

**MEDIAL AXIS OF REGIONS BOUNDED BY B-SPLINE
CURVES AND SURFACES**

by

Suraj Ravi Musuvathy

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

School of Computing

The University of Utah

December 2011

Copyright © Suraj Ravi Musuvathy 2011

All Rights Reserved

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of Suraj Ravi Musuvathy
has been approved by the following supervisory committee members:

Elaine Cohen, Chair 9/9/2011
Date Approved

Richard Riesenfeld, Member 9/9/2011
Date Approved

Suresh Venkatasubramanian, Member 9/9/2011
Date Approved

Sarang Joshi, Member 9/9/2011
Date Approved

Thomas Fletcher, Member 9/9/2011
Date Approved

and by Alan Davis, Chair of
the Department of School of Computing

and by Charles A. Wight, Dean of The Graduate School.

ABSTRACT

The medial axis of an object is a shape descriptor that intuitively presents the morphology or structure of the object as well as intrinsic geometric properties of the object's shape. These properties have made the medial axis a vital ingredient for shape analysis applications, and therefore the computation of which is a fundamental problem in computational geometry. This dissertation presents new methods for accurately computing the 2D medial axis of planar objects bounded by B-spline curves, and the 3D medial axis of objects bounded by B-spline surfaces. The proposed methods for the 3D case are the first techniques that automatically compute the complete medial axis along with its topological structure directly from smooth boundary representations.

Our approach is based on the eikonal (grassfire) flow where the boundary is offset along the inward normal direction. As the boundary deforms, different regions start intersecting with each other to create the medial axis. In the generic situation, the (self-) intersection set is born at certain creation-type transition points, then grows and undergoes intermediate transitions at special isolated points, and finally ends at annihilation-type transition points. The intersection set evolves smoothly in between transition points. Our approach first computes and classifies all types of transition points. The medial axis is then computed as a time trace of the evolving intersection set of the boundary using theoretically derived evolution vector fields. This dynamic approach enables accurate tracking of elements of the medial axis as they evolve and thus also enables computation of topological structure of the solution.

Accurate computation of geometry and topology of 3D medial axes enables a new graph-theoretic method for shape analysis of objects represented with B-spline surfaces. Structural components are computed via the cycle basis of the graph representing the 1-complex of a 3D medial axis. This enables medial axis based surface segmentation, and structure based surface region selection and modification. We also present a new

approach for structural analysis of 3D objects based on scalar functions defined on their surfaces. This approach is enabled by accurate computation of geometry and structure of 2D medial axes of level sets of the scalar functions.

Edge curves of the 3D medial axis correspond to a subset of *ridges* on the bounding surfaces. Ridges are extremal curves of principal curvatures on a surface indicating salient intrinsic features of its shape, and hence are of particular interest as tools for shape analysis. This dissertation presents a new algorithm for accurately extracting all ridges directly from B-spline surfaces. The proposed technique is also extended to accurately extract ridges from isosurfaces of volumetric data using smooth implicit B-spline representations. Accurate ridge curves enable new higher-order methods for surface analysis. We present a new definition of *salient regions* in order to capture geometrically significant surface regions in the neighborhood of ridges as well as to identify salient segments of ridges.

For Puja

CONTENTS

ABSTRACT	iii
LIST OF FIGURES	viii
LIST OF TABLES	xv
ACKNOWLEDGEMENTS	xvi
CHAPTERS	
1. INTRODUCTION	1
1.1 Problem Statement	1
1.2 Definition and Local Structure of Medial Axis	6
1.3 Properties of Medial Axis (2D and 3D)	9
1.4 Applications of Medial Axis	10
1.5 Ridges: Definition, Properties and Applications	10
1.6 Research Contributions	13
1.7 Dissertation Outline	14
2. RELATED WORK	16
2.1 2D Medial Axis	16
2.2 3D Medial Axis	18
2.3 Ridges	21
3. BACKGROUND	25
3.1 Differential Geometry	25
3.2 B-spline Representation for Curves and Surfaces	28
3.3 Nonlinear Multivariate B-spline Root Solving Techniques	30
4. 2D MEDIAL AXIS	34
4.1 Overview	34
4.2 Transition Points and Transition Events	36
4.3 Medial Curves	40
4.4 Results and Discussion	46
4.5 Summary	50

5. RIDGES	51
5.1 Overview	51
5.2 Computing Seed Points	54
5.3 Tracing	57
5.4 Results and Discussion	63
5.5 Summary	67
6. 3D MEDIAL AXIS	69
6.1 Overview	69
6.2 Transition Points and Transition Events	71
6.3 Curve Elements	84
6.4 Medial Surfaces	90
6.5 Results and Discussion	94
6.6 Multisurface Models	100
6.7 Summary	102
7. SHAPE ANALYSIS USING MEDIAL AXES AND RIDGES	104
7.1 Structural Analysis of 3D Medial Axes Using Graph Cycle Bases	105
7.2 Shape Analysis Using Scalar Functions on Surfaces	113
7.3 Ridges from Isosurfaces of Volumetric Data	117
7.4 Geometrically Significant Regions Associated with Ridges	128
8. CONCLUSIONS	135
REFERENCES	138

LIST OF FIGURES

1.1 2D medial axis of a planar region bounded by a closed curve. An example of a regular point (A_1^2) in purple, end point (A_3) in cyan, and junction point (A_1^3) in red is indicated along with their maximal circles.	3
1.2 A 3D object represented by B-splines shown in (a) and its medial axis shown in (b).	3
1.3 Ridges and crests of a femur B-spline surface model.	5
1.4 Medial axis point types. Surface S shown in grey, surface points in red, medial axis points in blue. In (a)-(d), medial axis surfaces are also shown in grey. Arrows point in the corresponding surface normal direction.	9
3.1 Patterns of principal curvature lines around umbilics. κ_1 curvature lines are indicated in pale blue and κ_2 curvature lines are in magenta.	28
3.2 Examples of B-spline geometry. (a) a B-spline curve. (b) a B-spline surface. Control points are shown in red.	29
3.3 Subdivision based approach for root subdomain isolation.	31
3.4 Expression tree for $f_i = a(u_1) + b(u_2)c(u_3, u_4) - d(u_n)$	32
4.1 Medial axis viewed as evolving self-intersections of the boundary under the eikonal (grassfire) flow. A self-intersection is created at the cyan colored point. The self-intersection point evolves with time to trace out a medial axis curve segment.	36
4.2 Transition events at (a) A_3 type critical point, (b) and (c) A_1^2 type critical point, (d) and (e) A_1^3 type transition point. Transition points are shown in cyan, green and red and the evolving A_1^2 curves are shown in black.	41
4.3 Normals n , \hat{n} of different regions of an offset curve γ , $\hat{\gamma}$ at time t . The evolution vector field ξ is tangent to the medial curve.	42
4.4 Medial axis in augmented parameter-time space $\mathbb{R}_{u,\hat{u},t}^3$. \mathcal{I} is the locus of self-intersection points of the offsets of the boundary with a tangent $T_{\mathcal{I}}$ defined away from transition points.	45
4.5 Medial axis of a region bounded by a B-spline curve.	47
4.6 Examples of 2D medial axis computed directly from parametric B-spline boundary curves ((a), (b)) and using a discrete Voronoi-based method [53]((c), (d)). In (a) and (b), A_3 end points are shown in cyan, A_1^3 junction points in red and A_1^2 critical points in green. Arrows on the A_1^2 curves indicate the shock graph structure.	48

4.7	Examples of 2D medial axis using a discrete Voronoi-based method [53]. (a) Boundary sampled with 50 points. (b) Boundary sampled with 100 points. The discrete medial axis converges to the accurate result as sampling density increases.	49
5.1	Tracing overview - advance and slide steps. Advance steps are shown in brown, projection operations are shown in black, slide steps are shown in green. Also shown are principal directions and ridges (dark blue).	53
5.2	Robust initial advance step size selection. r_i is a ridge point and r_j is the advance point. The graph of $\phi(w)$ between r_i and r_j ($w \in [0, 1]$) is shown as a thick curve. Broken curve segments indicate $\phi(w)$, $w < 0$, $w > 1$. δ_0 should be chosen such that $\phi(w)$ does not have any local extrema between r_i and r_j . a) indicates a correct step size selection. b) and c) indicate incorrect step size selections. In case b), the trace will get stuck in a local minimum of $\phi(w)$ and will not reach a ridge. In case c), the trace will converge to an adjacent ridge segment.	58
5.3	Advance step size (turning point aware). Advance steps are shown in brown, maximum curvature principal directions are shown in pale blue, ridge is shown in dark blue.	59
5.4	Sliding to a ridge. Advance step is shown in brown, slide steps are shown in green. a) First slide step is moving toward ridge but has not yet reached ridge, b) Second slide step has crossed ridge, c) Slide is recomputed with reduced step size, new slide point has not yet reached ridge, d) Slide has reached ridge after a few steps of b) and c).	60
5.5	Tracing around umbilics. Scout points are shown in green.	61
5.6	Ridges on a Bézier patch. κ_1 -ridges are in blue and κ_2 -ridges are in red. (a) Euclidean space. (b) Parametric space. Black cross-hairs indicate umbilics.	63
5.7	Results of ridge computation on a Femur bone model. a) B-spline representation of bone, b) Sampled κ_1 ridges (darker blue indicates ridge proximity), c) Sampled κ_2 ridges (brighter red indicates ridge proximity), d) Ridges overlaid on surface ($\kappa_1 \rightarrow$ blue, $\kappa_2 \rightarrow$ red), e) Traced κ_1 ridges (black) overlaid on b), f) Traced κ_2 ridges (black) overlaid on c).	64
5.8	Results of ridge computation on a terrain elevation data set. a) B-spline approximation of the elevation data, b) Sampled κ_1 ridges (darker blue indicates ridge proximity), c) Sampled κ_2 ridges (brighter red indicates ridge proximity), d) Ridges overlaid on surface ($\kappa_1 \rightarrow$ blue, $\kappa_2 \rightarrow$ red), e) Traced κ_1 ridges (black) overlaid on b), f) Traced κ_2 ridges (black) overlaid on c).	65

5.9	An example of a false positive in κ_1 -ridge samples (rectangle outline) and a false negative (ellipse outline) in κ_2 -ridge samples of the terrain elevation model. (a), (b) and (c) represent enlarged views of κ_1 -ridge samples, traced ridges and κ_2 -ridge samples of the same region of the terrain elevation model.	66
6.1	Critical point of type (a) A_1^2 , (b) A_1^3 , shown in blue. Different regions of the surface S are shown in gray and arrows point in the corresponding surface normal directions.	73
6.2	Creation events at (a) A_3 type critical point, (b) A_1^2 type critical point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in blue.	78
6.3	Annihilation events at (a) A_1^2 type critical point, (b) A_1^3 type transition point and (c) A_1^4 type transition point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.	79
6.4	Intermediate transition event at an A_3 type critical point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in blue.	79
6.5	Intermediate transition events at an A_1^2 type critical point. The evolving intersection curves meet and exchange branches. This can result in a merge event (shown in (a)) or a split event (shown in (b)). Transition points are shown in brown and the evolving A_1^2 surfaces are shown in blue.	80
6.6	Intermediate transition event at an A_1A_3 type critical point. (a) One evolving intersection curve splits at the A_1A_3 point to create two segments belonging to the same A_1^2 surface. One of the ends of another intersection switches from an A_3 curve onto an outward flowing A_1^3 curve. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.	80
6.7	Intermediate transition event at an A_1A_3 type critical point. One evolving intersection curve splits at the A_1A_3 point to create two segments belonging to the same A_1^2 surface. A new A_1^2 surface is created with one end of the intersection curve on an outward flowing A_3 curve and the other end on an outward flowing A_1^3 curve. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.	81
6.8	Intermediate transition event at an A_1A_3 type critical point. An evolving intersection curve with one end on an inward flowing A_3 curve and the other end on an inward flowing A_1^3 curve ends at the A_1A_3 point. Two other evolving intersection curve segments merge and the A_1^2 surface evolves smoothly beyond the A_1A_3 point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.	81
6.9	Intermediate transition event at an A_1^3 type critical point. Three evolving intersection curves meet, split and create two triples of intersection curves. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.	81

6.10	Intermediate transition event at an A_1^3 type critical point. Two evolving intersection curves meet and split. A third intersection curve on a new A_1^2 surface is created. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.	82
6.11	Intermediate transition event at an A_1^3 type critical point. Two evolving intersection curves annihilate. Another set of two intersection curves merge. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.	83
6.12	Intermediate transition event at an A_1^4 type transition point. Three evolving intersection curves annihilate. Each of three other intersection curves switch an end point from an inward flowing A_1^3 curve onto the outward flowing A_1^3 curve. Transition points are shown in brown and the A_1^2 surfaces are shown in various colors.	83
6.13	Intermediate transition event at an A_1^4 type transition point. One evolving intersection curve ends. Each of four other intersection curves switch an end point from an inward flowing A_1^3 curve onto the outward flowing A_1^3 curve. One new intersection curve with end points on the two outward flowing A_1^3 curves is created. Transition points are shown in brown and the A_1^2 surfaces are shown in various colors.	84
6.14	Tangent planes of offset surfaces σ_i with normals n_i , $i = 1, 2, 3$ at a point P . Pairwise intersections of offset surfaces are along Z_{12}, Z_{23}, Z_{13}	86
6.15	Tangent planes of offset surfaces σ_i with normals n_i , $i = 1, 2$, intersecting along $n_1 \times n_2$ at a point P . $W \in TS_{\sigma_2}$	91
6.16	Several stages of evolution of medial axis of a deformed ellipsoid. Transition points are shown in green, evolving intersection curves are shown in red.	94
6.17	Several stages of evolution of medial axis of a deformed ellipsoid with a fin. Transition points are shown in green, evolving intersection curves are shown in red.	95
6.18	Several stages of evolution of medial axis of an object with multiple interconnected medial surfaces. Transition points are shown in green, evolving intersection curves are shown in red.	96
6.19	Medial axis of region shown in Figure 6.18(a) with all entities labeled.	97
6.20	Medial axis computed using a discrete approach [37]	97
6.21	Two views of the medial axis of a region consisting of a single A_1^2 surface with an inessential fin type A_1^3 curve, where the A_1^2 surface turns into a fin onto itself [34].	98
6.22	Medial axis of a region where a subset of the A_1^3 curves form two loops connected at an A_1^4 point in the middle, and an A_1^4 point at either end of the loops. (a) Input object. (b) A_3 curves shown in blue, A_1^3 curves in yellow and transition points in green. (c) A_1^2 surfaces shown in dull violet.	99

6.23	Result of medial axis computation on a multisurface model. (a) A model consisting of six parametric B-spline surfaces and (b) its medial axis. Medial surfaces are shown in dull violet, A_1^3 curves in yellow, A_1^4 points as green spheres, sharp edge curves in dull violet and corners as dark blue spheres.	101
6.24	Several stages of evolution of medial surfaces of a multisurface model. A_1^4 points are shown in green, evolving intersection curves are shown in red. .	103
7.1	An illustration of the 1-complex of the medial axis from Figure 1.2 (b) and its MRG.	108
7.2	Cycle basis analysis of a B-spline model shown in Figure 6.19. (a) a sketch of the medial axis with MRG vertex labels. (b) basis cycles of the MRG. (c) cycle basis features of medial axis mapped onto surface to obtain segmentation (side view). (d) top view of the same result.	110
7.3	Structural component-based design. (a) The surface region corresponding to one basis cycle component is trimmed. (b) The corresponding surface region is shrunk using a curvature-based variable offset.	111
7.4	Cycle basis analysis of a hip bone model. (a) polygonal surface mesh of a hip bone. (b) medial axis of the surface mesh. (c) a sketch of the corresponding MRG. (d) basis cycles computed using the proposed approach shown in different colors. (e) salient features corresponding to basis cycles (features shown with color of the basis cycle). (f) medial basis features mapped on surface mesh to obtain segmentation.	112
7.5	Structural analysis using 2D medial axes. (a) Input polygonal mesh. (b) Scalar Morse function defined on surface. (c) Level sets of scalar function. (d) 2D medial axes of level sets. (e) Connected medial structure indicating structural features.	115
7.6	Swept medial structure (left) computed using our approach, and medial axis (right) computed using a discrete algorithm [91]. Different surface sheets in our results are shown in different colors. The scalar functions used for computing the swept medial structures are also visualized (left). .	116
7.7	Ridges and crests extracted from an isosurface corresponding to a skull from a CT scan. κ_1 ridges are in blue and κ_2 ridges are in red. Crests are indicated by thicker curves.	124
7.8	Ridges extracted from the level set of simulation data of the Silicium model using (a) the proposed approach; (b) method of [22] on isosurface mesh; (c) isocontouring method of [126] using Gaussian extremality. In (c) all ridges are shown in purple since the ridge type is unknown.	125

7.9	Ridges extracted from isosurfaces resulting from linear elasticity analysis on a cube represented as a trivariate B-spline under vertical loads. The loads in (a) and (b) vary slightly. The color map on part of the volume indicates the z displacement where blue indicates almost no displacement and red indicates larger displacement. The isosurfaces in both images correspond to z displacement value of $4e^{-5}$. The ridges and crests indicate regions where the interior of the volume with a specific z displacement changes sharply.	126
7.10	Ridges extracted using the proposed approach on an ellipsoid (algebraic function) represented with a single trivariate Bézier patch. Umbilic points are shown in green.	127
7.11	Ridges extracted using the proposed approach on algebraic functions represented with a single trivariate Bézier patch: (a) Tangle: $x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 11.8 = 0$. (b) Smooth dodecahedron: $x^6 + y^6 + z^6 + 20(x^4y^2 + y^4z^2 + z^4x^2) = 1$. Umbilic points are shown in green. In (b) thick blue and red curves correspond to crests, thin blue and red curves correspond to noncrest elliptic ridges, thin green and yellow correspond to hyperbolic ridges of κ_1 and κ_2 , respectively.	127
7.12	Major and secondary ridges of an object; a) side view, b) top view. Major ridges of κ_1 are shown in cyan and those of κ_2 are shown in magenta. Secondary ridges of κ_1 are shown in blue and those of κ_2 are shown in red.	130
7.13	Salience boundaries and salient regions associated with major ridges. For κ_1 , major ridges are shown in cyan, principal curve segments in salient regions are shown in green and inflection points are shown in dark blue. For κ_2 , major ridges are shown in magenta, principal curve segments in salient regions are shown in yellow and inflection points are shown in dark red. Also shown is a closeup of a region that indicates a degenerate inflection point.	131
7.14	Occurrence of multiple inflection points between a major and secondary ridge even without existence of additional ridges in between them. a) Salience boundary points and other inflection points on principal curves traced from a few major ridge points. b) Results superimposed on surface indicating sampling of κ_2 ridge function. For κ_1 , major ridges are shown in cyan, principal curve segments in salient regions are shown in green and inflection points are shown in dark blue. For κ_2 , major ridges are shown in magenta, principal curve segments in salient regions are shown in yellow and inflection points are shown in dark red.	132
7.15	Visualizing geometric differences using salient regions. a) front view and, b) back view of a slightly asymmetric object, c) and d) major ridges only, e) and f) all ridges, g) and h) salient regions.	133

7.16 Visualizing geometric differences using salient regions. a) the major ridge corresponding to the bump in the surface lies in the center on both asymmetric regions, other major ridges are slightly different across plane of symmetry, b) salient regions clearly indicate geometric differences in the asymmetric regions pointed by the arrows. Also shown is a close up view of a region that contains a point where the salience boundary is tangent to the principal curve. 134

LIST OF TABLES

1.1 Classification of Ridges	12
5.1 Seed Points	66
5.2 Computation Characteristics (on an Intel 2.4GHz Processor with 8GB Memory)	66
6.1 Computation Times (TP = Transition Point, C = Curves)	100
7.1 Entities and Their Labels	107
7.2 Isosurface Subdomains	120
7.3 Computation Time	123

ACKNOWLEDGEMENTS

I am very fortunate to have been advised by Elaine Cohen. Throughout my graduate student tenure, Elaine has been a great mentor and guide in helping me achieve my goals. I am extremely grateful to her for many things. To list a few, I thank her for introducing me to exciting new research areas in the field of computational geometry, for inspiring me to and patiently teaching me how to solve geometric problems with B-splines, for constantly encouraging and supporting my work, for teaching me how to write effective scholarly articles, and for ensuring that I was financially supported throughout.

I sincerely appreciate the support and encouragement of my PhD committee consisting of Richard Riesenfeld, Suresh Venkatasubramanian, Sarang Joshi and Thomas Fletcher. They have always been enthusiastic about my research and have provided useful guidance and advice. I would like to thank Rich for inspiring me to work with smooth B-spline representations of geometry, and for teaching me how to create effective oral presentations and slides; Suresh for introducing me to interesting ways of combining ideas from continuous computational geometry and discrete graph theory, and for ensuring that I was making satisfactory progress in his role as Director of Graduate Studies; Sarang and Tom for introducing me to interesting work in the area of anatomical shape analysis.

