument (Section 5.2.2) also holds in three dimensions.

(3) The sspec S_{ij} is empty because at least one of the conditions of Definition 5.14 other than condition (2) does not obtain and hence, the approximating polyhedral surface is not admissible. We shall not consider this case further.[12]

[11] The exact contents of the queue depend on which vertex of S_{26} the implementation of FORCV first returns.

[12] However, see section 3 of [Bookstein79a] for a description of Bookstein's *ad hoc* procedure for

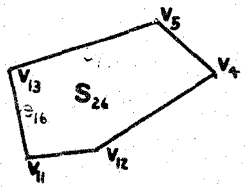
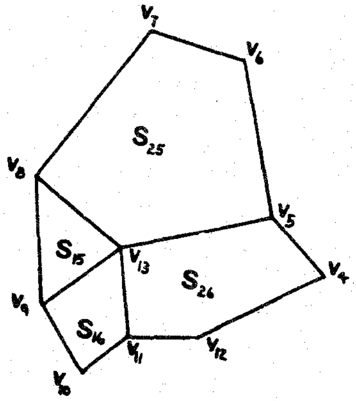
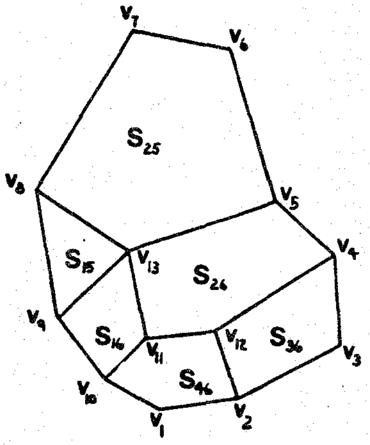
QUEUE	POLYHEDRAL SURFACE
$V_{13}, V_5, V_4, V_{12}, V_{11}$	 V_5 V_4 V_{13} S_{24} V_{11} V_{12}
$V_5, V_4, V_{12}, V_{11}, V_{10}, V_9,$ V_8, V_7, V_6	 V_7 V_6 S_{25} V_8 V_5 S_{15} V_{13} S_{24} V_4 V_9 S_{16} V_{11} V_{12} V_{10}
$V_4, V_{12}, V_{11}, V_{10}, V_9, V_8,$ V_7, V_6	NO CHANGE
$V_{12}, V_{11}, V_{10}, V_9, V_8, V_7, V_6$	NO CHANGE
$V_{11}, V_{10}, V_9, V_8, V_7, V_6, V_3,$ V_2, V_1	 V_7 V_6 S_{25} V_8 V_5 V_4 S_{15} V_{13} S_{24} V_9 S_{16} V_{11} V_{12} S_{36} V_3 S_{46} V_{10} V_1 V_2

Figure 5.21.: Illustration of Calls to VGROW

Thus, given a "seed" sspe, the extension procedure described in this section produces a polyhedral surface with boundary comprised of sspe's. Further-

closing the gaps formed by two-dimensional extension failure. Though I have not investigated the is-more, each boundary edge of the polyhedral surface is marked with the mode of extension failure at that edge, either A or B, corresponding respectively to end curves and extension past branch curves of the true symmetric surface.

5.6. The Three-dimensional Algorithm

We now have three-dimensional generalizations of each component of Bookstein's two-dimensional algorithm. Not surprisingly, the structure of the complete algorithm for computing a discrete approximation to the symmetric surface of a three-dimensional figure is almost identical to the structure of Bookstein's corresponding two-dimensional algorithm. Therefore, in this section, I only sketch the structure of the three-dimensional algorithm, assuming familiarity with Bookstein's algorithm as described in Section 5.2.2.

The first task is to find a "seed" sspe. The algorithm begins by picking an arbitrary face of the approximating polyhedral surface and finding, by exhaustive search, one of its true discrete involutes (Definition 5.16). The sspe thereby determined serves as the initial "seed".

Starting with the initial "seed" the algorithm carries out the following steps:

(1) Construct a polyhedral surface containing the "seed" by passing the "seed" to MAKE_SIMP_SEG.

(2) Scan the boundary edges of the polyhedral surface constructed in step (1). If all boundary edges are marked as failure by mode A, there is no evidence for a branch curve. Otherwise, each connected chain of boundary edges marked as failure by mode B is evidence of extension past a true branch curve. For each such connected chain determine the actual location of the branch curve by the following steps (cf. Section 5.2.2):

sue, I suspect that a similar *ad hoc* procedure can be devised for closing gaps in three dimensions.

(a) Choose one edge of the chain and call ADJBF to determine the pair of faces that would have defined an sspec attached to that edge had it not been empty.

(b) Choose one face of the pair determined in (a) and find one of its true discrete involutes by exhaustive search. That face and the chosen true discrete involute determine a new "seed" sspe.

(c) Call MAKE_SIMP_SEG to compute the branch polyhedral surface containing this new "seed."