I would also like to extend my appreciation and gratitude to James Damon at University of North Carolina Chapel Hill, whose elegant mathematical ideas greatly influenced my work. Although he was not officially part of my PhD committee, his contribution in guiding me through this work has been invaluable. I am deeply thankful to him for helping me better appreciate the theory and in transforming theoretical results into computational algorithms. The project on medial axis computation was conceived when Joon-Kyung Seong was a postdoctoral scholar here. His decision to

move to Korea during the early stages of this project resulted in its transfer to my hands, which has blossomed into my dissertation. I feel fortunate to have been given the opportunity to work on the project, and I thank Elaine and Jim Damon for trusting me with this project.

This work was supported in part by the University of Utah Graduate Research Fellowship 2010-2011. I am also thankful to Tom McCollough and Glenn McMinn for allowing me to work part time at Delcam USA during part of my PhD program.

It was a great experience working with fellow graduate students and friends Tobias Martin and Mathias Schott; colleagues Joon-Kyung Seong, Guoning Chen and David Johnson; as well as several past members of the GDC research group, with whom I have collaborated on research projects, coauthored papers and have had many stimulating discussions.

I am grateful to Karen Feinauer and the rest of the front office staff for ensuring that the administrative issues were handled smoothly.

I thank my parents for supporting my ambitions and my brother, Sanjeev, for his curiosity and excitement about my research. Finally, I am most grateful to my wife, Puja, for keeping my sanity in check through the highs and lows of this PhD adventure. Without her support, I would not have enjoyed, not even likely survived graduate student life.

CHAPTER 1

INTRODUCTION

In the late 1960s, Harry Blum introduced a geometric construct called the *medial axis* [13]. Blum's motivation was to identify a shape descriptor that would enable computational shape analysis in a manner that emulates the perceptual processes in the human visual system. It was originally described for planar two-dimensional (2D) shapes in order to characterize families of biological objects [13]. The concept was later extended to the three-dimensional (3D) case.

The medial axis intuitively presents the morphological structure of an object as a whole and intrinsic geometric properties of an objects' shape such as local thickness (also called local feature size [4]). Further, geometric properties of the object boundary can be computed directly from geometric properties of the medial axis [33]. Therefore, it is a shape descriptor that has become a vital ingredient for a variety of computational shape analysis applications that involve recognizing, comparing, analyzing and synthesizing objects. Many advantages of using the medial axis for such applications are outlined in a recent book on the subject [118]. Psychophysical experiments have confirmed that the medial axis is a very effective construct for shape perception [118]. With mild license of interpretation, one might see a *stick figure* drawing of a young child as a naive quest to abstract the essence of shape. Whether fundamentally tenable or not, this observation suggests that the medial axis concept is both natural and effective.

1.1 Problem Statement

This dissertation addresses the computation of the interior medial axis of objects directly from parametric B-spline representations of their boundaries. In particular, we consider the following two situations when the object encloses a:

1. 2D planar region, and is represented as a closed parametric B-spline curve.
2. 3D region, and is represented as a set of parametric B-spline surfaces.

This dissertation presents new approaches for accurately computing the medial axis and its correct topological structure for objects of both cases listed above. The key concept underlying the proposed approaches for the 2D as well as the 3D case builds on Blum’s original definition of the medial axis in terms of the *grassfire flow*. First, we present the approach for the 2D case, then extend it to the 3D case. Because of the increased complexity in structure, the extension of the approach to the 3D case is not straightforward. Whereas the 2D medial axis consists of a set of curve segments, the 3D medial axis consists of surfaces and curves as well as special isolated points. Hence, the term *axis* is misleading for the 3D case. Nonetheless, we follow this nomenclature since it is widely accepted in the existing literature. In this research, we address the computation and analysis of the part of the medial representation that is interior to the object.

Figure 1.1 shows a planar object bounded by a B-spline curve and its medial axis computed using the approach presented in this dissertation. The medial axis consists of curve segments, one for each bump of the shape, that are connected at some of the segments’ end points to form a graph structure. We address accurate computation of all curves and end points of the segments, as well as the topology of the graph structure.

Figure 1.2 shows an object in 3D bounded by a B-spline surface and its medial axis computed using the approach presented in this dissertation. The object consists of a base on which a protrusion is attached. This structure is captured clearly in the medial representation by a medial surface for each region and a junction curve where they meet indicated in thick yellow. Medial surfaces are also bounded by edge curves indicated in thick blue, that can meet junction curves at fin points highlighted as purple spheres. The 3D medial axis may also contain 6-junction points where junction curves meet. We address accurate computation of all such entities in a manner that assumes correct topological structure.

Most existing techniques for computing the medial axis (2D or 3D) require discrete representations of the boundary curves or surfaces. Discrete approximations of smooth

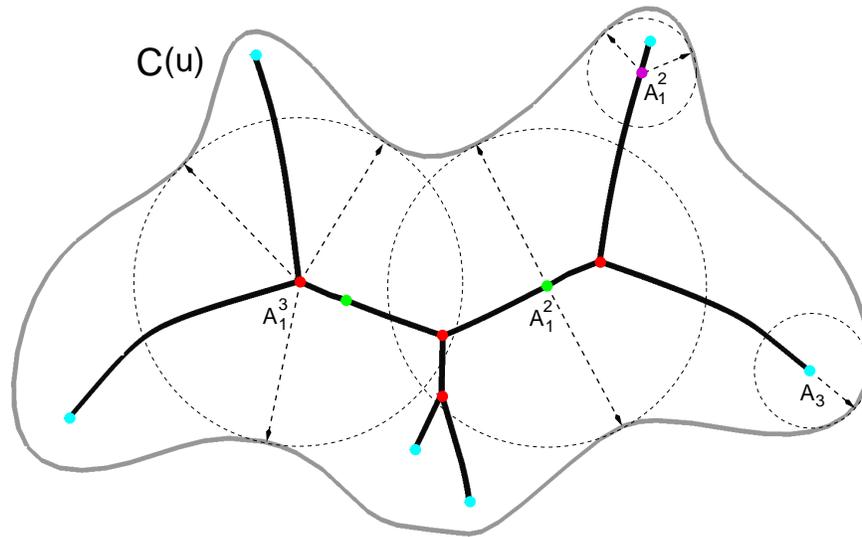
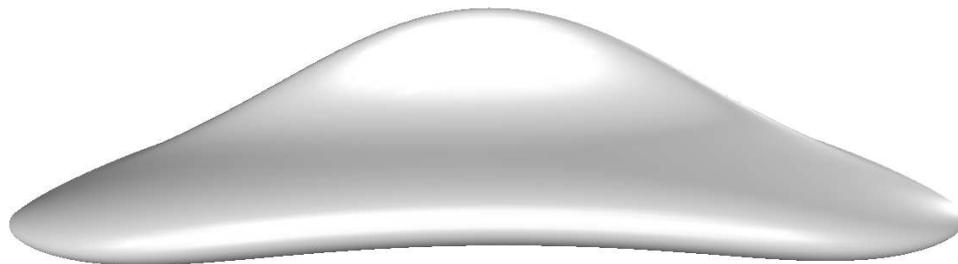
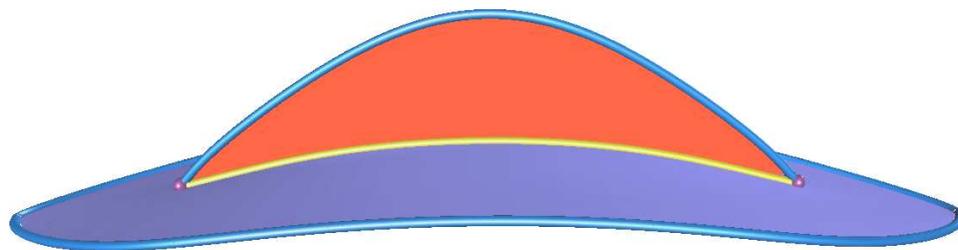


Figure 1.1. 2D medial axis of a planar region bounded by a closed curve. An example of a regular point (A_1^2) in purple, end point (A_3) in cyan, and junction point (A_1^3) in red is indicated along with their maximal circles.



(a)



(b)

Figure 1.2. A 3D object represented by B-splines shown in (a) and its medial axis shown in (b).

boundary representations introduce artifacts that are not part of the medial axis of the original smooth representations. Considerable effort, usually manual and hence time-consuming, is required to prune such artifacts. Several discrete techniques present approximations of the medial axis as a set of discrete elements (points in 2D or polygons in 3D) without topological structure. Considerable effort is required to infer this information in a process typically requiring extensive human interaction and is therefore tedious and time-consuming.

This dissertation presents the first techniques that automatically compute the complete interior medial axis along with its topological structure directly from smooth B-spline boundary representations. B-splines are a widely used form of smooth geometry representation and have many useful properties including variation diminishing and convex hull properties that support robust and efficient computation strategies [29].

1.1.1 Subproblem

Edge curves of the 3D medial axis correspond to a subset of *ridges* on an object’s surface linking sharp curvature related geometric features of the object’s shape to its structure. While the medial axis characterizes the form of an *object*, ridges indicate features on an object’s *surface*. Ridges have been used for several shape analysis applications, and are therefore addressed in detail in this dissertation. Most existing techniques for computing ridges require discrete representations of surfaces and, as a result provide limited accuracy, sometimes giving extraneous or spurious results. We present a new technique for accurately computing ridges directly from B-spline surfaces in a manner that manifests them as connected curve segments. Edge curves of the 3D medial axis are then derived from the extracted ridges. Using the proposed computational technique, Figure 1.3 shows ridges, and a special type of ridge called a *crest*, for a B-spline surface model of the upper part of a femur bone.

1.1.2 General Solution Framework

The proposed techniques for computing ridges and medial axes follow the general bottom-up framework of initially computing lower dimensional entities that are then used to compute higher dimensional entities, i.e., proceeding from points to curves to

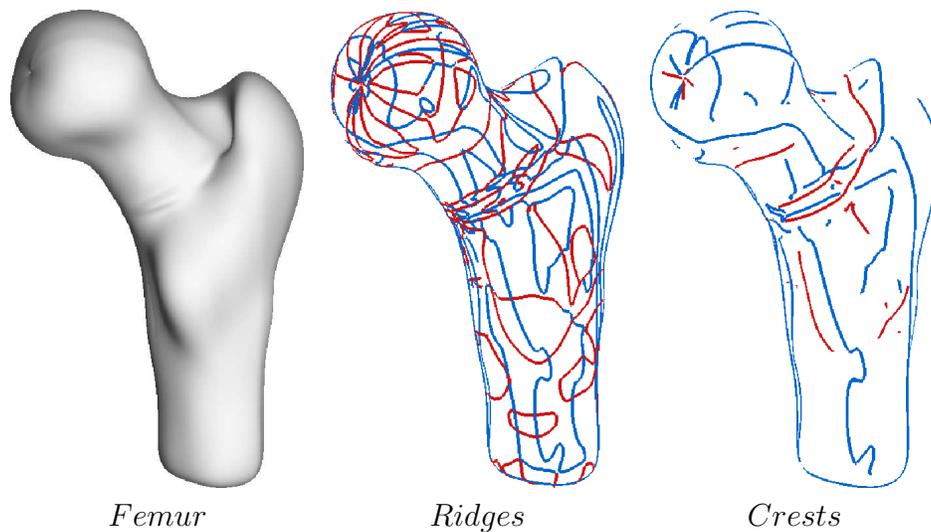


Figure 1.3. Ridges and crests of a femur B-spline surface model.

surfaces. Identifying all lower dimensional entities at which the topology of the solution transitions enables accurate computation of the topology of the entire solution.

The problems are further formulated in a dynamic setting that enables solutions to be computed as they are born, evolve and interact, and ultimately die. The general theoretical framework for the dynamic approach was developed by James Damon. We build on some of Damon's results in singularity theory, and subsequent work by Xianming Chen employing it to solve geometric problems that arise in a dynamic setting [26]. By treating the construction of the medial axis as a dynamic problem instead of a static one, this dissertation presents a new application of that theoretical framework.

For the part of the solution that is isolated points, the proposed research formulates them as roots of a system of geometric equations involving terms of the smooth B-spline representation of an object's boundary and its associated differential properties. These are typically nonlinear, high degree equations. With recent advances in multivariate nonlinear B-spline constraint solving techniques, addressing such problems has become more computationally tractable.

1.1.3 Additional Information from Accurate Solutions

Ridges and medial representations often exhibit complicated behaviors. Although the local structure of ridges and medial representations is well documented in the existing literature, existing approaches for computing ridges and medial representations provide limited information about local structure. Equipped with topologically accurate solutions, this dissertation also presents new techniques to analyze the structure of the solutions and identify suitable subsets. A complete list of contributions of this dissertation is presented in Section 1.6.

1.1.4 The Genericity Assumption

Generic properties of ridges and medial representations are well documented in the existing literature (See Sections 1.2 and 1.5.1). This research addresses such situations for which generic properties hold. The term *generic* includes “almost all” situations in the precise mathematical sense [33]. Assuming generic properties allows us to design algorithms that compute the topology of the solution with efficient algorithms. Later, we present special case examples where the genericity assumption is relaxed.

1.2 Definition and Local Structure of Medial Axis

In presenting the definitions of the medial axis for the 2D and 3D case separately, we also include a summary of the complete list of all generic local forms of the medial axis, as presented in [50] for the 2D case, and in [51] for the 3D case.

1.2.1 2D Medial Axis

Definition 1 *The medial axis of a planar region enclosed by a bounding curve C is the closure of the locus of centers of maximally inscribed circles that are tangent to at least 2 points on C .*

A complete characterization of the local structure of the 2D medial axis for generic regions is presented in [50]. Following the notation given in [50], a point of tangency of a maximally inscribed circle is of A_k type if the circle has order k contact with the boundary curve. Only the values $k = 1$ or 3 occur in the generic situation for medial axis points. When $k = 1$, the circle is tangent to the curve at the contact points.

For $k = 3$, in addition to the tangency property, the radius of the circle is a radius of curvature of the corresponding contact point on the curve and the curvature function at the contact point on the curve has a maximum value. A medial axis point whose maximal circle has m contact points of A_1 type is denoted by A_1^m . For each medial axis point, our work maintains the set of parameter values corresponding to each contact point on the boundary and the distance to the boundary (radius of maximal circle) in addition to the coordinate values of the center of the circle.

The medial axis of a generic 2D object is a set of curve segments whose end points satisfy special properties [50]. The 2D medial axis consists of the following point types (illustrated in Figure 1.1):

1. A_1^2 *curves*. The locus of A_1^2 points is a medial curve consisting of points whose maximal circle is tangent to 2 points on the curve.
2. A_3 *end points*. The maximally inscribed circle is in contact with 1 point on the curve with A_3 type. An A_3 point indicates the start or end of an A_1^2 curve segment.
3. A_1^3 *junction points*. The maximal circle at an A_1^3 point is tangent to 3 points on the curve. An A_1^3 point occurs at the location where 3 A_1^2 curves meet.

1.2.2 3D Medial Axis

Definition 2 *The medial axis of a region in R^3 enclosed by a bounding surface S is the closure of the locus of centers of maximally inscribed spheres that are tangent to at least 2 points on S .*

A complete characterization of the local structure of the 3D medial axis for generic regions is presented in [51]. For a maximally inscribed sphere, a point of tangency on the boundary is of A_k type¹ if the sphere has k -th order contact with the boundary [51]. Only the values $k = 1$ or 3 occur in the generic case for medial axis points. When $k = 1$, the sphere is tangent to the surface at the contact point. When $k = 3$, in addition to

¹We use the A_k^m notation for both the 2D and 3D cases, and their interpretation will be clear from the context in which they are presented

the tangency property, the radius of the sphere is also the radius of principal curvature of the corresponding contact point on the surface and the contact point on the surface is a ridge point. The definition of a ridge point is presented in Section 1.5. A medial axis point with a maximal sphere with m points of tangency is denoted $A_{k_1}A_{k_2} \cdots A_{k_m}$, and the type for a sphere with m A_1 contact points will be abbreviated to A_1^m . For each medial axis point, our work maintains the set of parameter values corresponding to each contact point on the boundary and the distance to the boundary (radius of maximal sphere) in addition to the coordinate values of the center of the sphere.

The medial axis of a generic 3D object consists of the following surface, curve and point entities [51] (See Figure 1.4 for illustrations of each medial axis point types):

1. A_1^2 *surfaces*. The set of medial surfaces consists of the locus of all A_1^2 points each with a maximal sphere tangent to two points on the surface.
2. A_3 *edge curves*. The maximally inscribed sphere at an A_3 point exhibits contact with an elliptic ridge point on the surface (See Section 1.5). The locus of A_3 points is a set of curve segments or loops that partially or completely bound A_1^2 surfaces.
3. A_1^3 *junction curves*. The maximal sphere at an A_1^3 point is tangent to three points on the surface. The locus of A_1^3 points is a set of curve segments or loops that partially or completely bound A_1^2 surfaces. Three A_1^2 surfaces meet along an A_1^3 curve.
4. A_1A_3 *fin points*. At such a point, an A_3 curve segment meets an A_1^3 curve segment and it marks the beginning/end of each curve. The maximal sphere at an A_1A_3 point is in contact with two surface points - one with A_1 type and the other with A_3 type contact.
5. A_1^4 *6-junction points*. The maximal sphere at an A_1^4 point is tangent to four points on the surface. Such points occur when six A_1^2 surfaces meet. This can also be viewed as points where four A_1^3 curves meet.

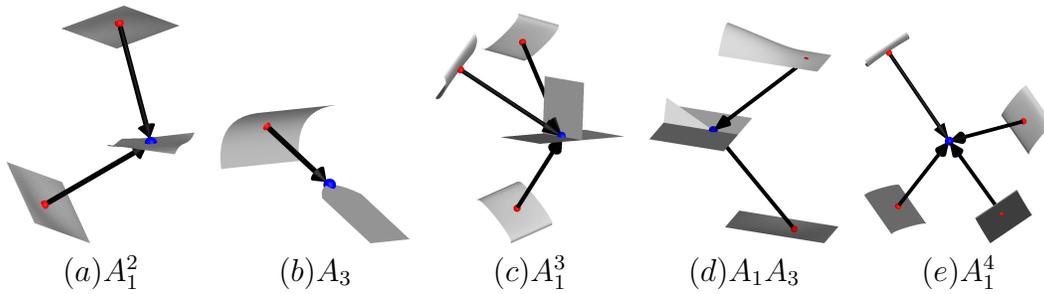


Figure 1.4. Medial axis point types. Surface S shown in grey, surface points in red, medial axis points in blue. In (a)-(d), medial axis surfaces are also shown in grey. Arrows point in the corresponding surface normal direction.

1.3 Properties of Medial Axis (2D and 3D)

The medial axis has a number of interesting properties, some of which are listed below. These properties hold for both the 2D and the 3D case.

1. The medial axis of an object in \mathbb{R}^n , $n = 2, 3$ generically has dimension \mathbb{R}^{n-1} . Hence, this is a lower dimensional representation of the shape of an object.
2. An object and its medial axis are homotopy equivalent [82, 116]. The medial axis of an object captures the connectivity structure of the object.
3. The medial axis of an object bounded by a G^1 boundary is a strong deformation retract of the boundary [116]. This implies that if the boundary is path connected, so is its medial axis.
4. The medial axis augmented with the radius of the corresponding maximally inscribed balls is called the *medial axis transform*. Given a medial axis and a valid radius field defined on it, the boundary of an object can be reconstructed [33, 51]. Further, the geometric properties of the boundary of an object can be determined from the geometric properties of its medial axis [33].
5. It is a well known fact that medial axes are sensitive to perturbations to the boundary of an object. It has been noted based on experimental observation that perturbations appear as fluctuating branches in the medial axis [5]. The full set of generic transitions of the medial axis under deformations of the boundary are characterized in [50, 52].

1.4 Applications of Medial Axis

Although medial axes computed using existing algorithms have limited accuracy and topological correctness, they have proven to be a vital ingredient in many application areas spanning diverse domains. A recent survey of application areas of medial axes “From the infinitely large to the infinitely small” object scales is presented in Chapter 11 of [118]. Approximate medial representations of anatomical objects have increased the accuracy of medical image analysis techniques [99] and enabled more accurate and meaningful statistical analyses of anatomical structures [71], thereby greatly enhancing research in medicine.² Medial axes have also been used for solid modeling [12], creating accurate volumetric models for engineering analysis such as stress calculations and aerodynamic simulations, searching databases of 3D objects for applications such as product design reuse and protein matching [119], surface segmentation [108], robot motion planning algorithms [48], and many other areas across science and engineering. Hierarchical multiscale methods requiring structural information of medial representations have vast potential for improving the efficiency of computationally demanding applications [71, 119].

1.5 Ridges: Definition, Properties and Applications

Ridge curves mark important intrinsic features of the shape of a surface. Consider a parametric surface $S(u, v) \in R^3$. Every point on $S(u, v)$, excluding umbilics, has two different principal curvatures ($\kappa_1 > \kappa_2$) and two corresponding principal directions (t_1, t_2) where t_1 and t_2 are 2D vectors, with the two elements of each vector denoting coefficients of S_u and S_v , respectively, at $S(u, v)$, for the principal directions. Ridges are defined in [59, 102] as follows:

Definition 3 *Ridges are loci of points on a surface where one of the principal curvatures attains a critical value (i.e., local maximum, minimum or inflection) along its corresponding principal direction.*

²also see http://midag.cs.unc.edu/MIDAG_FS.html

$$\phi_i(u, v) = \langle \nabla \kappa_i, t_i \rangle = 0, \quad i = 1 \text{ or } 2 \quad (1.1)$$

Other definitions of ridge-like structures in the existing literature include those of height ridges [40] and watershed ridges [90]. This dissertation addresses the extraction of principal curvature ridges following Definition 3.

Mathematical details of the derivation of Equation 1.1 are presented in [59]. In this research, we will henceforth refer to $\phi_i(u, v)$ as the *ridge function* and Equation 1.1 as the *ridge condition* for the corresponding principal curvature. Table 1.1 presents a classification of the various types of ridges. A ridge is called *elliptic* if κ_1 (κ_2) at a ridge point attains a local maximum (minimum) along its corresponding principal direction, and termed *hyperbolic* otherwise. Crests, valleys, and ravines are other frequently used terms in the existing literature to describe extremal curves of curvature. A *crest* is an elliptic ridge of the principal curvature with larger magnitude (See Table 1.1). The crest curve corresponding to the minimum principal curvature is typically called a *valley* or a *ravine*. Crests indicate perceptually salient ridges on a surface. It should be noted that some authors prefer to define ridges as the crest corresponding to the maximum principal curvature, while others refer to crests as κ_1 -ridges, where $|\kappa_1| \geq |\kappa_2|$. In this research, the term ridges encompasses crests, elliptic, and hyperbolic ridges.

1.5.1 Generic Properties of Ridges

Various aspects of the generic behavior of ridges on surfaces are summarized in this section (See [19, 20, 21, 76, 102] for discussions and proofs).

1. Two different ridges of the same principal curvature do not cross each other, except at umbilics. This property significantly reduces the complexity of ridge computation.
2. κ_1 -ridges may cross κ_2 -ridges at so called *purple* points.
3. Ridges of a particular principal curvature do not have start or end points on a surface (excluding the boundary of an open surface), except at umbilics.
4. Although principal directions are not defined at umbilics, ridges do occur at umbilics and exhibit complex behavior around umbilics.

Table 1.1. Classification of Ridges

Ridge Type	Definition
κ_1 -ridge	$\phi_1 \stackrel{def}{=} \langle \nabla \kappa_1, t_1 \rangle = 0$
κ_2 -ridge	$\phi_2 \stackrel{def}{=} \langle \nabla \kappa_2, t_2 \rangle = 0$
Elliptic ridge	$\phi_1 = 0, \quad t_1^T H_{\kappa_1} t_1 < 0$ $\phi_2 = 0, \quad t_2^T H_{\kappa_2} t_2 > 0$ $H_{\kappa_i} = \begin{bmatrix} \kappa_{iuv} & \kappa_{iuv} \\ \kappa_{iuv} & \kappa_{iuv} \end{bmatrix}, i = 1, 2$
Crest (κ_2 -crest: ravine or valley)	$\phi_1 = 0, \quad t_1^T H_{\kappa_1} t_1 < 0, \quad \kappa_1 > \kappa_2 $ $\phi_2 = 0, \quad t_2^T H_{\kappa_2} t_2 > 0, \quad \kappa_1 < \kappa_2 $

5. Elliptic ridges, and therefore crests, do not contain umbilics.
6. An umbilic may be classified as either a 1-ridge umbilic or a 3-ridge umbilic depending on the number of ridges arriving at the umbilic.
7. Ridges of a principal curvature intersect its corresponding principal direction transversally on the surface (R^3) except at a few isolated locations. This property enables tracing ridges on local coordinate systems formed by principal directions on a surface.
8. Locations on the surface where a ridge is tangential to the corresponding principal direction are called *turning* points (also known as A_4 points in geometry and singularity theory). Turning points are detected using this property in the approach presented in this research.
9. At a turning point, a ridge attains a local inflection in the corresponding principal direction and changes from being elliptic to hyperbolic or vice versa. This condition is given by $t_i^T H_{\kappa_i} t_i = 0, i = 1$ or 2 . Alternatively, Property 8 allows

identification of turning points without computing second order derivatives of curvatures.

10. A ridge of one principal curvature may be tangential to the other principal curvature direction on the surface (R^3), i.e., a κ_1 -ridge may be tangential to the minimum curvature direction and vice versa. A good example is the equator of an ellipsoid.

1.5.2 Applications of Ridges

The formal mathematical study of the role of ridges in geometry began with the research of Porteous [101] and was first emphasized for shape analysis by Koenderink [76]. Since then, ridges have proven valuable in a variety of applications spanning diverse domains. They are view independent curves and more stable with surface deformation compared to other feature curves such as curvature lines, which makes them very useful for shape matching [56, 73, 98, 122]. They are useful in visualization applications since they capture perceptually salient features of an object [30, 68, 84]. Other applications include freeform surface quality control [66] and geophysical analysis [83, 124].

1.6 Research Contributions

This dissertation presents the following contributions:

1. 2D Medial Axis

- (a) **Computation of accurate 2D medial axis from B-spline boundary curve.** A new approach is presented for automatically computing medial axes of two-dimensional objects accurately along with correct topological information and that does not generate nonmedial artifacts.
- (b) **Structural analysis of 3D objects using scalar functions defined on their surfaces.** We introduce a computational procedure to obtain structural information of 3D objects by computing 2D medial axes of level sets of scalar functions defined on the object surfaces.

2. 3D Medial Axis

- (a) **Computation of accurate 3D medial axis from B-spline boundary surface(s).** A new approach is presented for automatically computing medial axes of three-dimensional objects accurately along with correct topological information and that does not generate nonmedial artifacts.
- (b) **Shape analysis using structural features of 3D medial axis.** A new graph-theoretic technique for inferring, representing and analyzing structural components of medial axes is presented. The topological structure of the medial axis computed using the techniques presented in this work enables application of the graph based analysis technique directly for objects represented with B-splines.

3. Ridges

- (a) **Ridge extraction from B-spline surfaces.** A new algorithm is presented to accurately extract all generic ridges directly from B-spline surfaces. The smooth surface representation enables straightforward classification according to their subtypes.
- (b) **Ridge extraction from isosurfaces of volumetric data.** The algorithm is extended to compute ridges from volumetric data using implicit B-spline representations that enables accurate computation of ridges with improved quality.
- (c) **New method for identification of salient ridges and associated salient regions.** Current techniques only use information at ridge points. We present a new technique that uses geometric information in the neighborhood of ridges to not only identify salient subsets of ridges but also to identify geometrically salient regions of surfaces around ridges.

1.7 Dissertation Outline

This dissertation is organized as follows. Related work on the computation of medial axes in 2D and 3D, as well as ridges, is presented in Chapter 2. Relevant mathematical

background on differential geometry of curves and surfaces, the B-spline representation and its properties, as well as a summary of the existing approaches for subdivision based nonlinear geometric equation solving, is presented in Chapter 3. Chapter 4 presents the technique for computing 2D medial axes. Computation of ridge curves is addressed in Chapter 5. Chapter 6 presents the technique for computing 3D medial axes. New techniques for enriching computed medial axes and ridges for shape analysis are presented in Chapter 7. The contributions of this dissertation are summarized and concluded in Chapter 8.

CHAPTER 2

RELATED WORK

There has been tremendous interest over the last forty years in automatically computing medial axes of objects. This chapter presents a review of the techniques used for computing medial axes from smooth as well as discrete surface representations. The approaches can be broadly classified into Voronoi, distance field, eikonal and tracing based methods. We first present related work for the 2D medial axis in Section 2.1 and then for the 3D medial axis in Section 2.2. Many of the techniques presented for the discrete 2D case have analogous extensions to the 3D case, and we present references to both. This is by no means an exhaustive review, but enlists representative works in each area. Extensive surveys are presented in [118, 11].

Section 2.3 presents a review of existing techniques for computing ridges on surfaces. We cover methods that require discrete as well as smooth surface representations.

2.1 2D Medial Axis

2.1.1 Piecewise Smooth Representations

The tracing approach of [105] accurately computes medial axes of objects bounded by freeform curves. The method assumes that the object has at least one convex vertex (sharp corner point). A_1^2 curve segments of the medial axis are numerically traced from convex vertices. A_1^3 points are detected during tracing using distance and curvature checks and two new A_1^2 curves are spawned. Tracing ends either at a convex vertex or at an A_3 point which is detected using a curvature check during tracing. Since the method assumes that there is a convex edge on the curve, it is not suitable for computing medial axes bounded by planar curves without sharp corners.

An offset-based technique much like the eikonal flow for computing the medial axis of planar regions bounded by connected sets of line segments and circular arcs is presented

in [58, 57]. Transition points of the medial axis are determined using special geometric tests for combinations of line segments and circular arcs. Medial axis curve segments are then determined by computing offsets of the boundary contour. In their work, freeform spline curves are first approximated with line segments and circular arcs and the medial axis of the approximated boundary is then computed. There exist other techniques for computing approximate medial axes of freeform boundary curves that also use approximations of the boundary with simpler geometric entities such as line segments and circular arcs [2, 42, 62].

Several algorithms for computing the Voronoi diagram of regions with curve sites, which is closely related to the medial axis, have been presented [103, 28, 112, 60]. Differences between the Voronoi diagram and the medial axis are presented in [103]. An algorithm for computing the Voronoi diagram of planar NURBS curves based on computing bisector curves, critical points and trimming is presented in [112]. Error-bounded bisector curves are computed from the zero set of the distance function between curve pairs. Critical points including local and global intersection points of offset curves, and junction points of bisector curve segments are formulated as systems of geometric equations and computed using subdivision based constraint solving techniques. Bisector curve segments are then trimmed at critical points using distance checks to retain valid Voronoi segments. A similar approach for computing the Voronoi cell of a planar NURBS curve against a set of other curves is presented in [60]. An algorithm that incrementally adds Voronoi diagram segments from curve bisectors is presented in [103, 104]. Nonmedial segments are removed and missing medial segments are added in a postprocessing step in their approach.

2.1.2 Discrete Images

Thinning methods attempt to simulate the grassfire flow by an erosion like process of image pixels starting from the object boundary [77, 130]. Thinning methods are sensitive to Euclidean transformations of the image data and do not always guarantee single pixel width medial axes. Siddiqi et al. [117] simulate Blum's grassfire flow on discrete grids using partial differential equations to detect singularities of the flow based

on the average outward flux of the flow. Pixels corresponding to the medial axis are detected by combining the flux measurement with a homotopy preserving thinning process. Techniques based on discrete representation of distance fields on the image grid are presented in [128].

2.1.3 Point Sampled Boundary Curves

The Voronoi diagram of a set of point samples of a curve converges to the medial axis of the region as the sampling density increases [110, 17]. Several methods exploit this property by first computing Voronoi diagrams of the point samples and then identifying suitable subsets that belong to an approximation of the medial axis. Representative examples of such techniques include [94, 17]. Selection of pruning thresholds selecting subsets of Voronoi diagrams is an open problem [100].

2.1.4 Polygons

Since polygons represent a boundary as a collection of line segments, they form a subset of boundary representations consisting of line segments and circular arcs and hence, any of the approaches presented in [58, 2, 62] may be applied to obtain the medial axis.

2.2 3D Medial Axis

2.2.1 Piecewise Smooth Representations

The tracing approach of [106] seems to be the closest to obtaining accurate medial representations of objects bounded by piecewise smooth surfaces. The method assumes that the object has at least one convex vertex (corner point). A_1^3 junction curves (termed seams in their paper) are numerically traced from convex vertices. Six junction points are detected during seam tracing using distance to surface checks and three new junction curves are spawned. Seam tracing ends either at a convex vertex or at an A_3 point which is detected using a curvature distance check at every seam point. That paper presents ideas for computing interiors of medial sheets using bisectors of corresponding surfaces or numerical tracing; however, no results are presented. Since the method assumes that there is a convex edge on the surface that is part of bounding loop for a sheet, it

does not extend to medial representations with A_3 boundary curves that may end at fin points.

Bisectors of pairs of freeform surfaces are considered as building blocks of medial representations. Accurate techniques for computing bisectors of rational parametric surfaces are presented in [44]. However, there is no technique in the existing literature to identify all medial entities (such as fin points, six junction points) and their topology from the bisector surfaces. A similar technique based on bisectors of CSG objects is described in [64] but no results are presented.

A method for computing the 3D medial axis of extruded and revolved objects bounded by freeform surfaces is presented in [107]. The method computes the 2D medial axis of a planar profile face which is then transformed (extruded or revolved) to obtain a 3D medial axis.

2.2.2 Volumetric Images

A variety of techniques based on thinning of voxel data grids in an erosion like process starting from the object surface have been presented as surveyed in [77, 130]. Siddiqi et al. [117] simulate Blum's grassfire flow on discrete grids using partial differential equations to detect singularities of the flow based on the average outward flux of the flow. Medial skeleton voxels are detected by combining the flux measurement with a homotopy preserving thinning process. Methods that rely on the computation of distance fields are presented in [16, 7]. These techniques typically identify height ridges of the distance fields as medial voxels and perform thinning to obtain single voxel width approximations.

2.2.3 Point Clouds

Algorithms for computing medial representations from point sampled surfaces typically begin with computing Voronoi diagrams of the point sets and then identify medial skeletons as subsets of the Voronoi graphs. Amenta et al. [3, 4] identify a subset of Voronoi vertices called *poles* as medial skeleton points and compute a piecewise linear approximation of the medial skeleton using the connectivity of the poles in a weighted Voronoi diagram called the power diagram. Dey and Zhao [38, 39] use scale

and sampling density independent conditions to prune Voronoi diagrams. The λ -medial axis [24] selects a subset of the Voronoi diagram such that the maximal balls have radius of at least λ . Their result is a simplified approximation of the medial skeleton that preserves the homotopy type of the input object.

2.2.4 Polygonal Meshes

Approaches for computing medial representations of polygonal meshes can be classified into distance field methods [49], Voronoi methods [123] and tracing methods [31, 115]. Foskey et al. [49] analyze the gradient of the distance field to the surface to identify medial skeletons. A polygonal approximation that has the same homotopy type as the medial skeleton is computed as a subset of the Voronoi diagram in [123]. Methods based on tracing seam curves and computing intersections of seam curves are presented in [31, 115]. It should be noted that the seam curves are different from junction curves (loci of A_1^3 points) of generic medial skeletons. Seam curves arise due to the piecewise linear representation of the object surfaces.

2.2.5 Limitations of Existing Methods

Mathematical properties of the medial axis are well documented in the existing literature. However, to date state of the art techniques have been able to compute pieces of medial representations only for simplified approximations of objects. In addition, only approximate or simplified solutions using discrete techniques and partial solutions using higher order methods have been computed due to the complexity in structure and inherent nonlinearity of the medial axis. Existing discrete techniques are typically computationally fast. However, discrete techniques introduce artifacts that are not part of the medial axis and considerable effort is required to remove them. In addition, the output of existing techniques is typically a set of discrete elements without connectivity and structural information. Considerable manual effort is required in order to infer this information. This dissertation proposes a new higher order method, in conjunction with results from singularity theory, to automatically compute medial axes of regions bounded by B-splines accurately along with correct structural information and which does not generate nonmedial artifacts.

2.3 Ridges

This section presents a survey of existing methods for extracting ridges from various surface representations including smooth surfaces such as polynomials, B-splines and implicit functions, and discrete surface representations including isosurfaces of volumetric grid data and polygonal meshes. For discrete representations, estimating curvatures and their derivatives is a significant challenge. Smooth functions are typically used to estimate these quantities on the vertices of the tessellation. Approximate ridge points are identified on the edges and faces of the tessellation by linear interpolation of the ridge function estimates at the corresponding vertices. For C^3 smooth surface representations, curvatures and their derivatives can be computed exactly at any point. Ridges form continuous curves on smooth surfaces and extracting them accurately is more difficult.

2.3.1 Parametric Surfaces

Previous results on computing ridges of parametric surfaces can be classified into two categories: 1) lattice approach, and 2) sampling based methods. The research presented in Cazals et al. [19, 20, 21] falls into the first category. In [20], a system of equations that encodes all the ridges and umbilics of a parametric surface represented by a single polynomial is presented. This system is essentially the product of the ridge conditions for both the principal curvatures. An algorithm is presented to solve the resulting polynomial equations using algebraic techniques [19, 21], and to compute topologically correct ridges. Their research presents the first technique to compute the topology of ridges exactly at umbilics. Examples have been provided for single patch Bézier surfaces. The results computed using the technique presented in this research are validated with an example from their research.

Other prior results for computing ridges on parametric surfaces fall into the second category. A method based on sampling the ridge condition on the curvature lines of a parametric surface and reporting a collection of points (without connectivity information) that satisfy the ridge condition (within some error criteria) is presented in Hosaka [66]. The notion of *crest bands* has been introduced in Jefferies [69]. Crest bands are *soft ridges* that satisfy the ridge condition within a given threshold at uniform

samples in the parametric domain. This technique is also used for comparison and validation of the results presented in this research. A similar approach for sampling the ridge condition on a regular rectangular grid in the parametric space of a surface represented by thin-plate splines and connecting neighboring ridge points on the grid has been presented in Guéziec [56]. Approximation of ridges on the edges of a triangulation of the parametric domain has been presented in Kent et al. [73] and Morris [93]. In the research presented by Morris [93], umbilics are first detected and ridge points are identified on circles surrounding the umbilics.

Away from umbilics, classical tracing methods via solutions of ordinary differential equations (ODEs) representing the ridges may be employed. However, such methods require higher order surface smoothness and are computationally more demanding than the proposed method since derivatives of the ridge condition are required. In our experiments, much smaller step sizes were required by the ODE based method for achieving the same accuracy as the proposed algorithm. In addition, singular points of the ridge condition are required for robust tracing via ODEs and the task of locating such points is computationally demanding due to the complexity of the ridge condition and its derivatives. Due to all the above reasons, the proposed algorithm is computationally more suitable than ODE based methods for ridge tracing.

2.3.2 Implicits

A discrete method for computing intersections of an implicit function defining the surface and the ridge function is presented in [10]. An analytic solution for computing solutions of a system of equations describing ridges of a polynomial implicit function using a singularity theory approach is presented in [14]. The authors suggest using the implicit representation for discrete data but no results are presented. In our work, we adopt this idea but use a different approach to extract ridges using a different representation, piecewise polynomial implicit B-splines, that enables a global representation of large complicated discrete data sets.

2.3.3 Volumetric Data

The Marching Lines algorithm [127] presented a discrete technique to compute ridge curves on level sets of volumetric scalar fields such as medical images (MRI, CT scans). The technique computes intersection curves of an isosurface and the ridge function ϕ_i on the voxels of the data set. The *Gaussian extremality*, which is the product of the ridge functions ϕ_1 and ϕ_2 , was introduced in [126] and used to extract ridges from 3D images. The Gaussian extremality overcomes the problem of finding consistent principal direction orientations for evaluating the ridge functions. However, κ_1 and κ_2 ridges cannot be distinguished when computed as zeros of the Gaussian extremality. This causes additional errors in determining the topology of ridges around regions where a κ_1 ridge intersects a κ_2 ridge as noted in [22]. An image filtering approach is presented in [92] to first identify points on an isosurface and classify them as ridge points if they satisfied the ridge equation. Curvatures and their derivatives are estimated at required points using image filters in both techniques. In the work of [56], parametric B-splines are fit to isosurfaces and a discrete sampling technique is used to determine ridge curves on the isosurfaces.

2.3.4 Polygonal Meshes

Curvatures and their derivatives are estimated at mesh vertices by locally or globally fitting smooth surfaces using compactly supported radial basis functions [95], polynomials [129, 22, 121], MLS based implicit functions [74] or using discrete methods [63, 129]. Ridges are traced by detecting zero crossings of the ridge function on the vertices and edges of the meshes. Umbilics and ridges around them are detected in the method presented in [22]. All other approaches address only crests. Smoothing of the ridge function (as opposed to the surface) as well as smoothing crest space curves themselves was proposed in [63] to obtain crests with fewer undulations. Local angle measures between end points of crest segments have been used for connecting disjoint segments[129].