(d) If some sspe of the branch polyhedral surface intersects an sspe in the polyhedral surface determined in step (1) and the two sspe's are determined by a common face, say sspe's S_{ij} and S_{mj} (F_i, F_j, and F_m not adjacent) determined by the common face F_j, the branch curve has been located.[13] The other branch polyhedral surface that meets at the branch curve can be constructed by returning to step (1) using sspe S_{im} as "seed."[14]

(e) If, on the other hand, no such intersection occurs, return to step (2) using the branch polyhedral surface.

The entire algorithm terminates when all extensions terminate in failure by mode A.

5.7. Summary

I have presented a three-dimensional generalization of Bookstein's two-dimensional algorithm, using the same basic components and structure. I view the principal contribution of this chapter, then, as defining pseudonormal

[13] In practice, this check for intersection would be made in MAKE_SIMP_SEG as each new sspe was added. If each face of the approximating polyhedral surface is marked whenever an sspe defined by that face is generated, the intersection test does not require any extensive searching.

[14] That S_{im} exists is easily shown using a direct analog of Bookstein's corresponding argument. I have not proven formally that all three branch surfaces meet in a common branch curve, though a somewhat more complex version of Bookstein's argument should suffice.

pencils and symmetric surface planar elements, and proving that they have the properties necessary to be used in the same manner as Bookstein's fsle's.

5.8. Unsolved Problems and Research Directions

Here, as in previous chapters, both theoretical and applied work remains to be done. I believe that I have described the algorithm in sufficient detail so that a programmer familiar with the basic algorithms and techniques of computational geometry[15] and an awareness of the pitfalls of numerical computing could implement it. There are, however, several implementation issues that will need to be addressed:

(1) What data structures are most appropriate for implementing the abstract data types described in Section 5.5.3? The most likely candidate is one of the many variations of Baumgart's "winged-edge polyhedron" data structure[Baumgart75a, Newell79a].

(2) What are appropriate representations for faces of the approximating polyhedral surface and for pseudonormal pencils? The choice of appropriate representations depends upon the operations to be performed. Computing an individual sspe requires two primitive operations: finding bisector planes and computing the intersections of pseudonormal pencils with the appropriate bisector plane. Finding bisector planes is particularly simple and rapid if the planes containing faces of the approximating polyhedral surface are represented in affine coordinates, that is, as a unit vector normal to the plane and distance along that vector to the coordinate system origin. The necessary intersections can be computed in asymptotically optimal time using algorithms such as those described by Shamos[Shamos76a, Shamos78a] and Brown[Brown79a]. However, there is

[15] See, for example, Shamos's extensive (but very readable) treatise[Shamos78a] or his treatment of geometric intersection problems[Shamos75a, Shamos76a].

a particularly simple algorithm, reentrant polygon clipping[Sutherland74a], that, while not asymptotically optimal, is very fast in practice.

A number of theoretical issues also remain to be addressed:

(1) How can we approximate the radius function, symmetric surface, and boundary surface curvatures necessary for applying the simplified segment partitioning techniques described in Chapter 4? Since there are well-known expressions for curvatures at vertices and edges of polyhedral surfaces[Banchoff70a, Brehm81a], the problem reduces to one of interpolating from curvature values at vertices and edges. I have not investigated such interpolation schemes.

(2) Though I have given intuitive arguments that sspe's approximate neighborhoods of the symmetric surface, neither Bookstein nor I have given a formal proof that the approximation produced by his algorithm or by my three-dimensional generalization converges to the true symmetric axis as the approximating polyhedral surface converges to the underlying outline. See Section 6.3 of [Kelly79a] and [Brehm81a] for examples of metrics and techniques that might be useful in such a proof.

CHAPTER 6

SUMMARY AND DIRECTIONS FOR FUTURE WORK

Building upon Blum's seminal idea, I have begun to develop a three-dimensional structural shape description methodology. In this, the final chapter, I shall review the contributions of this dissertation and outline in broad fashion directions for further research.

6.1. Summary

In Chapter 1, I introduced three shape description paradigms—represent, then discard; decomposition; and prototypes—and suggested that most shape description techniques are elaborations of these paradigms. In particular, I believe that Blum's two-dimensional shape description methodology, as reviewed in Chapter 2, exploits simultaneously and naturally two of these three paradigms: represent, then discard and decomposition. In so doing, it provides an attractive mechanism for dealing with the crucial tradeoff between stability and sensitivity, largely because the symmetric axis transform makes it possible to decouple stable, constant figure properties from properties sensitive to subtle variations.

My work to generalize Blum's two-dimensional methodology to three dimensions consists of three parts, reported in Chapters 3, 4, and 5 of this dissertation. First, I have sought an understanding of the geometry of the three-dimensional symmetric axis transform. Second, I have used this understanding to generalize to three dimensions Blum's techniques for partitioning two-dimensional symmetric axes into width shapes, axis shapes, and boundary shapes. Finally, I have generalized from two to three dimensions Bookstein's algorithm for computing a discrete approximation to the symmetric axis transform.

The three-dimensional generalization of Blum's symmetric axis transform defines a unique, coordinate-system-independent decomposition of a figure into disjoint, two-sided pieces, each with its own simplified segment and associated boundary surfaces. In Chapter 3, I have defined measures of the radius function and have shown how these measures and the symmetric surface curvatures are related to the boundary surface curvatures. In particular, I have shown that the Gaussian and mean curvatures of the boundary surfaces are determined by nine measures, each with a geometric interpretation:

(1) the symmetric surface curvature as determined by two principal curvatures and a principal direction;

(2) the radius curvature as determined by two principal curvatures and a principal direction;

(3) directional derivatives of the radius function as determined by the angles between either boundary normal and the two symmetric surface principal directions; and

(4) the radius function itself.

These measures, and the curvature relationship derived from them, subsume the two-dimensional measures and curvature relationship given by Blum.

In Chapter 4, beginning with the result of the unique figure decomposition induced by the three-dimensional symmetric axis transform, I have used the aforementioned measures, together with the relationships among them, to propose a further decomposition into primitives drawn from three separate, but not completely independent, primitive sets: ***width primitives***, based on radius function properties, ***axis primitives***, based on simplified segment curvatures, and ***boundary primitives***, based on boundary surface curvatures. Since each primitive set is derived from different properties of the simplified segment and radius function, each captures different qualitative properties of the two-sided piece associated with the simplified segment. They can either be used separately or combined together to form cartesian-product primitive sets. I have also proposed a simple data structure, the labeled primitive adjacency graph, to be used to maintain information about the spatial relationships among primitives.

Since width primitives are defined by properties of radius function behavior, they reveal the behavior of the boundary surfaces associated with a simplified segment with respect to that simplified segment. Width primitives are themselves comprised of two components: slope districts and curvature districts. Using topological properties of scalar functions on surfaces, e.g. the radius function, I have proven that there are only three possible slope districts types.[1] There are six curvature district types.

Visualizing radius function behavior as if it were the height function of some mountainous terrain, each slope district corresponds to a mountain face together with the valley below it. At the bottom of the valley the associated boundary surfaces are "pinched" in, close to the simplified segment. As one climbs the mountain face, the associated boundary surfaces "bulge" out, each moving away symmetrically from the simplified segment until the mountaintop is reached. Curvature districts, on the other hand, further partition each slope district into regions that are locally either convex, concave, or saddle-like.

Using simplified segment Gaussian and mean curvatures, the simplified segment and its associated boundary surfaces also can be partitioned into a collec-

[1] This result does not depend on any special properties of the radius function.

tion of two-sided axis primitives, corresponding to regions of the simplified segment wherein the algebraic signs of the simplified segment Gaussian and mean curvatures are constant. Since simplified segment curvatures reflect the overall curvature trend of its associated two-sided piece, axis primitives, of which there are six, are two-sided pieces with constant overall curvature trend.

The final set of primitives, boundary primitives, are based on boundary surface curvatures. The simplified segment and associated boundary surfaces are partitioned into primitives each with the property that the algebraic signs of the Gaussian and mean curvatures are constant over each of the two boundary surfaces associated with the primitive. Each of the resulting 36 boundary primitives reflects the locally convex, concave, or saddle-like behavior of both associated boundary surfaces. Furthermore, the boundary primitives are related in a simple manner to properties of the simplified segment and radius function curvatures.

In Chapters 3 and 4, I have generalized much of Blum's two-dimensional shape description methodology to three dimensions. In Chapter 5, I have considered a different question: how does one compute the three-dimensional symmetric axis transform? After reviewing several of the many algorithms for computing the symmetric axis transform of a two-dimensional figure, I concluded that Bookstein's algorithm was the only one that deals explicitly with outline and symmetric axis tangents and that maintains symmetric axis continuity. Since, in my view, any symmetric axis algorithm must have these characteristics if it is to be useful for shape description, I have described, in Chapter 5, a three-dimensional generalization of Bookstein's two-dimensional algorithm.

From a polyhedral approximation to a smooth underlying outline wherein each polyhedron face is tangent to the outline, my three-dimensional generalization of Bookstein's algorithm yields a polyhedral surface approximating the symmetric surface of the outline. Outline normals are approximated by pseudonormal pencils and symmetric surface neighborhoods by symmetric surface planar elements. The algorithm, which consists primarily of using these approximations to simulate the geometry of the symmetric axis transform in continuous space, first finds, by exhaustive search, an initial "seed" symmetric surface planar element. Then, using this "seed," the simplified segment extension procedure constructs, without further searching, the entire simplified segment containing the "seed". The extension procedure fails at end curves and past branch curves. Once a branch is detected, a new "seed" is found, again by exhaustive search, and the extension procedure is invoked to construct branch simplified segments. When all such extensions terminate at end curves, the algorithm terminates, yielding the desired approximation.

6.2. Future Work

The results set forth in this dissertation lay the foundation for the experimental work necessary to evaluate the utility of the symmetric axis transform as a three-dimensional shape description tool. At the close of each chapter I have suggested directions for further research germane to the subject of each chapter. In this section, I outline a research programme, probably of several years duration if carried to completion, designed to yield a better understanding of the strengths and weaknesses of the symmetric axis transform as a shape description tool.