2.3.5 Limitations of Existing Approaches

Directly solving for the zero sets of the ridge condition is computationally expensive in terms of memory and processing time. In addition, to the best of our knowledge, there is no technique in the existing literature that can accurately compute the topology of ridges of NURBS surfaces at all locations from a disjoint set of ridge points. The authors of [19, 21] note that at the time of writing (2007), their technique for processing single polynomial surfaces was too slow to compute results in reasonable time for a bi-quintic Bézier patch. A domain tessellation or sampling-based method, albeit computationally fast, has other disadvantages. It is hard to obtain accurate ridges (exact zeros of the ridge condition) and connectivity information, especially at umbilics.

Existing approaches for extracting ridges from discrete data representations, including polygonal meshes and isosurfaces of volumetric data, tend to result in sets of disconnected ridge segments or tend to have undesirable undulations. There are several potential factors that contribute to this problem. First, most of the earlier techniques address only crests. However, crest curves on a smooth surface can turn into noncrest ridges that may, in turn, change back to crests. Extracting only crests therefore results in a disconnected subset of ridges. Second, curvatures and derivatives are typically not available with discrete data and hence are estimated. Being functions of the second and third order surface derivatives, respectively, this process is very sensitive to noise. Consequently, the quality of ridges extracted depends on the quality and consistency with which these quantities are estimated. In addition, fragmentation of the ridges can occur due to inconsistent choices of principal direction vector orientations [63] and when ridges are near parallel to mesh edges [129]. Our aim is to overcome the aforementioned problems with the discrete techniques and to extract all types of ridges for which generic conditions hold.

CHAPTER 3

BACKGROUND

This chapter presents a brief summary of results from the study of differential geometry of curves and surfaces, B-spline representation of curves of surfaces, and nonlinear multivariate B-spline root solving techniques that are used in this dissertation.

3.1 Differential Geometry

In this section, we review fundamental results on differential geometry that are utilized in this research. For details, the reader is referred to [29, 96, 97].

3.1.1 Curves

Consider a planar parametric curve $\gamma(u) : [u^1, u^2] \rightarrow R^2$, $\gamma \in C^2$. The tangent to the curve at any point is given by

$$T = \gamma_u \tag{3.1}$$

where subscripts indicate partial derivatives with respect to the parameter variable.

The curvature of the curve is given by

$$\kappa(u) = \frac{\|\gamma_u \times \gamma_{uu}\|}{\|\gamma_u\|^3} \tag{3.2}$$

The normal to the curve is given by

$$N = \frac{\gamma_u \times \gamma_{uu}}{\|\gamma_u \times \gamma_{uu}\|} - \frac{\gamma_u}{\|\gamma_u\|} \tag{3.3}$$

The curve normal, as defined by Equation 3.3, flips directions discontinuously at points where $\kappa = 0$. Suppose $\gamma(u)$ lies in the XY plane without loss of generality. Then $\gamma_u = [\gamma_u^{(x)} \ \gamma_u^{(y)} \ 0]^T$ is a vector in the XY plane. Then, Equation 3.4 represents consistently oriented normals and avoids flips. This representation is sometimes referred

to as the *manufacturing normal*, a term arising from its application in computer-aided manufacturing.

$$N = [0 \ 0 \ 1]^T \times \gamma_u \quad (3.4)$$

3.1.2 Surfaces

3.1.2.1 Principal Curvatures and Principal Directions

Consider a parametric surface $S(u, v) : [u^1, u^2] \times [v^1, v^2] \rightarrow R^3$, $S \in C^2$. The unit-length surface normal, $n(u, v) = \frac{S_u \times S_v}{\|S_u \times S_v\|}$ (assumed oriented inward for a closed surface, and $\|S_u \times S_v\| \neq 0$ since S is regular) where, subscripts indicate the partial derivatives with respect to the corresponding parameter variable. The matrix of the first fundamental form of the surface is given by,

$$I = \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \langle S_u, S_u \rangle & \langle S_u, S_v \rangle \\ \langle S_u, S_v \rangle & \langle S_v, S_v \rangle \end{bmatrix} \quad (3.5)$$

The matrix of the second fundamental form is given by,

$$II = \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} \langle S_{uu}, n \rangle & \langle S_{uv}, n \rangle \\ \langle S_{uv}, n \rangle & \langle S_{vv}, n \rangle \end{bmatrix} \quad (3.6)$$

Let A, B, C be defined as follows.

$$\begin{aligned} A &= EG - F^2 \\ B &= 2FM - GL - EN \\ C &= LN - M^2 \end{aligned} \quad (3.7)$$

Then, the principal curvatures at a point on the surface are given by,

$$\kappa_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad ; \quad \kappa_2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad (3.8)$$

$$\kappa_1 \geq \kappa_2$$

κ_1 is termed the maximum principal curvature and κ_2 is termed the minimum principal curvature. The corresponding principal curvature directions (nonunit magnitude) are given by,

$$\begin{aligned} t_1 &= \begin{bmatrix} t_1^1 \\ t_1^2 \end{bmatrix} = \begin{bmatrix} -(M - \kappa_1 F) \\ L - \kappa_1 E \end{bmatrix} \text{ or } \begin{bmatrix} -(N - \kappa_1 G) \\ M - \kappa_1 F \end{bmatrix} \\ t_2 &= \begin{bmatrix} t_2^1 \\ t_2^2 \end{bmatrix} = \begin{bmatrix} -(M - \kappa_2 F) \\ L - \kappa_2 E \end{bmatrix} \text{ or } \begin{bmatrix} -(N - \kappa_2 G) \\ M - \kappa_2 F \end{bmatrix} \end{aligned} \quad (3.9)$$

The coefficients in the above equation are chosen so that the principal direction vectors are nondegenerate and as well-conditioned as possible. If only one of the vectors is well-conditioned, the following property enables computation of the other vector.

3.1.2.2 Orthogonal Property of Principal Directions

The model or Euclidean space vectors, denoted by T_1 and T_2 are given by,

$$\begin{aligned} T_1 &= t_1^1 S_u + t_1^2 S_v \\ T_2 &= t_2^1 S_u + t_2^2 S_v \end{aligned} \tag{3.10}$$

T_1 and T_2 lie in the tangent plane at $S(u, v)$ spanned by S_u and S_v . If $\kappa_1 \neq \kappa_2$, then $\langle T_1, T_2 \rangle = 0$, so the two principal directions are orthogonal at all nonumbilic points on the surface. (see lemma 12.47 of [29] or [96]). This property enables tracing on a surface using local coordinate systems formed by T_1 and T_2 at nonumbilic points.

3.1.2.3 Three Types of Umbilics

Umbilics are points on the surface at which the normal curvatures in all directions are equal. Therefore, $\kappa_1 = \kappa_2$ and principal directions are not defined. However, lines of curvature exhibit three different patterns around generic umbilics denoted as the lemon, star, and (le)monstar patterns. Figure 3.1 shows the three different patterns formed by the principal curvature lines around umbilics. At a lemon umbilic, there is a single principal direction that changes from being a maximum curvature principal direction to a minimum curvature principal direction. At a star umbilic, there are three such principal directions. A monstar umbilic is similar to a star umbilic, except that all maximum (minimum) curvature directions are contained within a right angle.

3.1.2.4 Principal Directions Around Umbilics

Maekawa and Patrikalakis [86, 97] presented a technique to classify umbilics as lemon, star and monstar, and to compute exact principal direction patterns around umbilics. The principal directions were used in their work to trace curvature lines on a surface around umbilics. We employ this method to characterize behavior of ridges near umbilics. The idea is to represent the surface locally as a Monge patch in a reference frame centered at the umbilic and aligned with the tangent space of the surface at the umbilic. Then, the position vectors of the local maxima and minima of the Monge patch around the umbilic in the tangent space represent the maximum and minimum principal direction vectors. Details of this method can be found in [97]. Their approach

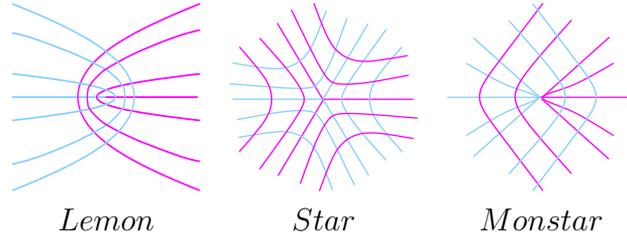


Figure 3.1. Patterns of principal curvature lines around umbilics. κ_1 curvature lines are indicated in pale blue and κ_2 curvature lines are in magenta.

also detects nongeneric umbilics. We have not found any literature characterizing the behavior of principal curvature lines and ridges around nongeneric umbilics and hence chose not to address them in this research.

At generic umbilics, there can either be a single maximum and minimum (lemon) or three maxima and minima (star and monstar). Maxima and minima occur on opposite sides of each other, i.e., the angle between their position vectors is π . This phenomenon is shown in Figure 3.1, where the relevant maximum and minimum principal directions are opposite to each other.

3.2 B-spline Representation for Curves and Surfaces

3.2.1 B-spline Curves

A B-spline curve (piecewise polynomial) [29] is given by,

$$\gamma(u) = \sum_{i=0}^n R_i B_{i,d}(u) \quad (3.11)$$

R_i are the control vertices of $\gamma(u)$. $B_{i,d}(u)$ are B-spline basis functions of degree d . Let $\Gamma = \{\tau_p\}_{p=0}^{(n+d+1)}$ be the corresponding knot vector. The support of $B_{i,d}(u) = [\tau_i, \tau_{i+d+1})$. It should be noted that the upper end of each interval is open. $\gamma(u)$ is defined over the interval $[\tau_d, \tau_{n+1})$. Figure 3.2 (a) shows a B-spline curve and its control points.

B-spline basis functions are piecewise polynomial, i.e., they are polynomial functions within each knot interval. The smoothness at each knot is determined by its multiplicity. Let π be the set of unique knots (breakpoints) and μ be the multiplicities of each breakpoint. Then, at each breakpoint π_p , $\gamma(u) \in C^{(d-\mu_p)}$. The derivatives of $\gamma(u)$ are

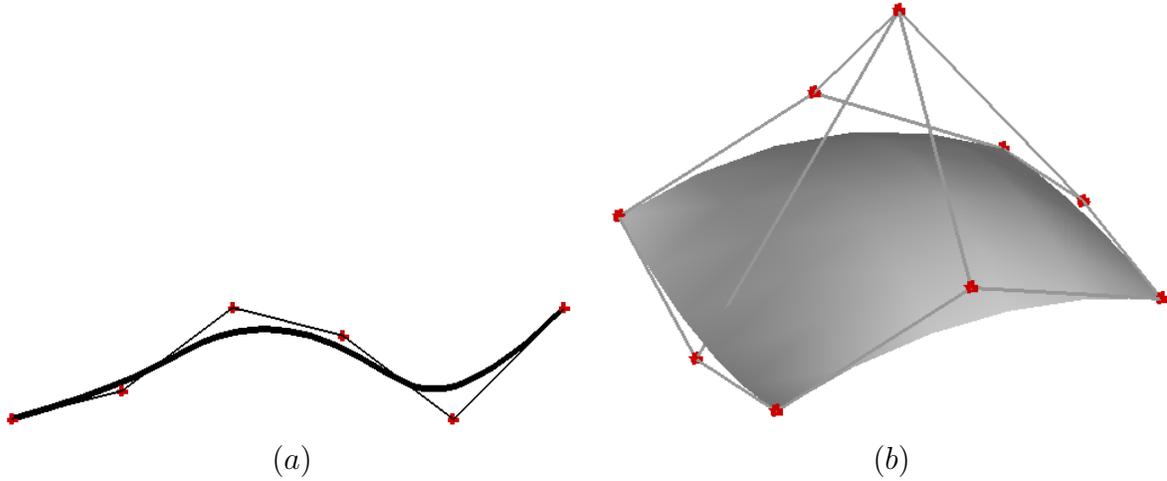


Figure 3.2. Examples of B-spline geometry. (a) a B-spline curve. (b) a B-spline surface. Control points are shown in red.

right continuous at a knot but $\gamma(u)$ fails to be C^l when the l^{th} derivatives are not left continuous at the knot.

3.2.2 Tensor Product B-spline Surfaces

A tensor product B-spline surface [29] is given by,

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n R_{ij} B_{i,d(u)}(u) N_{j,d(v)}(v) \quad (3.12)$$

R_{ij} are the control vertices of $S(u, v)$. $B_{i,d(u)}(u)$ and $N_{j,d(v)}(v)$ are B-spline basis functions of degrees $d(u)$ and $d(v)$, respectively. Let $\Gamma^{(u)} = \{\tau_p^{(u)}\}_{p=0}^{(m+d(u)+1)}$ and $\gamma^{(v)} = \{\tau_q^{(v)}\}_{q=0}^{(n+d(v)+1)}$ be the knot vectors in the u and v parametric directions, respectively. The support of $B_{i,d(u)}(u) = [\tau_i^{(u)}, \tau_{i+d(u)+1}^{(u)})$ and the support of $N_{j,d(v)}(v) = [\tau_j^{(v)}, \tau_{j+d(v)+1}^{(v)})$. It should be noted that the upper end of each interval is open. $S(u, v)$ is defined over the interval $[\tau_{d(u)}^{(u)}, \tau_{m+1}^{(u)}) \times [\tau_{d(v)}^{(v)}, \tau_{n+1}^{(v)})$. Figure 3.2 (b) shows a B-spline surface and its net of control points.

Let $\pi^{(u)}$ and $\pi^{(v)}$ be the set of unique knots (breakpoints) and $\mu^{(u)}$ and $\mu^{(v)}$ be the multiplicities of each breakpoint in the u and v parametric direction, respectively. Then, at each breakpoint $\pi_p^{(u)}$, $S(u, v) \in C^{(d(u)-\mu_p^{(u)})}$ in the u direction and at each knot $\pi_q^{(v)}$, $S(u, v) \in C^{(d(v)-\mu_q^{(v)})}$ in the v direction. The derivatives of $S(u, v)$ are right continuous

at a knot but $S(u, v)$ fails to be C^l when the l^{th} derivatives are not left continuous at the knot.

3.2.3 Properties of B-spline Curve and Surfaces

B-spline curves and surfaces exhibit two important properties that enable robust and efficient computation strategies, especially for multivariate nonlinear equation solvers as presented in Section 3.3.

1. **Variation diminishing property.** Consider a planar B-spline curve $\gamma(u)$ and a polyline connecting all of its control points. A horizontal line will intersect $\gamma(u)$ no more often than it will intersect the control polyline. In other words, a B-spline curve cannot have more undulations than its control polyline. A similar notion for B-spline surfaces exists.
2. **Convex hull property.** A B-spline curve or surface is completely contained within the convex hull of its control points.

3.3 Nonlinear Multivariate B-spline Root Solving Techniques

At several points in this dissertation, simultaneous roots of a set of nonlinear equations are required. The equations are derived from geometric properties of B-spline curves and surfaces, and are hence represented as multivariate B-splines.

Consider a system of n equations in n unknowns as shown in Equation 3.13. Each equation $f_i = 0$ is an n -variate scalar valued B-spline function. The goal is to compute all simultaneous roots of this system of equations. Each root is an n -tuple of corresponding parameter values (u_1, u_2, \dots, u_n) . We will refer to n as the dimensionality of the system.

$$\begin{aligned}
 f_1(u_1, u_2 \dots u_n) &= 0 \\
 f_2(u_1, u_2 \dots u_n) &= 0 \\
 \dots & \\
 f_n(u_1, u_2 \dots u_n) &= 0
 \end{aligned}
 \tag{3.13}$$

From the variation diminishing and convex hull properties of B-spline functions, we can deduce that f_i does not have a root if all of its control points (that are scalars) are

either less than or greater than zero. This result enables a divide-and-conquer strategy to successively identify smaller subdomains of the multivariate functions f_i , $i = 1..n$ in the n -dimensional space where simultaneous solutions exist as presented in [45]. B-spline subdivision is used to split domains into successively smaller regions until only a single solution exists within each subdomain. Subdomains that do not contain a root are pruned. Figure 3.3 shows an example of the B-spline subdivision approach for isolating a subdomain that contains a single root. For each candidate subdomain that contains a single solution, a numerical scheme such as a multivariate Newton's method is used to converge to an accurate solution. The approach presented in [45] enables computation of *all* roots of a system of nonlinear equations.

An equation typically consists of a set of terms, with each term represented as a multivariate B-spline function. For example, suppose $f_i = a(u_1) + b(u_2)c(u_3, u_4) - d(u_n)$, where a , b , c and d are scalar valued B-spline functions. In [45], symbolic B-spline operations (addition, subtraction and multiplication) are used to represent f_i as a single

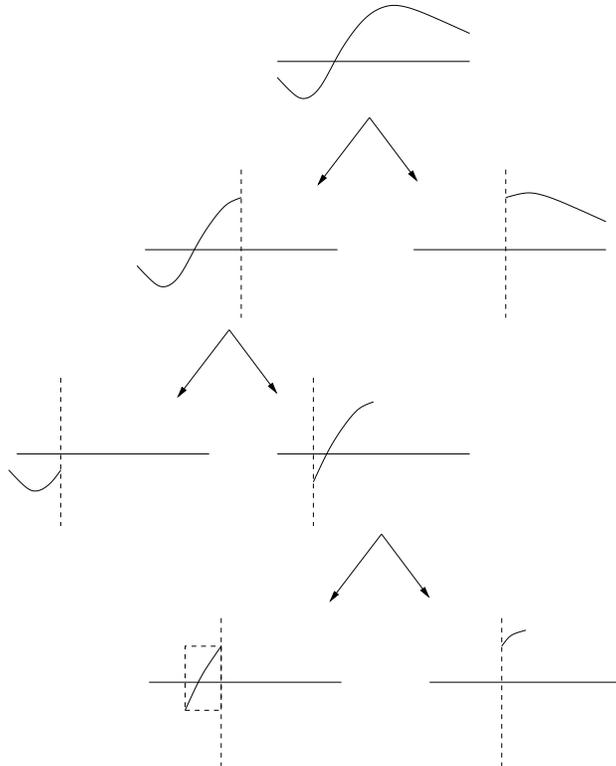


Figure 3.3. Subdivision based approach for root subdomain isolation.

B-spline function of u_1, u_2, \dots, u_n . A computational drawback of this approach is that the number of control points of a multivariate B-spline function grows exponentially with the number of dimensions. Therefore, computational complexity as well as space requirements to store the B-spline control points increase exponentially with increasing dimensionality. High degree equations also increase the computation time to perform B-spline operations. Further, B-spline subdivision increases the total number of control points over all subdomains. These drawbacks render this framework prohibitive for equations with higher dimensions and degree.

An expression tree approach was recently proposed [43] to overcome some of these limitations. This approach avoids symbolic operations on constituent terms of an equation and instead represents each equation as an expression tree, where the leaves are the terms and interior nodes are arithmetic operators (See Figure 3.4). Range values of each leaf are computed using B-spline properties, and interval arithmetic is used to aggregate range values to the root node of an expression tree. An equation contains a root if and only if the aggregated range values contain zero. Used in conjunction with the divide and conquer approach, where only leaf nodes are subdivided, this approach significantly improves computation time and reduces space requirements.

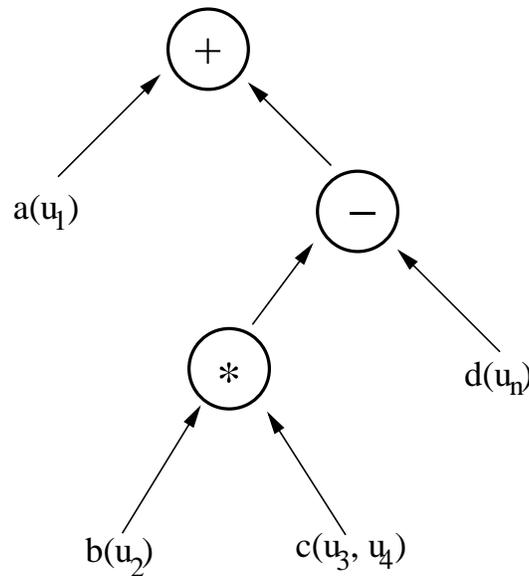


Figure 3.4. Expression tree for $f_i = a(u_1) + b(u_2)c(u_3, u_4) - d(u_n)$.

The expression tree approach, while being the current state of the art in the literature, still does not efficiently address high dimensional systems. We present specialized strategies to address such problems that arise in this work. (See Sections 6.2.4, 6.2.5 and 7.3.2 for details.)

CHAPTER 4

2D MEDIAL AXIS

This chapter presents a new technique to compute the interior medial axis of a planar region bounded by a closed parametric B-spline curve. The approach presented in this chapter for the 2D case introduces the techniques that will be generalized to enable computation of the 3D medial axis. Recall from Section 1.2.1 that the 2D medial axis consists of curves bounded by end points, or junction points where three medial curves meet. Precise mathematical definitions of each entity type are presented in the Section. Our approach computes all entities of the medial axis with arbitrary user specified accuracy along with correct topology. Parameter values of boundary curve points corresponding to every medial axis point and the respective distance is also computed, which gives the medial axis transform. Further, the method also indicates the direction of increasing distance along medial curves at all points, giving the shock structure [50]. The chapter is organized as follows. An overview of the proposed algorithm is presented in Section 4.1 with details in Sections 4.2 and 4.3. Results are presented in Section 4.4 and a summary of the chapter is presented in Section 4.5.