No matter what the application, I believe that some early experience with symmetric surfaces of the kinds of figures one is likely to encounter is essential. If nothing else, such experience is likely to yield an intuitive "feel" for whether similar figures have similar symmetric surfaces, for whether perturbations introduced by noise cause difficulty, and for what seem to be the most important features of the symmetric axis transform. Therefore, an implementation of the algorithm described in Chapter 5 is needed.

Several problems other than the implementation issues raised in Chapter 5 will need to be resolved. Most pressing is the form of the available three-dimensional data. The algorithm expects to receive a polyhedral approximation of the figure, yet the most common sources of data are point samples and stacks of two-dimensional slices. Though algorithms exist for converting such data into polyhedral approximations (see, for example, [Schumaker76a] and [Fuchs77a]), the suitability of the resulting approximations as input to the symmetric surface algorithm has yet to be investigated.

Once a suitable implementation is available, many interesting possibilities arise. Consider, for example, studying organs isolated from computed tomography (CT) studies. Does the symmetric axis transform appear to have potential as a "feature generator" for distinguishing among different organs? How do organ descriptions derived from the primitive sets proposed in Chapter 4 vary across subjects? Are such descriptions correlated with disease states? Which primitive sets are most useful? These, and other questions cannot be adequately studied without active involvement of medical experts. However, early *ad hoc* experiments are useful if only to build intuition, indicate promise (or lack thereof), and foster curiosity.

Should these *ad hoc* experiments indicate that further investigation is warranted, three separate research directions immediately present themselves:

(1) If measures derived from the symmetric axis transform are to be useful as indicators of abnormal conditions or to study variations in organ shape, some statistical tests of significance seem essential.

(2) In the structural pattern recognition literature, there is an increasing interest in inexact matching of labeled graphs. It might be fruitful to investigate matching primitive adjacency graphs as described in Chapter 4

against prototype graphs.

(3) Generalized cylinders have been studied extensively as both a shape analysis tool and as a representation to provide *a priori* information in computer vision systems. As discussed in Chapter 1, the symmetric axis transform seems to have several benefits over generalized cylinders. Further investigation is needed.

I do not expect the three-dimensional symmetric axis transform and the techniques described in this dissertation to answer the dreams of all shape description practitioners, if there be such. I do hope, however, that the early results I have presented here encourage others to study and apply the symmetric axis transform.

References

Agin76a.
Agin, G.J. and T.O. Binford, "Computer Description of Curved Objects," *IEEE Trans. Computers* **25**(4), pp. 439-449 (April 1976).

Aho74a.
Aho, A.V., J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley Publishing Co., Reading, MA (1974).

Alexander71a.
Alexander, R.M., *Size and Shape*, The Institute of Biology's Studies in Biology no. 29, Arnold (Publishers) Ltd, London (1971).

Alt62a.
Alt, F.L., "Digital Pattern Recognition by Moments," *J. ACM* 11, pp. 240-258 (1962).

Apostol74a.
Apostol, T.M., *Mathematical Analysis*, 2nd ed., Addison-Wesley Publ. Co., Reading, MA (1974).

Artzy80a.
Artzy, E., G. Frieder, and G.T. Herman, "The Theory, Design, Implementation and Evaluation of a Three-Dimensional Surface Detection Algorithm," *Computer Graphics (SIGGRAPH '80)* **14**(3), pp. 2-9 (July 1980).

Ashkar78a.
Ashkar, G.P. and J.W. Modestino, "The Contour Extraction Problem with Biomedical Applications," *Comp. Graphics and Im. Processing* **7**, pp. 331-355 (1978).

Attneave56a.
Attneave, F. and M.D. Arnoult, "The Quantitative Study of Shape and Pattern Perception," *Psychological Bulletin* **53**(6), pp. 452-471 (1956).

Badler79a.
Badler, N. and C. Dane, "The Medial Axis of a Coarse Binary Image Using Boundary Smoothing," *Proc. IEEE Comp. Soc. Conf. on Pattern Rec. and Image Proc.*, pp. 286-291 (Aug. 1979).

Bajcsy81a.
Bajcsy, R., C. Broit, P. Karp, and A. Stein, "Three-dimensional Computerized Anatomy Atlas of the Brain and the Matching of the Atlas to CAT and PETT Data," in *Proc. VII International Conference on Information Processing in Medical Imaging*, (June 1981).

Ballard78a.
Ballard, D.H., "Model-Directed Detection of Ribs in Chest Radiographs," TR11, Computer Science Department, Univ. of Rochester (March 1978).

Banchoff70a.

Banchoff, T.F., "Critical Points and Curvature for Embedded Polyhedral Surfaces," *Am. Mathematical Monthly* **77**, pp. 475-485 (May 1970).

Baumgart75a.

Baumgart, B.G., "A Polyhedron Representation for Computer Vision," pp. 589-596 in *Proc. NCC 1975*, AFIPS Press, Montvale, NJ (1975).

Bjorklund81a.

Bjorklund, C.M. and T. Pavlidis, "Global Shape Analysis by *k*-Syntactic Similarity," *IEEE Trans. PAMI* **3**(2), pp. 144-155 (March 1981).