4.1 Overview

Our approach is based on Blum's original definition of the medial axis in terms of the *grassfire flow*, also called the *eikonal flow*. Derived from the physical analogy of a burning field of grass, the grassfire flow follows the evolution of a fire front started simultaneously all along the boundary of a field of grass. The fire front grows inward isotropically until it starts to meet itself. The fire gets extinguished at all such intersection points and the set of intersection points evolves smoothly until the fire is completely extinguished. The medial axis in 2D is exactly the set of all points where the fire fronts meet.

The B-spline representation enables computation of the medial axis as a time trace of the evolving (self-) intersection set of the boundary curve under the eikonal flow. The eikonal flow is given by offsets of the curve along the inward normal direction. Let \mathcal{B} be the boundary of a region in \mathbb{R}^2 represented by a parametric B-spline curve $C(u)$. The offset curve resulting from the eikonal flow at a time t is given by $\gamma(u, t) = C(u) + t n(u), t \geq 0$, where $n(u)$ is the inward directed unit curve normal. The variable t , which is exactly the offset distance, is referred to as time to emphasize the dynamic aspect of the algorithm. As the curve evolves under the eikonal flow in the increasing t direction, different regions start intersecting with each other. The self-intersection points trace out curves that grow until they reach certain end points. Figure 4.1 shows an illustration of this behavior. It should be noted that we are concerned only with the first intersection points of any two given points on the curve since only those intersection points belong to the medial axis.

The eikonal flow results in the creation of special transition points where A_1^2 curves are created or get annihilated. Away from transition points, the A_1^2 points evolve smoothly to trace out A_1^2 curves of the medial axis. The proposed approach presents techniques to accurately compute all types of transition points using the B-spline representation. A_1^2 curves are then computed by evolving intersection points over time using theoretically derived evolution vector fields. Algorithm 1 summarizes the steps involved in computing the medial axis.

Algorithm 1 Computing 2D medial axis

1. Compute transition points and classify as creation or annihilation.
 2. Trace medial axis curves from creation points until annihilation points are reached.
-

The eikonal flow method has been used to compute the medial axis of regions specified within discrete scalar grids [117] (pixellated images). That method classifies points on the discrete grid as whether or not they belong to the medial axis using computed properties of the eikonal flow at the grid points. Our method dynamically

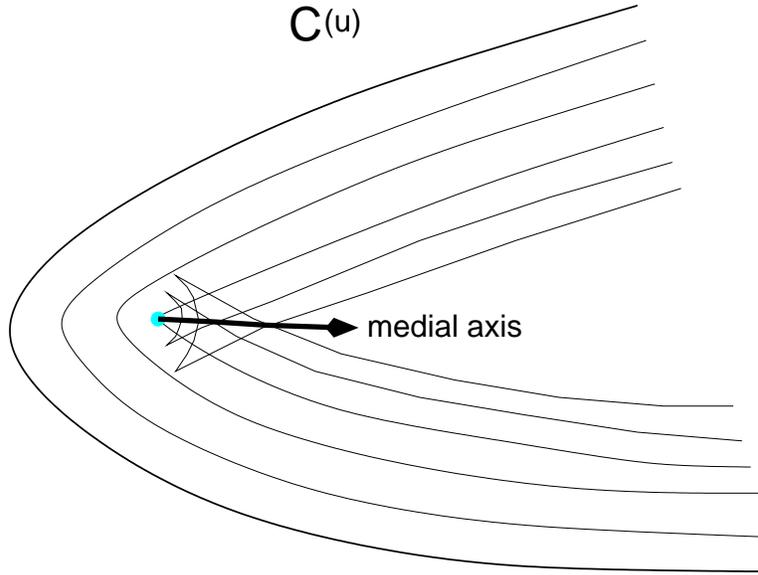


Figure 4.1. Medial axis viewed as evolving self-intersections of the boundary under the eikonal (grassfire) flow. A self-intersection is created at the cyan colored point. The self-intersection point evolves with time to trace out a medial axis curve segment.

emulates the eikonal flow using higher order methods to accurately compute the medial axis of regions bounded by B-spline curves.

4.2 Transition Points and Transition Events

This section lists and presents techniques to compute all types of transition points of the eikonal flow for computing the 2D medial axis. The transition points are those points where the distance to the boundary attains a critical value. Following are the types of all transition points of the eikonal flow for computing the 2D medial axis.

1. A_3 end points
2. A_1^2 critical points
3. A_1^3 junction points

See Figure 1.1 for an illustration of each type of transition point. An A_1^2 critical point is indicated in green. We present geometric equations for computing all three types of transition points. Roots of nonlinear geometric equations are computed us-

ing adaptations of robust subdivision based techniques [45, 43]. Extensions of these methods to improve efficiency of computing A_1^3 points are presented.

4.2.1 A_3 End Points

Denote the curvature of $C(u)$ by $\kappa(u)$. A_3 end points of the medial axis correspond to points on $C(u)$ corresponding to maximum values of $\kappa(u)$. These points are computed by first solving for the roots of Equation 4.1 that gives critical point of curvature. From the set of computed roots, those corresponding to maximum values are identified as points on $C(u)$ such that their centers of curvature correspond to A_3 end points of the medial axis.

$$\frac{\partial \kappa(u)}{\partial u} = 0 \quad (4.1)$$

4.2.2 A_1^2 Critical Points

Let $C_i(u_i)$, $i = 1, 2$ be two representations of the boundary curve to denote different regions.

$$\mathcal{D}(u_1, u_2) = \|C_1 - C_2\|^2 = \langle C_1 - C_2, C_1 - C_2 \rangle \quad (4.2)$$

Taking the partial derivatives of \mathcal{D} with respect to u_1, u_2 , we obtain

$$\begin{aligned} \langle C_1 - C_2, \frac{\partial C_1}{\partial u_1} \rangle &= 0 \\ \langle C_1 - C_2, \frac{\partial C_2}{\partial u_2} \rangle &= 0 \end{aligned} \quad (4.3)$$

Equation 4.3 implies that the curve normals at $C_1(u_1)$ and $C_2(u_2)$ must point exactly in opposite directions as illustrated by the green points in Figure 1.1. The parameter values corresponding to critical points of \mathcal{D} are obtained by computing solutions of Equation 4.3. Some solutions of Equation 4.3 may not correspond to maximal circles, so the maximality condition must be checked for all solutions using the procedure in Section 4.2.4. Trivial solutions where $u_1 = u_2$ are ignored. A_1^2 critical points on the medial axis are given by $\frac{C_1(u_1) + C_2(u_2)}{2}$.

4.2.3 A_1^3 Junction Points

Let $C_i(u_i)$, $i = 1, 2, 3$ be three representations of the boundary curve to denote different regions, and let N_i denote the unnormalized normals of C_i , respectively. For a point $P \in \mathbb{R}^2$ to be an A_1^3 point, the following equations must be satisfied.

$$\langle P - C_i, \frac{\partial C_i}{\partial u_i} \rangle = 0, \quad i = 1, 2, 3 \quad (4.4a)$$

$$\| P - C_1 \| = \| P - C_j \|, \quad j = 2, 3 \quad (4.4b)$$

Since $P = (x, y)$ is unknown, Equations 4.4(a)-(b) form a system of five equations in five unknowns. This system is reduced using the simplification techniques presented for the surface-surface bisector example in [45] as follows.

$$\begin{aligned} & \text{Let } P = C_1 + \alpha N_1 \\ & \text{substitute for P in } \| P - C_1 \| = \| P - C_2 \| \\ & \text{to obtain } \alpha = \frac{-\langle C_1 - C_2, C_1 - C_2 \rangle}{2\langle C_1 - C_2, N_1 \rangle} \end{aligned} \quad (4.5)$$

Denote $C_1 - C_i$ by C_{1mi} , $i = 2, 3$. Substituting for P and α in Equations 4.4(a)-(b) and simplifying yields a system of three equations in three variables.

$$2\langle C_{1m2}, N_1 \rangle \langle C_{1mi}, \frac{\partial C_i}{\partial u_i} \rangle - \| C_{1m2} \|^2 \langle N_1, \frac{\partial C_i}{\partial u_i} \rangle = 0, \quad i = 2, 3 \quad (4.6a)$$

$$\langle C_{1m2}, N_1 \rangle \| C_{1m3} \|^2 - \| C_{1m2} \|^2 \langle N_1, C_{1m3} \rangle = 0 \quad (4.6b)$$

Note that Equations 4.4(a) for $i = 1$ are automatically satisfied. It is possible to solve the system of equations in 4.6(a)-(b) by using subdivision based methods directly on the 3-variate functions [45, 43], but the computation time was fairly long. In our experiments, a quintic B-spline curve with 50 control points required about 20 minutes for computing A_1^3 points. We present an optimized approach below that reduced the computation time to less than a minute for the same input.

We have improved the computation time for this problem using hierarchical and parallel computation strategies. Equation 4.6(a) for $i = 2$ is dependent only upon u_1 and u_2 . Therefore, the problem size is reduced by first finding subdomains that satisfy these two equations, and then finding subdomains that also satisfy Equations 4.6(a)-(b)

for $i = 3$. Intuitively, Equation 4.6(a) for $i = 2$ requires two points on the curve that are equidistant from an offset point P in \mathbb{R}^2 . The remaining equations find a subset of those points for which there is a third curve point that is also equidistant to P .

The expression tree based approach is used to subdivide u_1, u_2 parametric domains until a user specified threshold is reached while using interval arithmetic to reject subdomains that do not satisfy Equation 4.6(a) with $i = 2$. Also, subdomains that do not correspond to regions that satisfy the maximal condition are pruned. The maximal condition is checked by computing ranges of $\rho = \alpha \| N_1 \|$ with interval arithmetic and testing whether 1) the range contains a part of the positive real line, and 2) is bounded by the size of the region enclosed within C . Negative values of ρ correspond to points outside the region since the offset would be in the outward normal direction and thus cannot contribute to the solution. Each of the remaining subdomains along with variables u_3 is then tested with Equations 4.6(a)-(b) using subdivision along u_3 parametric direction, interval arithmetic and pruning. Moreover, this step is performed in parallel since the subdomains are independent. Trivial solutions are ignored. The centers of remaining subdomains of u_i , $i = 1, 2, 3$ are used in a Newton-Raphson refinement step [45] to obtain accurate solutions and nonmaximal solutions are ignored. This step is also performed using parallel processing techniques.

4.2.4 Maximal Condition Check

In several steps of the algorithms presented in this chapter, it is necessary to check whether a given point P corresponds to the center of a circle with radius d is maximally inscribed within the region bounded by \mathcal{B} . In this work, the maximal condition is checked by computing the closest point to P on $C(u)$ and comparing the distance to the closest point with the value d . Closest points are computed by first computing points on the curve where the distance to P , given by $\| C(u) - P \|$, attains a critical value. Such points are obtained by finding roots of Equation 4.7

$$\langle C - P, C_u \rangle = 0 \tag{4.7}$$

All roots are then inspected to select the one with the lowest distance value. For P to correspond to a maximally inscribed circle, the smallest distance must equal d .

The roots are computed using robust and efficient subdivision based B-spline equation solving techniques presented in [45, 43].

4.2.5 Catalog of Transition Events

A complete list of all generic transitions of the 2D medial axis is presented in [50]. At transition points, corresponding evolution curves are either created, annihilated or undergo intermediate transitions. We present a summary of all generic transitions for the medial axis that are also illustrated in Figure 4.2.

1. *Creation Events:*

- i) At an A_3 point where κ has a local maximum. A new A_1^2 curve segment grows out from the A_3 point. (Figure 4.2 (a)).
- ii) At an A_1^2 critical point where the distance to C has a local minimum. Two new A_1^2 curve segments grow out from the critical point (Figure 4.2 (b)).

2. *Annihilation Events:*

- i) At an A_1^2 critical point where the distance to C has a local maximum. Two A_1^2 curve segments flow into the critical point and end there (Figure 4.2 (c)).
- ii) At an A_1^3 point where three A_1^2 curves flow inward and end (Figure 4.2 (d)). For this case, the A_1^3 point lies inside the triangle formed by the contact points on the boundary curve [50].

3. *Intermediate Transition Events:*

- i) At an A_1^3 point where two A_1^2 curves flow inward and a new A_1^2 curve flows outward (Figure 4.2 (e)). For this case, the A_1^3 point lies outside the triangle formed by the contact points on the boundary curve [50].

4.3 Medial Curves

This section presents evolution vector fields defined on the ambient space to trace A_1^2 curves resulting from intersections of the boundary deforming under the eikonal

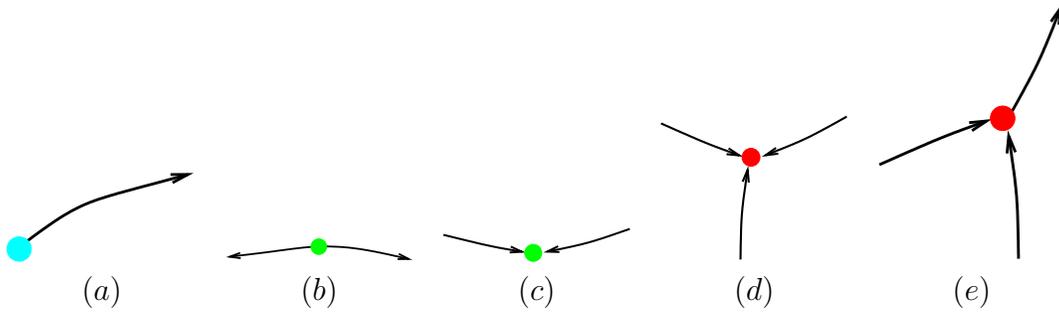


Figure 4.2. Transition events at (a) A_3 type critical point, (b) and (c) A_1^2 type critical point, (d) and (e) A_1^3 type transition point. Transition points are shown in cyan, green and red and the evolving A_1^2 curves are shown in black.

flow. Between transition points, the evolution vector fields are integrated to compute a time trace of the evolving A_1^2 curves that, together with the transition points, form the medial axis.

4.3.1 Evolution Vector Field

Consider two separate regions $C(u)$ and $\hat{C}(\hat{u})$ and denote their offsets under the eikonal flow by $\gamma(u, t)$ and $\hat{\gamma}(\hat{u}, t)$. Let n and \hat{n} denote the unit normal vectors of C and \hat{C} respectively at a point P on an intersection curve. If P is not a transition point, then γ and $\hat{\gamma}$ are not parallel and hence, n and \hat{n} are independent. Since n and \hat{n} are independent, $\{\gamma_u, \hat{\gamma}_{\hat{u}}\}$ are independent vector fields, and hence a basis for \mathbb{R}^2 in the neighborhood. Thus, in the neighborhood of P ,

$$\hat{n} - n = a\gamma_u - \hat{a}\hat{\gamma}_{\hat{u}} \quad (4.8)$$

From Equation 4.8, we define an evolution vector field, ξ , in the neighborhood of P in \mathbb{R}^2 given by two equivalent representations in Equation 4.9. (See Figure 4.3 for illustration).

$$\xi = n + a\gamma_u = \hat{n} + \hat{a}\hat{\gamma}_{\hat{u}} \quad (4.9)$$

Proposition 4.3.1 ξ is tangent to the intersection curve of γ and $\hat{\gamma}$ over all times t .

Proof. As γ deforms under the eikonal flow with varying t , there will be a unique curve point for a given time t in the neighborhood of any point Q that lies within

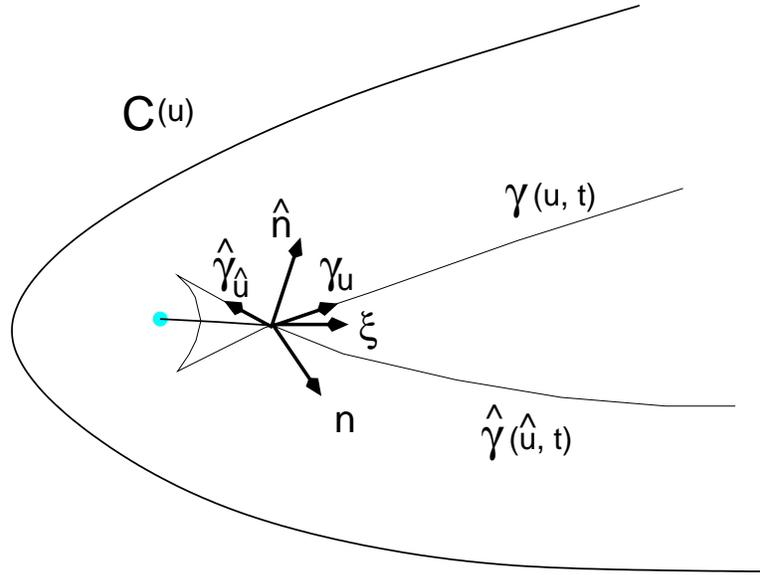


Figure 4.3. Normals n , \hat{n} of different regions of an offset curve γ , $\hat{\gamma}$ at time t . The evolution vector field ξ is tangent to the medial curve.

the region covered by the deforming curves. By the inverse function theorem, t is a differentiable function in the neighborhood of Q . Consider the values of t in the neighborhood of Q as a scalar field. Since each point on γ is deforming along the corresponding curve normal vector n , $\nabla t = n$. Therefore, the directional derivative of t along n , $\nabla_n t = \langle \nabla t, n \rangle = 1$. Further, the directional derivative of t along γ_u is zero since it is in the tangent direction at $\gamma(u)$. Therefore, the directional derivative of t along ξ is $\nabla_\xi t = \langle \nabla t, \xi \rangle = \langle \nabla t, n + a\gamma_u \rangle = 1$.

For the moment, consider separate time variables t and \hat{t} for the two different curves γ and $\hat{\gamma}$. Define $\phi(Q) = t - \hat{t}$, where t (\hat{t}) is the time when γ ($\hat{\gamma}$) reaches Q . The directional derivative of ϕ along ξ is $\nabla_\xi \phi = \nabla_\xi t - \nabla_\xi \hat{t} = 1 - 1 = 0$. Thus, ξ is tangent to the level curves of ϕ . Setting $t = \hat{t}$ gives $\phi = 0$, that correspond to intersection points of the offset curves that are on the medial axis. Hence, ξ is tangent to the A_1^2 curve at an A_1^2 point. ■

Thus, we can follow the evolution of the intersection curve (where the evolving curves remain transverse) by integrating the vector field ξ with initial conditions as the points on the intersection curve.

$$\frac{dx}{dt} = \xi(x), \quad x(0) = P, \quad x(t) \in \mathbb{R}^2 \quad (4.10)$$

Furthermore, define vector fields on the parameter-time space as $\zeta = e_t + ae_u$ and $\hat{\zeta} = e_t + \hat{a}e_{\hat{u}}$, where e_i denotes the unit vector in the parameter-time space direction i , $i = t, u, \hat{u}$. Then, $\xi = d\gamma(\zeta) = d\hat{\gamma}(\hat{\zeta})$. This implies that the integral curves of the ζ are mapped by γ to integral curves of ξ , and similarly for $\hat{\gamma}$. The corresponding integral curves of ζ and $\hat{\zeta}$ will trace the evolution of the intersections curves in the parameter space.

4.3.2 Tracing Algorithm for A_1^2 Curves

The transition points are first classified as source and sink points. A_3 points and creation type A_1^2 points are source points. Annihilation type A_1^2 points and annihilation type A_1^3 points are sink points. An intermediate type A_1^3 point corresponds to both a source and a sink point. Starting points are computed from each type of source point. For each starting point, two initial points corresponding to the tangency points of an A_1^2 point are computed. The parameter values of the two initial points are then refined to correspond to a point on the A_1^2 curve using a Newton's method for Equation 4.11.

$$C_1 + rn_1 = C_2 + rn_2 \quad (4.11)$$

where $r = r_{src} + \delta_{src}$ is the radius of the maximal circle at the starting A_1^2 point, r_{src} is the radius of the maximal circle at the source point and δ_{src} is a user-specified step size ($\delta_{src} = 10^{-3}$ was used in our examples). The projection also computes the parameter values of the two boundary curve points corresponding to the starting A_1^2 point.

Initial points are computed from the three types of source points as follows:

1. A_3 point. Let u be the parameter value at the A_3 point. Initial points are computed with parameter values $u + \delta$ and $u - \delta$ ($\delta = 10^{-3}$ in our experiments).
2. Creation type A_1^2 point. Let u_1 and u_2 be the parameter values at the contact points on C . Initial points are created at $u_1 + \delta$, $u_2 - \delta$ and $u_1 - \delta$, $u_2 + \delta$.
3. Intermediate transition type A_1^3 point. Initial points for a starting A_1^2 point are created at parameter values u_1 and u_2 , that correspond to a source point. They

are determined as two of the three contact points for which the line segment between them is closest to the circumcenter of the triangle formed by the three contact points on C .

Given an A_1^2 point at time t , a discrete marching algorithm is used to compute a new A_1^2 point at time $t + dt$ using Equation 4.10. Suppose, an A_1^2 point $P = \gamma(u_1, t)$ evolves to a point $Q = \gamma(u_2, t + dt)$ after a small time dt , then $u_2 = (u_1 + a dt)$. The corresponding parameter variable of $\hat{\gamma}$ can be computed as $\hat{u}_2 = (\hat{u}_1 + \hat{a} dt)$. a and \hat{a} can be obtained by solving Equations 4.8. In order to avoid numerical errors accumulating over time, Equation 4.11 with $r = t + dt$ is used to project points accurately onto A_1^2 curves and refine parameter values simultaneously. Starting from $t = 0$, Algorithm 2 is used to compute A_1^2 curves.

Algorithm 2 Computing A_1^2 curves

1. Sort transition points in order of increasing time.
 2. Increment t by small timestep dt .
 3. If no transition points are encountered, evolve all current A_1^2 points to $t + dt$.
 4. Otherwise, perform transition for each transition point encountered in increasing order of t and evolve intersection curves not involved in the transition event.
 5. If all transitions have been completed, then stop. Otherwise, repeat Step 2.
-

4.3.3 Alternate Approach for Computing Medial Curves

This section presents an alternate formulation of a vector that is tangent to A_1^2 curves away from transition points. This tangent vector is defined in parametric space and can be used in place of the evolution vector field to trace A_1^2 curve segments.

Define a mapping

$$F(u, \hat{u}, t) = \gamma(u, t) - \hat{\gamma}(\hat{u}, t) : \mathbb{R}_{u, \hat{u}, t}^3 \rightarrow \mathbb{R}^2, \quad u \neq \hat{u} \quad (4.12)$$

where $\mathbb{R}_{u, \hat{u}, t}^3$ is the augmented parameter space consisting of the parametric directions of the two curves and the time domain.

The Jacobian of the mapping F is given by

$$J_F = \begin{bmatrix} \gamma_u^{(1)} & -\hat{\gamma}_{\hat{u}}^{(1)} & \delta n^{(1)} \\ \gamma_u^{(2)} & -\hat{\gamma}_{\hat{u}}^{(2)} & \delta n^{(2)} \end{bmatrix} \quad (4.13)$$

where the superscript (j) represents the j -th coordinate of the corresponding vectors and $\delta n = n(u) - \hat{n}(\hat{u})$. Since J_F has the full rank, F is differentiable.

The preimage, \mathcal{I} , of $F(u, \hat{u}, t) = [0 \ 0]^T$ is the set of all intersection points of the two offset curves over all time values. Since the F is a differentiable function, according to the implicit function theorem, \mathcal{I} is a well-defined 1-manifold in $\mathbb{R}_{u, \hat{u}, t}^3$. See Figure 4.4 for an illustration.

Let $F^{(1)}$ and $F^{(2)}$ represent the first and second vector components of F . $F^{(1)} = 0$ and $F^{(2)} = 0$ represent hypersurfaces in $\mathbb{R}_{u, \hat{u}, t}^3$. \mathcal{I} is an implicit space curve that is the locus of intersection points of the two hypersurfaces. The normals to the hypersurfaces are given by their gradients, $\nabla F^{(1)}$ and $\nabla F^{(2)}$, respectively. The tangent to the intersection curve \mathcal{I} is given by [54]

$$T_{\mathcal{I}} = \nabla F^{(1)} \times \nabla F^{(2)} \quad (4.14)$$

The medial axis curve segments can then be traced by solving the differential equation

$$\frac{dx}{dt} = T_{\mathcal{I}}(x), \quad x(0) = P, \quad x(t) \in \mathbb{R}_{u, \hat{u}, t}^3 \quad (4.15)$$

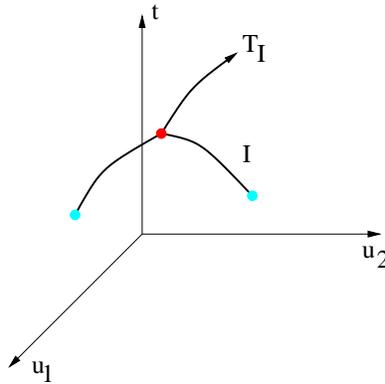


Figure 4.4. Medial axis in augmented parameter-time space $\mathbb{R}_{u, \hat{u}, t}^3$. \mathcal{I} is the locus of self-intersection points of the offsets of the boundary with a tangent $T_{\mathcal{I}}$ defined away from transition points.

Algorithm 3 is used to trace all A_1^2 curves from source to sink points. Transition points are first computed as presented in Section 4.2 and classified as source (creation) and sink (annihilation) points. In order to avoid numerical errors from numerical integration, points are projected onto \mathcal{I} at each step.

Algorithm 3 Tracing A_1^2 curves

INPUT SRC, SINK

OUTPUT m_{all} , the set of all A_1^2 curves

$m_{all} := \emptyset$

for $p := (u_1, \hat{u}_1, t) \in \text{SRC}$ **do**

$m := \{p\}$

$q := p$

while $q \notin \text{SINK}$ **do**

 Integrate Equation 4.15 to obtain new u_1, \hat{u}_1 at time $t + dt$

$q := (u_1, \hat{u}_1, t + dt)$

$m := m \cup \{q\}$

end while

$m_{all} := m_{all} \cup m$

end for

4.4 Results and Discussion

Figures 4.5 and 4.6 (a), (b) present the 2D medial axis computed using the proposed approach. In each example, the region of interest is bounded by a parametric quintic B-spline curve shown in gray. In the figures, A_3 points are indicated by cyan circles, A_1^2 critical points by green circles, A_1^3 points by red circles and A_1^2 curves by thick black curves. Also indicated by arrows on the A_1^2 curves is the shock graph structure [50] that is automatically computed using our approach.

Figure 4.7 presents a comparison of the medial axis computed in Figure 4.5 using a discrete Voronoi-based approach [53]. Results using two different samplings of the boundary are shown. The computed medial axis approximates the true medial axis more accurately as the sampling density increases. Figure 4.6 (c), (d) also present comparisons of medial axes computed using the discrete approach.

As explained in Section 4.1, the transition points are computed prior to computing A_1^2 curves of a medial axis. In addition to accurate points on the medial axis, the

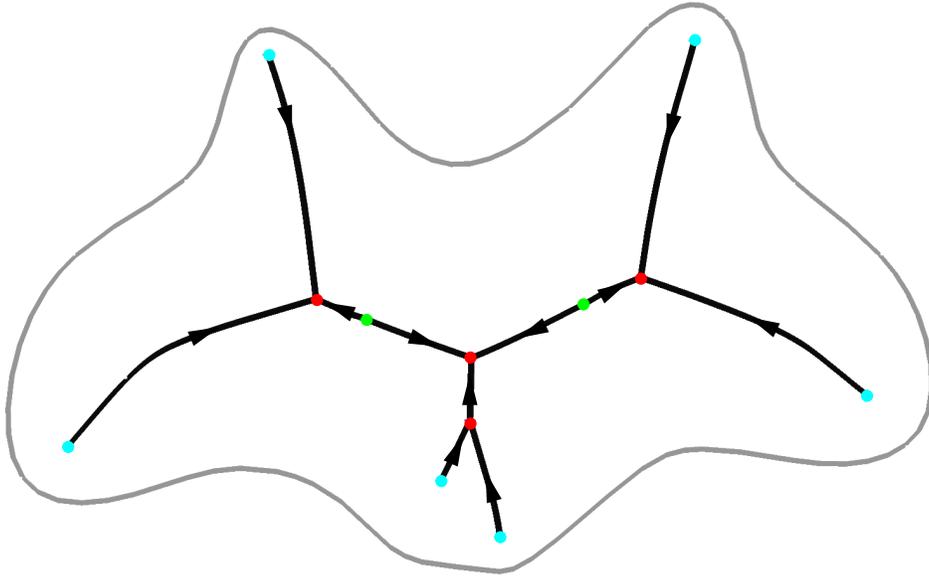


Figure 4.5. Medial axis of a region bounded by a B-spline curve.

proposed approach also accurately captures the topological structure of the medial axis. In addition, since the evolution tracing is performed in parametric space, the boundary locations corresponding to each medial axis point as well as the corresponding distance is also known. This gives the complete medial axis transform.

The computational complexity of the proposed algorithm is directly proportional to the number of transition points since it bounds the total number of A_1^2 curves for a given model. The example boundary curves shown in Figures 4.5 and 4.6 contain close to 50 control points. Running times on an Intel x64 machine with four cores and 8GB RAM were less than a second for A_3 , A_1^2 transition point computation, approximately one minute for A_1^3 point computation and approximately one second for A_1^2 curve tracing. Without the optimizations for the A_1^3 point computation, the running time was approximately 20 minutes for each example. Although the total running time is still longer than is desirable, the technique is automatic and presents accurate solutions with topology. Therefore, pruning and structure extraction steps are not required.

The techniques for computing transition points of A_3 and A_1^2 types provide all solutions. In order to reduce computation time for A_1^3 transition points, they were required to be at least further apart than 0.5% of the size of the parametric domain in

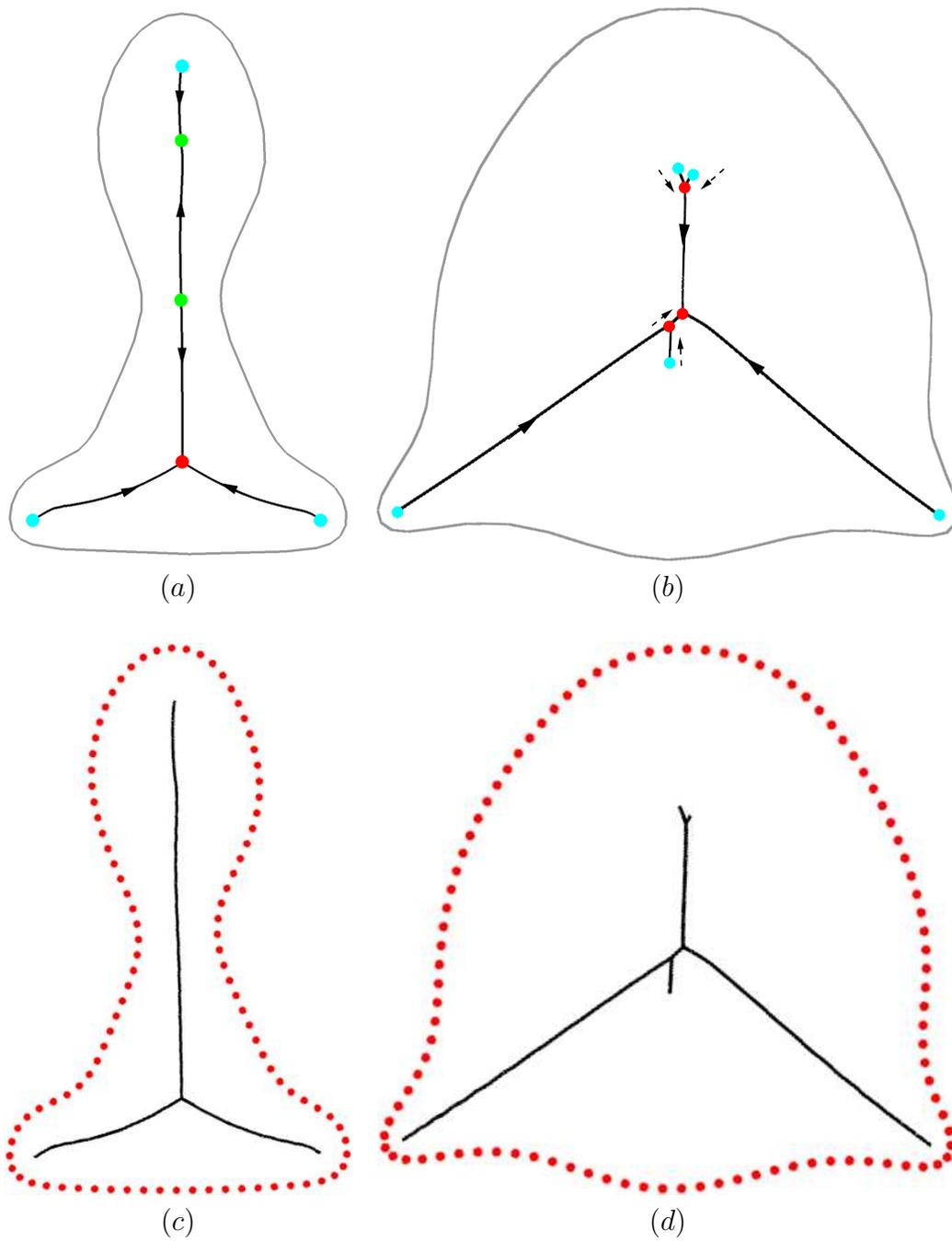
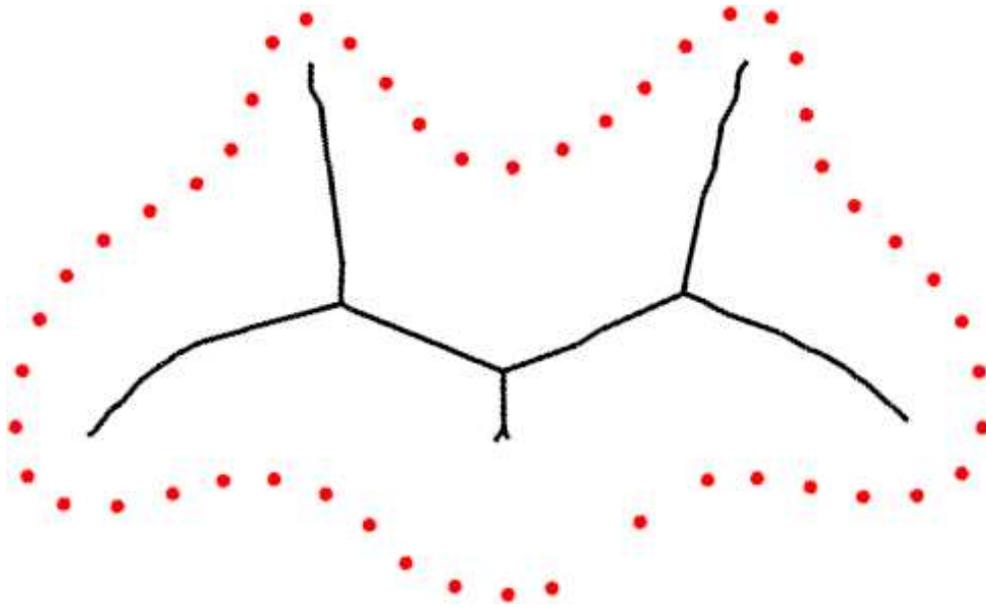
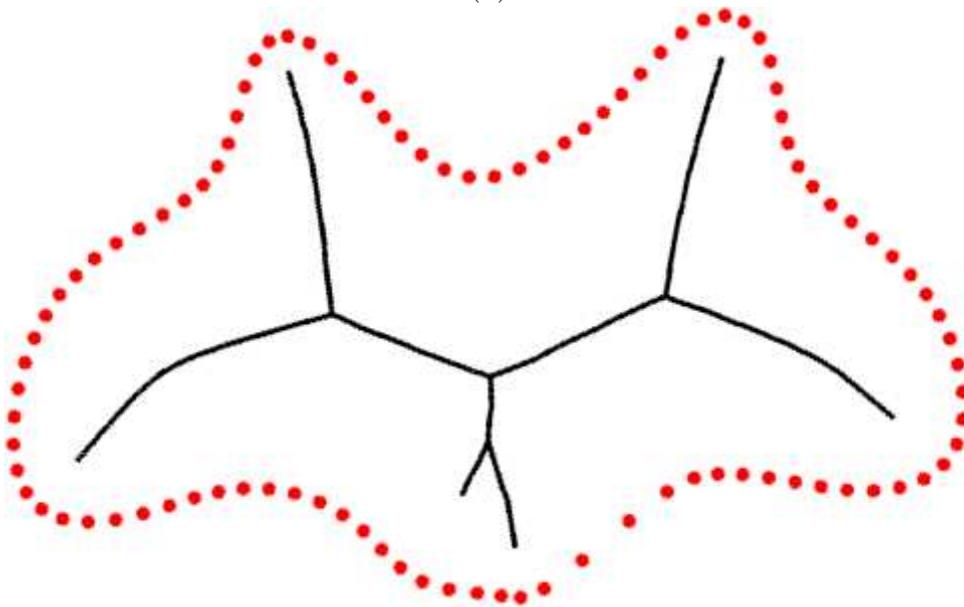


Figure 4.6. Examples of 2D medial axis computed directly from parametric B-spline boundary curves ((a), (b)) and using a discrete Voronoi-based method [53]((c), (d)). In (a) and (b), A_3 end points are shown in cyan, A_1^3 junction points in red and A_1^2 critical points in green. Arrows on the A_1^2 curves indicate the shock graph structure.



(a)



(b)

Figure 4.7. Examples of 2D medial axis using a discrete Voronoi-based method [53]. (a) Boundary sampled with 50 points. (b) Boundary sampled with 100 points. The discrete medial axis converges to the accurate result as sampling density increases.

the examples shown. Therefore, no two transition events could occur within a region of this size. The topology of the computed medial axis is accurate up to this specified accuracy. Between transition events, a much higher accuracy (10^{-3} of the size of the model in \mathbb{R}^2) was used to locate the medial axis points since the evolution of the intersection curves is smooth.

4.5 Summary

This chapter presents a new technique to automatically compute accurate and topologically correct (up to user specified numerical accuracy) interior medial axes of planar regions bounded by single closed parametric B-spline curves. The technique is based on tracking self-intersections of offsets of the boundary curves under the eikonal flow. Transition points of the self-intersections are computed in advance and thus accurate topology is guaranteed. Further, the method also indicates the direction of increasing radius of maximal circles at each transition point, giving the shock graph structure [50].

Our method is more general than the approach presented in [105] for computing the medial axis of a planar region since convex corner vertices are not required. All begin/end points of medial axis segments are computed in advance and therefore our method avoids computationally expensive distance and curvature checks used during tracing in that method.

The technique presented in this chapter is extended for the 3D case to compute the medial axis of regions bounded by B-spline surfaces. First, Chapter 5 presents a new technique for accurately computing ridges on B-spline surfaces. Edge curves of the medial axis are computed as a subset of the computed ridges. Chapter 6 then presents the technique for computing the 3D medial axis.

CHAPTER 5

RIDGES

This chapter presents a new tracing algorithm to compute all generic ridges on a surface accurately in the form of connected curve segments. The algorithm traverses ridge segments by detecting ridge points while advancing and sliding in principal directions on a surface in a novel manner, thereby computing connected curves of ridge points. This chapter presents a technique to compute ridges on tensor product parametric B-spline surfaces. This technique is then generalized to extract ridges from isosurfaces of volumetric data represented using trivariate implicit B-splines (See Section 7.3).

The input surface, $S(u, v)$, is assumed to be regular (i.e., $S_u \times S_v \neq 0$), having only isolated umbilics, and ridges that exhibit only generic properties as specified in Section 1.5.1. The surface is also required to be C^3 smooth, in order to have continuous first order derivatives of principal curvatures. κ_1 -ridges and κ_2 -ridges are traced separately. In our discussion, we present the algorithm for tracing ridges corresponding to the maximum principal curvature (κ_1). The tracing procedure for κ_2 -ridges is similar, and the differences are indicated at the end of the section after an overview of the algorithm has been presented.

5.1 Overview

Traces are started at three types of *seed points* including 1) critical points of curvature, 2) umbilics, and 3) ridge points on the surface boundaries. Curvature critical points trivially satisfy the ridge condition since the curvature gradient is identically zero at these locations. Umbilics are also included as seed points since these are the only points on a surface where generic ridges of the same type may meet. Recall from Section 1.5.1 that ridges either form closed curves, or end at only umbilics or points on the surface boundary. All ridges on a surface pass through at least one type of seed

point. When a ridge forms a closed loop, it passes through a curvature extremum point (See Proposition 5.1.1). Therefore, no ridge is missed when these seed points are used for tracing.

Proposition 5.1.1 *A closed generic ridge curve has at least one principal curvature critical point on it.*

Proof. Consider a closed κ_1 -ridge curve c . Then, as κ_1 is continuous on c , it takes on a maximum and minimum value. Let x_0 be a point where the maximum occurs. If v is a tangent vector of c at x_0 , then $\frac{d\kappa_1}{dv}|_{(x_0)} = 0$. If x_0 is not a turning point, then the principal curve p_1 for κ_1 through x_0 is transverse to c at x_0 . Let w be a tangent vector for p_1 at x_0 . By the definition of a κ_1 -ridge point, κ_1 has a critical point along p_1 at x_0 , so $\frac{d\kappa_1}{dw}|_{(x_0)} = 0$. Hence, as w and v are independent, x_0 is a critical point for κ_1 .