Blum67a.

Blum, H., "A Transformation for Extracting New Descriptors of Shape," pp. 362-380 in *Models for the Perception of Speech and Visual Form*, ed. W. Wathen-Dunn, MIT Press, Cambridge, MA (1967).

Blum73a.

Blum, H., "Biological Shape and Visual Science (Part I)," *J. Theoretical Biology* **38**, pp. 205-287 (1973).

Blum74a.

Blum, H., "A Geometry for Biology," *Ann. N.Y. Acad. Sci.* **231**, pp. 19-30 (April 1974).

Blum79a.

Blum, H., *3-D Symmetric Axis Coordinates: An Overview and Prospectus*, Draft of a presentation given at the NSF Workshop on Representation of Three Dimensional Objects, U. Penn. May 1979.

Blum78a.

Blum, H. and R.N. Nagel, "Shape Description Using Weighted Symmetric Axis Features," *Pattern Recognition* **10**(3), pp. 167-180 (1978).

Bookstein78a.

Bookstein, F.L., *The Measurement of Biological Shape and Shape Change*, Lecture Notes in Biomathematics no. 24, Springer-Verlag, New York (1978).

Bookstein79a.

Bookstein, F.L., "The Line-Skeleton," *Comp. Graphics and Im. Proc.* **11**(2), pp. 123-137 (Oct. 1979).

Bosch78a.

Bosch, W., "A Procedure for Quantifying Certain Geomorphological Features," *Geographical Analysis* **X**(3), pp. 241-247 (July 1978).

Braun75a.

Braun, M., *Differential Equations and Their Applications: An Introduction to Applied Mathematics*, Applied Mathematical Sciences, Vol. 15, Springer-Verlag, New York (1975).

Brehm81a.

Brehm, U. and W. Kuhnel, "Smooth Approximation of Polyhedral Surfaces with Respect to Curvature Measures," pp. 64-68 in *Global Differential Geometry and Global Analysis Proceedings, 1979, (Lecture Notes in Mathematics no. 838)*, ed. D. Ferus, W. Kuhnel, U. Simon, and B. Wegner, (1981).

Broit81a.

Broit, C., *Optimal Registration of Deformed Images*, Ph.D. Dissertation, Dept. Computer and Information Science, U. Penn. (1981).

Brooks75a.

Brooks, R.A. and G. DiChiro, "Theory of Image Reconstruction in Computed Tomography," *Radiology* **117**, pp. 561-572 (Dec. 1975).

Brooks79a.

Brooks, R.A., R. Greiner, and T.O. Binford, "Progress Report on A Model-Based Vision System," pp. C1-C13 in *Proc. Workshop on the Representation of Three-Dimensional Objects*, ed. R.K. Bajcsy, (May 1979). U. Pennsylvania

Brown79a.

Brown, C.M., "Some Issues and Answers in Geometric Modelling," pp. F1-F35 in *Proc. Workshop on the Representation of Three-Dimensional Objects*, ed. R.K. Bajcsy, (May 1979). U. Pennsylvania

Brown67a.

Brown, D.R. and D.H. Owen, "The Metrics of Visual Form: Methodological Dyspepsia," *Psychological Bulletin* **68**(4), pp. 243-259 (1967).

Brown79b.

Brown, K.Q., "Geometric Transforms for Fast Geometric Algorithms," Ph.D. dissertation, Dept. Comp. Sci., Carnegie-Mellon Univ., Report CMU-CS-80-101 (Dec. 1979).

Calabi68a.

Calabi, L. and W.E. Hartnett, "Shape Recognition, Prairie Fires, Convex Deficiencies and Skeletons," *Am. Math. Monthly* **75**(4), pp. 335-342 (April 1968).

Castleman79a.

Castleman, K.R., *Digital Image Processing*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1979).

Catmull78a.

Catmull, E., "The Problems of Computer-Assisted Animation," *Computer Graphics (SIGGRAPH '78)* **12**(3), pp. 349-353 (Aug. 1978).

Cayley59a.

Cayley, A., "On Contour and Slope Lines," *The London, Edinburgh, and Dublin Philosophical Magazine and J. of Science* **18**(120), pp. 264-268 (Oct. 1859).

Clark73a.
Clark, W.A.V. and G.L. Gaile, "The Analysis and Recognition of Shapes," *Geografiska Annaler, Series B* **55**, pp. 153-163 (1973).

Courant53a.
Courant, R. and D. Hilbert, *Methods of Mathematical Physics, Vol. I*, Interscience Publishers, Inc., New York (1953).

DeSouza77a.
DeSouza, P.V. and P. Houghton, "Computer Location of Medial Axes," *Comp. and Biomedical Research* **10**(4), pp. 333-343 (Aug. 1977).

Duda73a.
Duda, R.O. and P.E. Hart, *Pattern Classification and Scence Analysis*, John Wiley & Sons, New York (1973).

Eastman77a.
Eastman, C., "The Concise Structuring of Geometric Data for CAD," in *Data Structures, Computer Graphics, and Pattern Recognition*, ed. A. Klinger, K.S. Fu, and T. Kunii, Academic Press, New York (1977).