Further, x_0 is not a turning point for the ridge. If it were, then both $\frac{d\kappa_1}{dv}|_{(x_0)} = 0$ (x_0 is a local maximum for κ_1 along c), and x_0 a turning point implies $\frac{d^2\kappa_1}{dv^2}|_{(x_0)} = 0$. However, generically when the first two derivatives are zero, $\frac{d^3\kappa_1}{dv^3}|_{(x_0)} \neq 0$. Thus, $\frac{d^2\kappa_1}{dv^2}$ changes sign at x_0 , so it is an inflection point and hence not a local maximum. Hence, generically a local maximum will not be a turning point. ■

The tracing strategy is based on the property that κ_1 -ridges intersect the κ_1 curvature lines transversally except at a few isolated turning points. The idea is to trace a κ_1 curvature line to a zero of the ridge function, where the κ_1 curvature line intersects a ridge. In order to progress to the next trace point, the algorithm steps along the κ_2 curvature line (T_2 direction) and then traces the κ_1 curvature line from the new location. Since the curvature lines are orthogonal, the algorithm is guaranteed to progress further along a ridge.

Each trace consists of several progress operations. Each progress operation consists of two steps viz., an *advance step*, and a *slide step*, as illustrated in Figure 5.1.

1. Advance step - compute a new point in the tangent plane of the current ridge point in the Euclidean minimum principal direction (T_2) and project it onto the surface.

Step sizes are varied adaptively (See Section 5.3). A trace ends when it reaches either another seed point or a parametric domain boundary. A new trace is also computed from the same seed point but by advancing in the opposite ($-T_2$) direction. Special care is needed when the trace is close to a turning point and when the trace is started at an umbilic. The following sections present details on computing seed points and the different tracing steps.

The algorithm for tracing κ_2 -ridges differs in that the advance step is done along the maximum principal direction (T_1) and the slide step is performed along the minimum principal direction (T_2).

5.2 Computing Seed Points

This section presents systems of equations required to compute curvature critical points and umbilics based on [85, 97]. A robust and efficient subdivision-based constraint solving technique [43, 45] is used to compute the roots of relevant piecewise rational equations.

5.2.1 Curvature Critical Points

Critical points of curvature occur at the locations on a surface where the curvature gradient is identically zero. Using the notation introduced in Section 3.1.2.1, and writing both principal curvatures in one equation,

$$\kappa(u, v) = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (5.1)$$

It is necessary to solve for simultaneous roots of

$$\begin{aligned} \kappa_u(u, v) &= 0, \\ \kappa_v(u, v) &= 0. \end{aligned} \quad (5.2)$$

$B(u, v)$, and hence κ_u and κ_v , are not rational functions since they contain the coefficients of the second fundamental form, which in turn have a square root in the denominator. They are converted to rational functions as follows. Noting that $\|S_u \times S_v\| = \sqrt{A}$,

$$\begin{aligned}
L &= \frac{\widehat{L}}{\sqrt{A}}, & M &= \frac{\widehat{M}}{\sqrt{A}}, & N &= \frac{\widehat{N}}{\sqrt{A}} \\
B &= \frac{\widehat{B}}{\sqrt{A}}, & C &= \frac{\widehat{C}}{A} \\
\kappa(u, v) &= \frac{-\widehat{B} \pm \sqrt{\widehat{B}^2 - 4A\widehat{C}}}{2A^{\frac{3}{2}}}
\end{aligned} \tag{5.3}$$

where \widehat{L} , \widehat{M} , \widehat{N} , \widehat{B} , \widehat{C} are piecewise polynomial or piecewise rational depending on whether $S(u, v)$ is piecewise polynomial or rational, respectively.

The first order derivatives of $\kappa(u, v)$ are given by,

$$\begin{aligned}
\kappa_u &= P^{(u)} \pm \frac{R^{(u)}}{\sqrt{Q}} = 0 \\
\kappa_v &= P^{(v)} \pm \frac{R^{(v)}}{\sqrt{Q}} = 0
\end{aligned} \tag{5.4}$$

where,

$$\begin{aligned}
P^{(u)} &= \frac{1}{2}[(-A^{-\frac{3}{2}}\widehat{B}_u + \frac{3}{2}A^{-\frac{5}{2}}A_u\widehat{B})] \\
P^{(v)} &= \frac{1}{2}[(-A^{-\frac{3}{2}}\widehat{B}_v + \frac{3}{2}A^{-\frac{5}{2}}A_v\widehat{B})] \\
R^{(u)} &= \frac{1}{2}[(A^{-\frac{3}{2}}\widehat{B}_u\widehat{B} - 2A^{-\frac{1}{2}}\widehat{C}_u + 4A^{-\frac{3}{2}}A_u\widehat{C} - \frac{3}{2}A^{-\frac{5}{2}}A_u\widehat{B}^2)] \\
R^{(v)} &= \frac{1}{2}[(A^{-\frac{3}{2}}\widehat{B}_v\widehat{B} - 2A^{-\frac{1}{2}}\widehat{C}_v + 4A^{-\frac{3}{2}}A_v\widehat{C} - \frac{3}{2}A^{-\frac{5}{2}}A_v\widehat{B}^2)] \\
Q &= \widehat{B}^2 - 4A\widehat{C}
\end{aligned} \tag{5.5}$$

Note that,

$$\begin{aligned}
\kappa_{1u} &= P^{(u)} + \frac{R^{(u)}}{\sqrt{Q}}, & \kappa_{1v} &= P^{(v)} + \frac{R^{(v)}}{\sqrt{Q}} \\
\kappa_{2u} &= P^{(u)} - \frac{R^{(u)}}{\sqrt{Q}}, & \kappa_{2v} &= P^{(v)} - \frac{R^{(v)}}{\sqrt{Q}}
\end{aligned} \tag{5.6}$$

Moving the terms with the square root in Equation 5.4 to the right hand side, squaring both sides and simplifying we get,

$$\begin{aligned}
QP^{(u)2} - R^{(u)2} &= 0 \\
QP^{(v)2} - R^{(v)2} &= 0
\end{aligned} \tag{5.7}$$

The above equations encode the critical points of both κ_1 and κ_2 . After solving for the roots of the above system of equations, they are classified as critical points of κ_1 or κ_2 by evaluating Equation 5.6.

5.2.2 Umbilics

At umbilics, $\kappa_1 = \kappa_2$. Therefore, from Equation 5.1 it is apparent that,

$$Q(u, v) = \widehat{B}^2 - 4A\widehat{C} = 0$$

$$\widehat{B} = \frac{B}{\sqrt{A}} \tag{5.8}$$

$$\widehat{C} = \frac{C}{\sqrt{A}}$$

In addition, $Q(u, v)$ attains a minimum at the umbilic (since $Q(u, v) \geq 0$). Therefore, the roots of the following system of equations are computed.

$$\frac{\partial Q(u, v)}{\partial u} = 2\widehat{B}\widehat{B}_u - 4A_u\widehat{C} - 4A\widehat{C}_u = 0$$

$$\frac{\partial Q(u, v)}{\partial v} = 2\widehat{B}\widehat{B}_v - 4A_v\widehat{C} - 4A\widehat{C}_v = 0$$
(5.9)

Equation 5.8 is then evaluated to ensure $Q(u, v) = 0$ (since there may be local extrema of $Q(u, v)$ that do not occur at umbilics).

5.2.3 Boundary Ridge Points

Boundary seeds are computed as zeros of the following four univariate versions of the ridge equation:

$$\phi_i(\tau_{begin}^{(u)}, v) = 0 \tag{5.10a}$$

$$\phi_i(\tau_{end}^{(u)} - \epsilon, v) = 0 \tag{5.10b}$$

$$\phi_i(u, \tau_{begin}^{(v)}) = 0 \tag{5.10c}$$

$$\phi_i(u, \tau_{end}^{(v)} - \epsilon) = 0 \tag{5.10d}$$

where ϕ_i is the ridge function for κ_i , and $\tau_{begin}^{(u)}$, $\tau_{end}^{(u)}$, $\tau_{begin}^{(v)}$, $\tau_{end}^{(v)}$ are the parameter values corresponding to the boundary knots of the domain of $S(u, v)$. Since the upper end

of each knot interval is open (Section 3.2), the isoparametric knot lines at $\tau_{end}^{(u)}$ and $\tau_{end}^{(v)}$ are not considered part of the boundary and ridge seeds are computed at $\tau_{end}^{(u)} - \epsilon$ and $\tau_{end}^{(v)} - \epsilon$ instead. Since $\phi_i(u, v)$ is not piecewise rational, the equation $\phi_i(u, v) = 0$ is converted into $\tilde{\phi}_i(u, v) = 0$ by rearranging terms and squaring so that $\tilde{\phi}_i(u, v)$ is piecewise rational.

5.3 Tracing

As mentioned in Section 5.1, each trace consists of a series of advance and slide steps. For both these steps, consistent orientation of the principal directions and prudent step sizes must be chosen in order to successfully trace a ridge. We discuss each operation with respect to tracing a κ_1 -ridge. The strategy for tracing at umbilics and the technique used for projecting points onto the surface at every step are also presented.

5.3.1 Advance Step

5.3.1.1 Orientation

At every advance step, it is necessary to ensure that the new T_2 vector is along the same direction as the previous T_2 vector and not opposite (by ensuring that the angle between the vectors is acute). The heuristic, called the acute angle rule [22], has been used for tracing ridges on polygonal meshes.

5.3.1.2 Step Size

A judicious choice of step size is critical when two ridges are close. At every advance step, an initial step size δ_0 is first selected¹. Let $r_i \in R^2$ be the parameter values for the current ridge point and $r_j \in R^2$ be the parameter values for the point arrived at by advancing along the T_2 direction and projecting onto the surface. Let $\phi(u, v)$ represent the ridge condition for the current principal curvature. $\gamma(w) = (u(w), v(w)) = r_i + w(r_j - r_i)$, $w \in [0, 1]$ is the line segment joining r_i and r_j . $\phi(\gamma(w))$ is the corresponding curve segment of the ridge function between r_i and r_j . In order to guarantee robustness, the trace must not slide to either a local extremum of $\phi(u, v)$ or an adjacent ridge

¹In our experiments, initial step sizes of 0.1% of the length of the diagonal of the bounding box of the surface worked well.

segment (See Figure 5.2). This condition can be enforced by ensuring that $\phi(\gamma(w))$ does not have any local extrema. The test would then involve checking whether or not the graph of $\phi(\gamma(w))$ has a zero slope at any point. However, computation of the slope requires higher order surface smoothness. In addition, $\phi(\gamma(w))$ is not a rational function. Therefore a computationally efficient approach similar to a Monte Carlo method requiring only samples of $\phi(\gamma(w))$ is used. The interval $w = [0, 1]$ is sampled randomly and $\phi(\gamma(w))$ is evaluated at the samples. The robustness test then checks if the samples of $\phi(\gamma(w))$ are monotonic with respect to w . δ_0 must be reduced until this condition is satisfied.

The advance step size is additionally varied adaptively during the trace depending on nearness to a turning point. The step size can be additionally scaled using curvature magnitude and curvature gradient magnitude at every step. Initially, at a seed point, there is no information about the ridge direction. From the next advance step onward the ridge direction is tracked using the previous trace points. The angle between the ridge direction and the T_1 direction computed at the current location is related to the proximity of a turning point. An angle close to zero implies that a turning point is

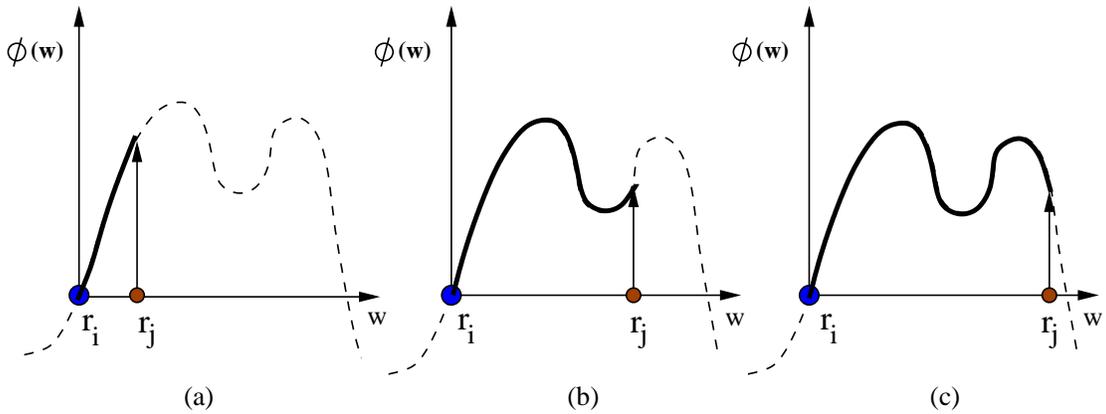


Figure 5.2. Robust initial advance step size selection. r_i is a ridge point and r_j is the advance point. The graph of $\phi(w)$ between r_i and r_j ($w \in [0, 1]$) is shown as a thick curve. Broken curve segments indicate $\phi(w)$, $w < 0$, $w > 1$. δ_0 should be chosen such that $\phi(w)$ does not have any local extrema between r_i and r_j . a) indicates a correct step size selection. b) and c) indicate incorrect step size selections. In case b), the trace will get stuck in a local minimum of $\phi(w)$ and will not reach a ridge. In case c), the trace will converge to an adjacent ridge segment.

very close. The step size is reduced accordingly during the trace until it falls below a threshold (`turning_point_stepsize_threshold`²). Once it falls below the threshold, the orientation of the T_2 vector is reversed since the ridge will now progress in the opposite direction. From the next advance step onward, the trace will use the new orientation of the T_2 vector. The ridge progress direction is used to avoid backtracking along the previously computed trace after a flip. The adaptive step size variation in the vicinity of a turning point is illustrated in Figure 5.3. Also, using a larger step size immediately after detecting a potential turning point and searching for a ridge by sliding from an advance step in both T_2 and $-T_2$ directions helps detect a geodesic inflection point of the ridge.

5.3.2 Slide Step

After an advance step is done, the slide begins with an initial step size and a local search is performed for a ridge in both T_1 and $-T_1$ directions. Initial step sizes can be robustly selected using the technique presented for advance step size selection. Figure 5.4 shows a sample sliding scenario. The values of the ridge function at the

²We have used a threshold value of 10^{-6} in our experiments.

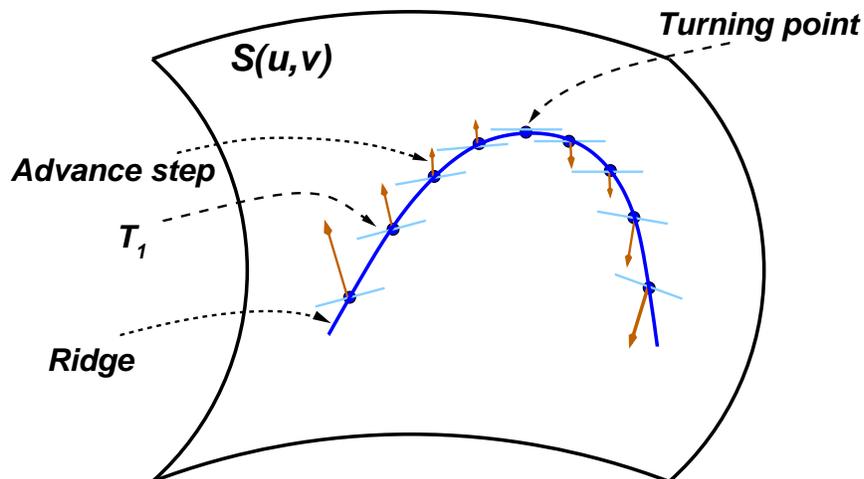


Figure 5.3. Advance step size (turning point aware). Advance steps are shown in brown, maximum curvature principal directions are shown in pale blue, ridge is shown in dark blue.

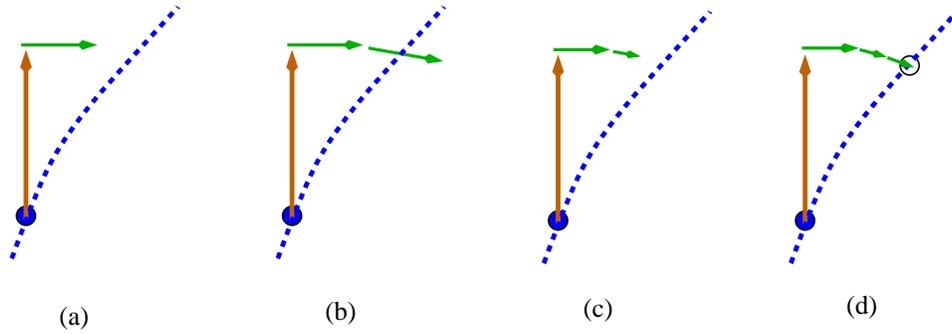


Figure 5.4. Sliding to a ridge. Advance step is shown in brown, slide steps are shown in green. a) First slide step is moving toward ridge but has not yet reached ridge, b) Second slide step has crossed ridge, c) Slide is recomputed with reduced step size, new slide point has not yet reached ridge, d) Slide has reached ridge after a few steps of b) and c).

current location and a step from the current location in the T_1 direction are compared. If the ridge functions at the two points have the same sign and have increasing magnitude, a slide is not performed in that direction. If they have the same sign and are decreasing in magnitude, the new location is accepted and the slide is repeated from the new location. If the signs are different (implying that the slide crossed a ridge), the slide step size is reduced³ and a new location is recomputed from the current point along the T_1 direction at the current point. This process is repeated iteratively until the ridge condition falls below a specified threshold (`ridge_accuracy_threshold`)⁴. A local acute angle heuristic is used to select consistent T_1 vector orientations.

5.3.3 Tracing from Umbilics

The algorithm sweeps around umbilics using the principal curvature directions to detect ridges. If a ridge is found, a seed point is created and a trace is started in the direction away from the umbilic (See Figure 5.5 for illustration).

Recall from Section 3.1.2.4 that there are either one or three pairs of minimum and maximum curvature directions oriented opposite each other. In all of these cases, a scout point is created at a small distance from the umbilic along each of the minimum

³In our experiments, we found that halving the step size works well.

⁴`ridge_accuracy_threshold` value of 10^{-3} was used in our experiments.

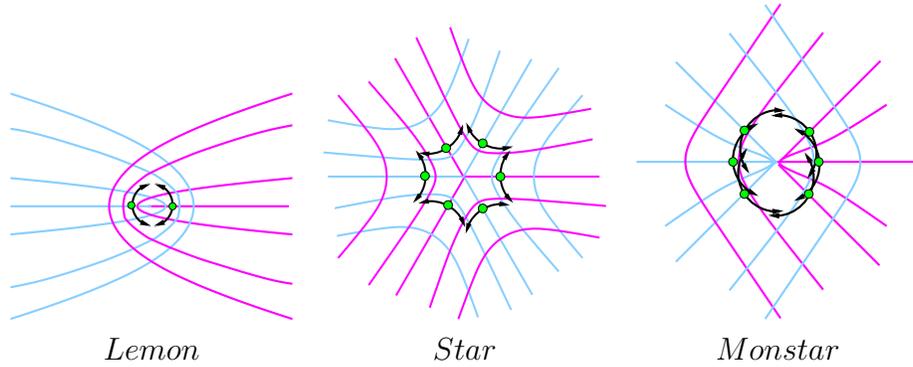


Figure 5.5. Tracing around umbilics. Scout points are shown in green.

curvature directions and traces are started from each of them as presented in the previous section. If the scout does not detect a ridge, it will stop automatically. If the scout does detect a trace, it will continue as if it were tracing from any other regular point. In addition, scout points are also created along the maximum principal directions in order to completely sweep around the umbilic.

It should be noted that it is possible to trace the same ridge multiple times (from start to end and reverse). At umbilics, a ridge can be traced multiple times from different seed points. Duplicate ridges are detected by inspecting the start and end points of the respective traces. In the latter case, the start points are the same umbilic point. In addition, the curvature lines may have a large geodesic curvature very close to an umbilic. The local acute angle rule may not guarantee consistent orientations in such cases, as noted by [22]. Therefore, to avoid such situations, the scout points must not be created too close to an umbilic. Tracing κ_2 ridges from umbilics is identical, since scout points are created in all principal directions.

5.3.4 Projecting Points onto Surface

At every step, when a motion is performed in either principal direction in the tangent plane of the surface, it is necessary to project the point onto the surface. In order to find the point on the surface $S(u, v)$ (and the corresponding parameter values) closest to a given point $X \in R^3$, a global approach involves solving the following system of equations.

$$\begin{aligned}
\langle S_u(u, v), (S(u, v) - X) \rangle &= 0 \\
\langle S_v(u, v), (S(u, v) - X) \rangle &= 0
\end{aligned} \tag{5.11}$$

The solution set of this system of equations gives all points on the surface where the vector from the point X to a point on the surface is in the direction of the surface normal at that point. The actual closest point is determined by computing the distances from X to all the solutions and selecting the nearest one.