Evans69a.
Evans, T.G., "Descriptive Pattern-Analysis Techniques: Potentialities and Problems," pp. 147-157 in *Methodologies of Pattern Recognition*, ed. S. Watanabe, Academic Press, New York (1969).

Feng75a.
Feng, H. and T. Pavlidis, "Decomposition of Polygons into Simpler Components: Feature Generation for Syntactic Pattern Recognition," *IEEE Trans. Comp.* **24**(6), pp. 636-650 (June 1975).

Fuchs77a.
Fuchs, H., Z.M. Kedem, and S.P. Uselton, "Optimal Surface Reconstruction from Planar Contours," *C. ACM* **20**(10), pp. 693-702 (Oct. 1977).

Gel'fand61a.
Gel'fand, I.M., *Lectures on Linear Algebra*, Interscience Publishers, New York (1961).

Granlund72a.
Granlund, G.H., "Fourier Preprocessing for Hand Print Character Recognition," *IEEE Trans. Comp.* **21**, pp. 195-201 (Feb. 1972).

Grunbaum67a.
Grunbaum, B., *Convex Polytopes*, Interscience Publishers (John Wiley & Sons), New York (1967).

Guillemin74a.
Guillemin, V. and A. Pollack, *Differential Topology*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1974).

Guttag78a.
Guttag, J.V., E. Horowitz, and D.R. Musser, "Abstract Data Types and Software Validation," *C. ACM* **21**(12), pp. 1048-1064 (Dec. 1978).

Haralick78a.
Haralick, R.M. and J. Kartus, "Arrangements, homomorphisms, and discrete relaxation," *IEEE Trans. SMC* **8**, pp. 600-612 (Aug. 1978).

Harary69a.
Harary, F., *Graph Theory*, Addison-Wesley Publ. Co., Reading, MA (1969).

Hilbert52a.
Hilbert, D. and S. Cohn-Vossen, *Geometry and the Imagination*, tr. P. Nemenyi, Chelsea Publishing Co., New York (1952).

Hu62a.
Hu, M.K., "Visual Pattern Recognition by Moment Invariants," *IRE Trans. Inf. Theory* **8**(2), pp. 179-187 (Feb. 1962).

Hursh76a.
Hursh, T.M., "The Study of Cranial Form: Measurement Techniques and Analytical Methods," pp. 465-491 in *The Measures of Man: Methodologies in Biological Anthropology*, ed. E. Giles and J.S. Friedlaender, Peabody Muesum Press, Cambridge, MA (1976).

Johnson78a.
Johnson, C.K., "Interactive Analysis of Critical Point Networks in Macromolecule Density Maps," *Acta Crystallographa, A* **34**, p. S353 (1978).

Kelly79a.
Kelly, P.J. and M.L. Weiss, *Geometry and Convexity: A Study in Mathematical Methods*, John Wiley & Sons, New York (1979).

Kirkpatrick79a.
Kirkpatrick, D.G., "Efficient Computation of Continuous Skeletons," *20th Annual Symposium on the Foundations of Computer Science (IEEE)*, pp. 18-27 (1979).

Kuhn70a.
Kuhn, T.S., *The Structure of Scientific Revolutions*, 2nd ed., The University of Chicago Press, Chicago (1970).

Kuratowski76a.
Kuratowski, K. and A. Mostowski, *Set Theory*, North-Holland Publishing Co., Amsterdam (1976).

Lee77a.
Lee, D.T., "An Alternate Method for Finding the Medial Axis of a Convex Polygon," in *Steps into Computational Geometry*, ed. F.P. Preparata, Coordinated Science Lab, U. Ill. Urbana (March 1977).

le Gros Clark45a.
le Gros Clark, W.E. and P.B. Medawar, *Essays on Growth and Form*, Clarendon Press, Oxford (1945).

Marr78a.
Marr, D. and H.K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," *Proc. Royal Soc. London B.* **200**(1140), pp. 269-294 (1978).

Massey67a.
Massey, W.S., *Algebraic Topology: An Introduction*, Harcourt, Brace & World, Inc., New York (1967).

Maxwell70a.
Maxwell, J.C., "On Hills and Dales," *The London, Edinburgh, and Dublin Philosophical Magazine and J. of Science, 4th Series* **40**(269), pp. 421-425 (Dec. 1870).

Meisel72a.
Meisel, W., *Computer-Oriented Approaches to Pattern Recognition*, Academic Press, New York (1972).

Millman77a.
Millman, R.S. and G.D. Parker, *Elements of Differential Geometry*, Prentice Hall, Englewood Cliffs, NJ (1977).

Montanari68a.
Montanari, U., "A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance," *J. ACM* **15**(4), pp. 600-624 (Oct. 1968).

Montanari69a.
Montanari, U., "Continuous Skeletons from Digitized Images," *J. ACM* **16**(4), pp. 534-549 (Oct. 1969).

Morse34a.
Morse, M. and G.B. Van Schaack, "The Critical Point Theory Under General Boundary Conditions," *Annals of Mathematics* **35**(3), pp. 545-571 (July 1934).

Nevatia77a.
Nevatia, R. and T.O. Binford, "Description and Recognition of Curved Objects," *Artifical Intelligence* **8**(1), pp. 77-98 (1977).

Newell79a.
Newell, M.E., "Geometric Representations in Computer Graphics: Part II," pp. K1-K12 in *Proc. Workshop on the Representation of Three-Dimensional Objects*, ed. R.K. Bajcsy, (May 1979). U. Pennsylvania

O'Rourke79a.

O'Rourke, J. and N. Badler, "Decomposition of Three-Dimensional Objects into Spheres," *IEEE Trans. PAMI* 1(3), pp. 295-305 (July 1979).

Osserman68a.

Osserman, R., *Two-Dimensional Calculus*, Harcourt, Brace & World, Inc., New York (1968).

Pavlidis68a.

Pavlidis, T., "Analysis of Set Patterns," *Pattern Recognition* 1, pp. 165-178 (1968).

Pavlidis72a.

Pavlidis, T., "Structural Pattern Recognition: Primitive and Juxtaposition Relations," pp. 421-451 in *Frontiers of Pattern Recognition*, ed. S. Watanabe, Academic Press, New York (1972).

Pavlidis77a.

Pavlidis, T., *Structural Pattern Recognition*, Springer-Verlag, New York (1977).

Pavlidis78a.

Pavlidis, T., "A Review of Algorithms for Shape Analysis," *Comp. Gr. and Im. Proc.* **7**(2), pp. 243-258 (April 1978).

Pavlidis80a.

Pavlidis, T., "Algorithms for Shape Analysis of Contours and Waveforms," *IEEE Trans. PAMI* **2**(4), pp. 301-312 (July 1980).

Persoon77a.

Persoon, E. and K.S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. SMC* **7**(3), pp. 170-179 (March 1977).

Pfaltz76a.

Pfaltz, J.L., "Surface Networks," *Geographical Analysis* **8**(1), pp. 77-93 (Jan. 1976).

Pfaltz78a.

Pfaltz, J.L., *Surface Networks, An Analytic Tool for the Study of Functional Surfaces*, Final Report on NSF Grant MCS-74-13353, Dept. Applied Math and Computer Science, Univ. of Virginia (July 1978).

Philbrick68a.

Philbrick, O., "Shape Description with the Medial Axis Transformation," pp. 395-407 in *Pictorial Pattern Recognition*, ed. G.C. Cheng, R.S. Ledley, D.K. Pollock, and A. Rosenfeld, Thompson Book Co., Washington, D.C. (1968).

Poston78a.

Poston, T. and I.N. Stewart, *Catastrophe Theory and its Applications*, Pitman Publishing Ltd., London (1978).

Preparata77a.

Preparata, F.P., "The Medial Axis of a Simple Polygon," pp. 443-450 in *Mathematical Foundations of Computer Science 1977*, ed. G. Goos and J. Hartmanis, Springer-Verlag, New York (1977).

Pullan78a.

Pullan, B.R., R.A. Fawcitt, and I. Isherwood, "Tissue Characterization by an Analysis of the Distribution of Attenuation Values in Computed Tomography Scans: A Preliminary Report," *J. Computer Assisted Tomography* **2**(1), pp. 49-54 (Jan. 1978).

Reingold77a.

Reingold, E.M., J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1977).

Requicha77a.

Requicha, A.A.G., "Mathematical Models of Rigid Solid Objects," TM-28, Production Automation Project, University of Rochester (Nov. 1977).

Requicha78a.

Requicha, A.A.G. and R.B. Tilove, "Mathematical Foundations of Constructive Solid Geometry: General Topology of Closed Regular Sets," TM-27, Production Automation Project, University of Rochester (March 1978).

Richards55a.

Richards, O.W., "D'Arcy W. Thompson's Mathematical Transformation and the Analysis of Growth," *Ann. N.Y. Acad. Sci.* **63**, pp. 456-473 (1955).

Rosenfeld76a.

Rosenfeld, A. and A.C. Kak, *Digital Picture Processing*, Academic Press, New York (1976).

Rosenfeld66a.

Rosenfeld, A. and J.L. Pfaltz, "Sequential Operations in Digital Picture Processing," *J. ACM* **13**(4), pp. 471-494 (Oct. 1966).

Sadjadi80a.

Sadjadi, F.A. and E.L. Hall, "Three-Dimensional Moment Invariants," *IEEE Trans. PAMI* **2**(2), pp. 127-136 (March 1980).

Saunders80a.

Saunders, P.T., *An Introduction to Catastrophe Theory*, Cambridge University Press, Cambridge (1980).

Schudy79a.

Schudy, R.B., *Spherical Harmonic Surface Models and Their Use for Extracting Moving Heart Surfaces from 3-D Cardiac Ultrasound Data*, Colloquium, Dept. Computer Science, Duke University March 27, 1979.

Schumaker76a.
Schumaker, L.L., "Fitting Surfaces to Scattered Data," pp. 203-268 in *Approximation Theory II*, ed. G. Lorentz, C. Chui, and L. Schumaker, Academic Press, New York (1976).

Searle70a.
Searle, N.H., "Shape Analysis by Use of Walsh Functions," in *Machine Intelligence 5*, ed. B. Meltzer and D. Michie, American Elsevier, New York (1970).

Sen76a.
Sen, A.K., "On a Class of Map Transformations," *Geographical Analysis* **8**(1), pp. 23-37 (Jan. 1976).

Shamos75a.
Shamos, M.I., "Geometric Complexity," *Proc. 7th ACM Symp. Theory Computing*, pp. 224-253 (May 1975).

Shamos78a.
Shamos, M.I., "Computational Geometry," Ph.D. dissertation, Dept. Comp. Sci., Yale Univ (May 1978).

Shamos76a.
Shamos, M.I. and D. Hoey, "Geometric Intersection Problems," *Proc. 17th Annual IEEE Symp. Foundations of Comp. Sci.*, pp. 208-215 (Oct. 1976).

Shani80a.
Shani, U., "A 3-D Model Driven System for the Recognition of Abdominal Anatomy from CT Scans," TR 77, Computer Science Department, Univ. of Rochester (May 1980).

Shapiro80a.
Shapiro, L.G., "A Structural Model of Shape," *IEEE Trans. PAMI* **2**(2), pp. 111-126 (Mar. 1980).

Shapiro79a.
Shapiro, L.G. and R.M. Haralick, "Decomposition of Two-Dimensional Shapes by Graph Theoretic Clustering," *IEEE Trans. PAMI* **1**(1), pp. 10-20 (Jan. 1979).

Shapiro81a.
Shapiro, L.G. and R.M. Haralick, "Structural Descriptions and Inexact Matching," *IEEE Trans. PAMI* **3**(5), pp. 504-519 (Sept. 1981).

Soroka79a.
Soroka, B.I., "Generalised Cylinders from Parallel Slices," *Proc. IEEE Comp. Soc. Conf. on Pattern Rec. and Image Proc.*, pp. 421-426 (Aug. 1979).

Soroka79b.

Soroka, B.I., "Generalised Cylinders and Serial Sections," pp. P1-P31 in *Proc. Workshop on the Representation of Three-Dimensional Objects*, ed. R.K. Bajcsy, (May 1979). U. Pennsylvania

Soroka78a.

Soroka, B.I. and R.K. Bajcsy, "A Program for Describing Complex Three-Dimensional Objects Using Generalized Cylinders as Primitives," *Proc. IEEE Comp. Soc. Conf. on Pattern Rec. and Image Proc.*, pp. 331-339 (1978).

Sprent72a.

Sprent, P., "The Mathematics of Size and Shape," *Biometrics* **28**(1), pp. 23-37 (March 1972).

Stoker69a.

Stoker, J.J., *Differential Geometry*, Wiley-Interscience, New York (1969).

Sunguroff78a.

Sunguroff, A. and D. Greenberg, "Computer Generated Images for Medical Applications," *Computer Graphics (SIGGRAPH '78)* **12**(3), pp. 196-202 (Aug. 1978).

Sutherland74a.

Sutherland, I.E. and G.W. Hodgman, "Reentrant Polygon Clipping," *C. ACM* **17**(1), pp. 32-42 (Jan. 1974).

Thomas60a.

Thomas, G. B., *Calculus and Analytic Geometry*, Third Edition, Addison-Wesley Publishing Co., Reading, MA (1960).

Thompson42a.

Thompson, D.W., *On Growth and Form*, Second edition, The University Press, Cambridge (1942).

Tobler78a.

Tobler, W.R., "Comparison of Plane Forms," *Geographical Analysis* **X**(2), pp. 154-162 (April 1978).

Tobler78b.

Tobler, W.R., "Comparing Figures by Regression," *Computer Graphics (SIGGRAPH '78)* **12**(3), pp. 193-195 (Aug. 1978).

Todd-Pokropek81a.

Todd-Pokropek, A., Personal communication. April 1981.

Turner-Smith80a.

Turner-Smith, A.R., *Shape Measurement and Scoliosis*, Oxford Orthopaedic Engineering Centre Report 7 1980.

Turner-Smith81a.
Turner-Smith, A.R., Personal communication. May 1981.

Warntz66a.
Warntz, W., "The Topology of a Socio-Economic Terrain and Spatial Flows," *The Regional Science Association Papers* **17**, pp. 47-61 (1966).

Webber79a.
Webber, R.L. and H. Blum, "Angular Invariants in Developing Human Mandibles," *Science* **206**, pp. 689-691 (Nov. 9 1979).

Widrow73a.
Widrow, B., "The "Rubber-Mask" Technique – I. Pattern Measurement and Analysis," *Pattern Recognition* **5**, pp. 175-197 (1973).

Williams82a.
Williams, T., *A Man-Machine Interface for Interpreting Electron Density Functions*, Ph.D. dissertation, in preparation, Dept. Computer Science, U. North Carolina at Chapel Hill (1982).

Zahn72a.
Zahn, C.T. and R.Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Comp.* **21**(3), pp. 269-281 (March 1972).