The global approach is too slow since the tracing algorithm may involve a very large number of projection operations. In this work, a two-dimensional Newton's method is used to find the closest point on the surface. This technique is very fast and has been used for interactive applications that require computing closest points at a very large rate (several hundred times a second) [70]. Since the step sizes used in the algorithm are typically very small, this works well. In the event that the Newton's method fails to give accurate results, the algorithm reverts to the global method. In our experiments, this situation did not occur very often. The two-dimensional Newton's method involves solving the following linear system of equations for variables u and v .

$$\begin{bmatrix} \frac{\partial(\langle S_u, Z \rangle)}{\partial u} & \frac{\partial(\langle S_u, Z \rangle)}{\partial v} \\ \frac{\partial(\langle S_v, Z \rangle)}{\partial u} & \frac{\partial(\langle S_v, Z \rangle)}{\partial v} \end{bmatrix} \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} = - \begin{bmatrix} \langle S_u, Z \rangle \\ \langle S_v, Z \rangle \end{bmatrix} \tag{5.12}$$

$$Z = S - X$$

The Jacobian matrix can be expanded as,

$$\begin{bmatrix} \langle S_{uu}, Z \rangle + \langle S_u, S_u \rangle & \langle S_{uv}, Z \rangle + \langle S_u, S_v \rangle \\ \langle S_{uv}, Z \rangle + \langle S_u, S_v \rangle & \langle S_{vv}, Z \rangle + \langle S_v, S_v \rangle \end{bmatrix} \tag{5.13}$$

Symbolic representations of the partial derivatives of the surface are precomputed so that they can be evaluated quickly for the projection operations. The parameter values of the current advance or slide point are used as the initial point (u_0, v_0) . This system is solved iteratively until the error is small enough. The error is computed as the residual from the evaluation of Equation 5.11.

5.4 Results and Discussion

We first present results for a simple biquartic Bézier patch (See Figure 5.6). This surface was selected to allow direct comparison with the results presented in [21] (See Figures 8.4 and 8.5 therein), which are topologically correct at all locations including umbilics. The results are best compared in parametric space. Figure 5.6 (b) and Figure 8.4 of [21] are indeed very similar.

We also present ridges traced on complex models from different application domains including a human femur bone model and a terrain elevation map [61]. The surfaces are represented by tensor product biquartic B-splines. The results are compared with a brute force sampling of the ridge function in the parametric domain in Figures 5.7 and 5.8. Soft ridges, similar to the style presented in [69], are computed to present a better visualization. However, there is no topology associated with the sampled ridges. The images showing the sampled κ_1 -ridges are colored with regions varying from blue fading into yellow. Images of the sampled κ_2 -ridges are colored with regions varying from red fading into cyan. Darker blue colors in the former, and brighter red in the latter images correspond to regions closer to κ_1 and κ_2 ridges, respectively. For these examples, C^3 is sufficient since crests are not identified. The fourth order derivatives are not continuous only at the knots and since there are only a few knots for both examples, crests can be identified accurately away from them.

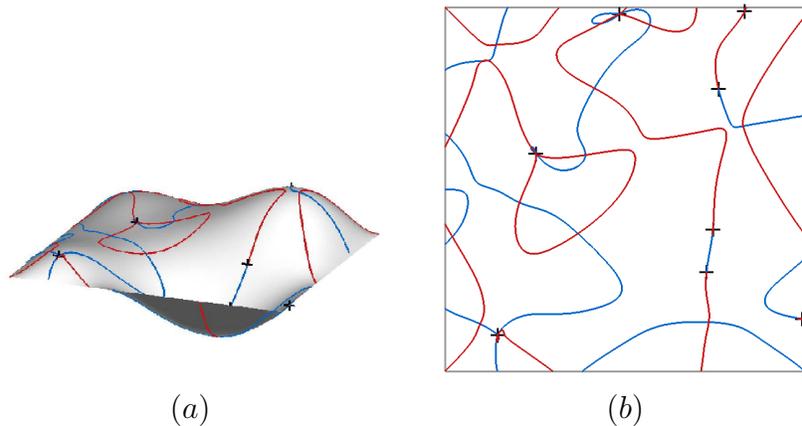


Figure 5.6. Ridges on a Bézier patch. κ_1 -ridges are in blue and κ_2 -ridges are in red. (a) Euclidean space. (b) Parametric space. Black cross-hairs indicate umbilics.

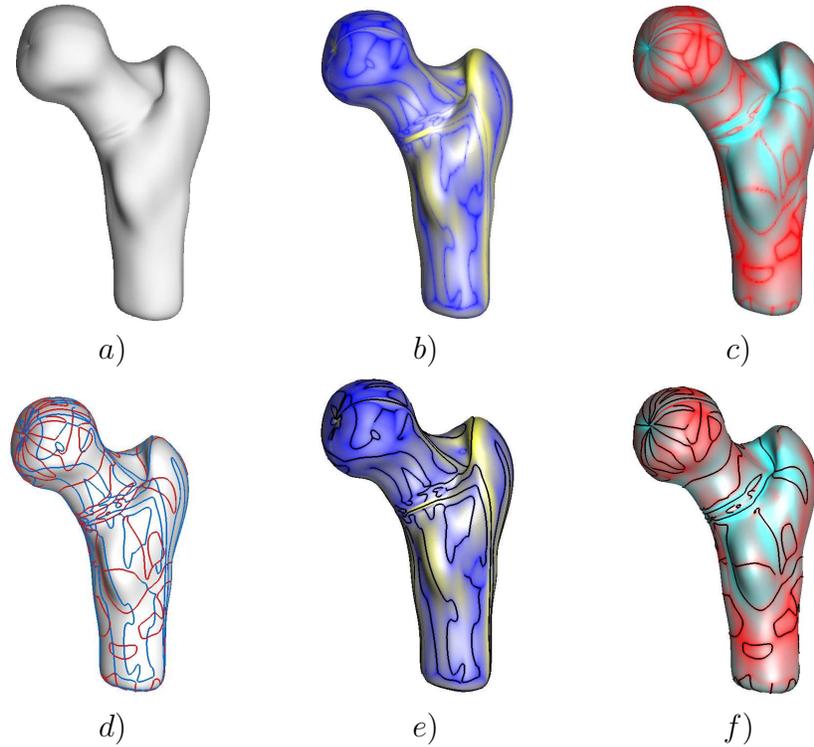


Figure 5.7. Results of ridge computation on a Femur bone model. a) B-spline representation of bone, b) Sampled κ_1 ridges (darker blue indicates ridge proximity), c) Sampled κ_2 ridges (brighter red indicates ridge proximity), d) Ridges overlaid on surface ($\kappa_1 \rightarrow$ blue, $\kappa_2 \rightarrow$ red), e) Traced κ_1 ridges (black) overlaid on b), f) Traced κ_2 ridges (black) overlaid on c)

The sampling approach can falsely indicate the presence (false positives) or absence (false negatives) of ridges. In Figure 5.9, the rectangular outline region shows an example where the presence of a κ_1 -ridge is falsely indicated on the terrain elevation model. The κ_1 ridge samples indicate the presence of a ridge. However, a close inspection of the κ_2 ridge samples and the traced ridges indicate that there is a κ_2 -ridge in that region, which is verified by the topology of the traced ridges in the surrounding region. False positives (+ve) occur when the magnitude of the ridge function is small enough to pass a threshold used for coloring the samples, but not zero. Figure 5.9 also shows an example where the sampling approach fails to detect a κ_2 -ridge (elliptic outline region) of the terrain elevation model, but is accurately captured using the tracing approach. False negatives(-ve) occur when width of the ridges is narrower (which can be arbitrarily narrow) than the sampling fineness. Figure 5.9.(b) shows that the tracing

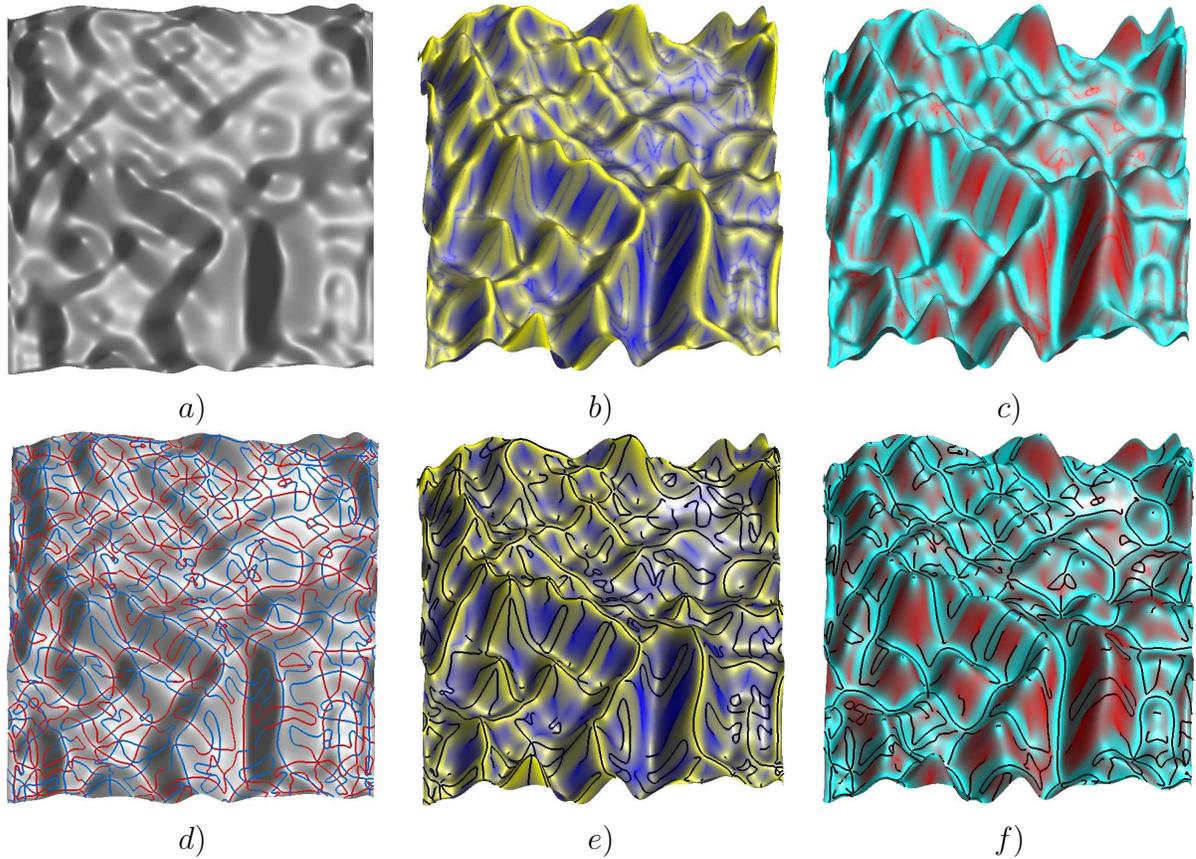


Figure 5.8. Results of ridge computation on a terrain elevation data set. a) B-spline approximation of the elevation data, b) Sampled κ_1 ridges (darker blue indicates ridge proximity), c) Sampled κ_2 ridges (brighter red indicates ridge proximity), d) Ridges overlaid on surface ($\kappa_1 \rightarrow$ blue, $\kappa_2 \rightarrow$ red), e) Traced κ_1 ridges (black) overlaid on b), f) Traced κ_2 ridges (black) overlaid on c)

approach presented here avoids the problems associated with sampling-based techniques and accurately captures ridge behavior.

To total number of seed points for the different models are presented in Table 5.1. Computational aspects for the different models on a single CPU are compared in Table 5.2. The computation times vary depending on the complexity of the models, not only in terms of representation size, but also in terms of the features on the surfaces, on the accuracy of the ridge tracing and the average trace length. We used a ridge accuracy of 10^{-3} for all data sets. The femur and the terrain models are quite complex, and our technique gives results in a few minutes. In comparison, an ODE based tracing

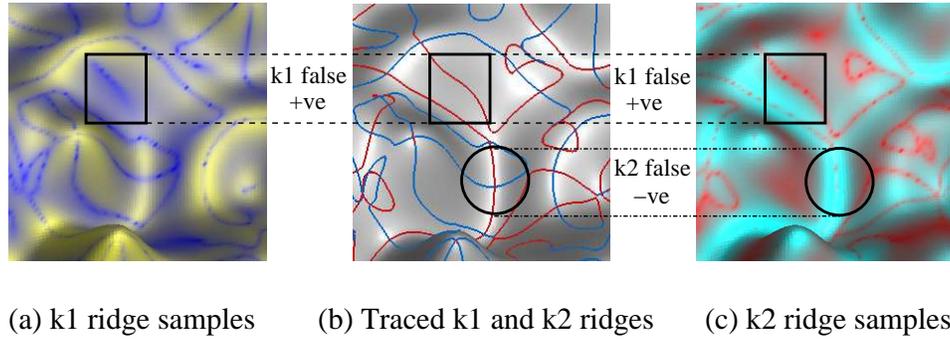


Figure 5.9. An example of a false positive in κ_1 -ridge samples (rectangle outline) and a false negative (ellipse outline) in κ_2 -ridge samples of the terrain elevation model. (a), (b) and (c) represent enlarged views of κ_1 -ridge samples, traced ridges and κ_2 -ridge samples of the same region of the terrain elevation model.

Table 5.1. Seed Points

Model	Control Mesh Size	# Curvature Critical Points	# Umbilics
Bézier patch	5×5	23	8
Femur	22×21	458	12
Terrain	20×20	508	314

Table 5.2. Computation Characteristics (on an Intel 2.4GHz Processor with 8GB Memory)

Model	Seed Points Time (minutes)	Ridge Tracing Time (minutes)
Bézier patch	0.08	0.07
Femur	7.03	3.68
Terrain	7.5	14.96

method took several hours to compute ridges with the same accuracy.

While our algorithm is not designed for nongeneric situations, end points of non-generic ridges that stop within the surface boundary are detected in our algorithm when the trace cannot detect any ridge during the slide step. Nongeneric ridges of the same type may cross each other. The technique presented here does not capture the topology at the junctions of ridges of the same type. This is an area for future work. Some nongeneric ridge segments may be missed if the algorithm does not find seed points

in those segments. Determining seed points that are neither curvature critical points nor umbilics to account for nongeneric ridges is also an area for future work. However, since the curvature critical points and umbilics represent important feature points on a surface, our algorithm presented is guaranteed to capture salient ridges on a surface. The constraint solver for computing seed points may give extraneous roots if too large a tolerance is allowed. These false roots will result in traces that end within a few steps, which is a nongeneric situation. Such ridge traces are detected and removed.

Most of the previous applications use crests mainly because there are few methods for robust extraction of all types of ridges. Noncrest ridges are more sensitive to subtle variations in geometry than crests and along with their topology, indicate higher order local geometric variation. These local curvature variations may not be desirable for smooth product designs. Therefore noncrest ridges are useful for evaluating product designs where undesirable curvature variations of a freeform surface are detected [69]. Since these are higher order surface properties, they may not be immediately perceptible even on high quality renderings of the objects. Noncrest ridges are useful for statistical shape analysis tasks over a group of similar objects such as anatomical organs. Since crests are more stable, they may occur at very similar locations and may seem to have similar structure across the group of objects. In this case, the sensitivity of noncrest ridges to local geometric variation will reveal additional geometric differences. In addition, computing the full set of ridges helps in understanding the relationship between crests and the topological structure of ridges. Noncrest ridges may connect two seemingly separate crest segments. This information is useful for shape analysis tasks. Umbilics represent important surface features and have been used for shape fingerprinting [75]. Since noncrest ridges exhibit topological changes at umbilics, it is also essential to compute ridges around umbilics accurately.

5.5 Summary

Ridges are important feature curves and have a wide variety of applications. Umbilics and therefore, ridges around umbilics, also represent important aspects of the shape of a surface. Ridges exhibit complex behavior around umbilics. This chapter presents

a new numerical tracing technique for accurately computing ridges on B-spline surfaces that has been designed using generic properties of ridges.

The tracing algorithm involves traversing curvature lines in a novel manner and accurately captures the behavior of ridges at all points on a surface including umbilics. The technique takes into account turning points without directly computing them, thereby allowing ridge computation on C^3 models, instead of requiring C^4 smoothness. Our technique has been designed for rational tensor product B-spline surface representations. Since ridge computation is local to a tensor product patch, it is directly extensible for models with multiple patches. Some special cases, such as ridges parallel to a domain boundary, may need to be addressed. Trimmed freeform surfaces also may be addressed by minor modifications to the algorithm. The algorithm design enables optimization using parallel processing techniques, which would further improve computation time. The approach can be further extended to surfaces with isolated irregular points. The technique presented in this dissertation avoids errors in ridge computation associated with sampling-based approaches, while at the same time, it can generate results for complex models that were previously computationally intractable.

The result is a set of polyline segments that are available for other applications, such as surface segmentation, matching, quality control and visualization. We utilize computed ridges to identify edge curves of the 3D medial axis. Chapter 6 presents the technique for computing the 3D medial axis.

CHAPTER 6

3D MEDIAL AXIS

This chapter presents a new approach for computing the complete topologically correct interior medial axis of three-dimensional regions directly from parametric B-spline representations of their boundary surfaces. Recall from Section 1.2.2 that the medial axis consists of surfaces bounded by edge curves, and branch curves where surfaces meet. The medial axis also contains fin points where edge curves meet branch curves, and six junction points where six surfaces (and four branch curves) meet at a point. Precise mathematical definitions of each entity type are presented in that Section. Our approach computes all entities of the medial axis with arbitrary user specified accuracy along with correct topology. Parameter values of boundary surface points corresponding to every medial axis point and the respective distance is also computed, which gives the medial axis transform. The chapter is organized as follows. An overview of the proposed algorithm is presented in Section 6.1 with details in Sections 6.2, 6.3 and 6.4. Results are presented in Section 6.5 and a summary of the chapter is presented in Section 6.7.

6.1 Overview

We first present the approach when the region is bounded by a single closed surface. Section 6.6 presents extensions of the approach to address regions specified by more than one surface stitched together.

Let \mathcal{B} be the boundary of a region in \mathbb{R}^3 represented by a closed tensor product parametric B-spline surface $S(u, v) \in C^{(4)}$. The surface normal, $n(u, v) = \frac{S_u \times S_v}{\|S_u \times S_v\|}$ (assumed oriented inward for a closed surface, with $\|S_u \times S_v\| \neq 0$) where subscripts indicate the partial derivatives with respect to the corresponding parameter variable. The offset surface resulting from the eikonal flow at a time t is given by $\sigma(u, v, t) =$

$S(u, v) + t n(u, v), t \geq 0$. The variable t is exactly the offset distance and is also referred to as *time* to emphasize the dynamic aspect of the algorithm.

Recall from Chapter 4 for the 2D case that as the boundary curve evolves under the eikonal flow, different regions start intersecting at points, and that the trace of the intersection points is the medial axis. By extension to the 3D situation, as the boundary surface evolves under the eikonal flow in the increasing t direction, different regions start intersecting with each other creating (self-) intersection curves that grow and interact with each other until they collapse to single points and die. The approach presented in this dissertation exactly models this behavior to compute the medial axis. The medial axis consists of only the first intersection points of evolving offsets of any two given points on the surface.

During the course of the eikonal flow, special changes to the structure of intersection curves occur at certain *transition points* (including A_1A_3 and A_1^4 points), where intersection curves are created, interact with each other to undergo intermediate transitions or get annihilated. Away from transition points, the intersection curves evolve smoothly to sweep out A_1^2 surfaces of the medial axis. An intersection curve consists of a connected set of A_1^2 points, and A_3 , A_1^3 , A_1A_3 or A_1^4 points at curve ends where necessary, all sharing the same offset distance.

Algorithm 4 summarizes the steps involved in computing the medial axis. The proposed approach first computes transition points as well as A_3 and A_1^3 curves using properties of the B-spline representation. A_1^2 surfaces are then computed by evolving intersection curves over time using theoretically derived evolution vector fields. Connectivity between intersection curves at consecutive time instants is maintained as they evolve. The topology of intersection curves are appropriately modified during transition events. This computational approach is equivalent to following the level sets of the distance field of $S(u, v)$. Since the set of all local normal forms of the distance field given in [89] are considered, our approach presents the complete topological structure of the medial axis.

Algorithm 4 Computing medial axis

1. Compute transition points.
 2. Compute A_3 curves and A_1A_3 points.
 3. Compute A_1^3 curves.
 4. Classify transition points as creation, intermediate or annihilation.
 5. Compute medial axis surfaces.
-

6.2 Transition Points and Transition Events

This section presents techniques to compute all types of transition points of intersection curves for the 3D medial axis. A summary of all types of transition events is also presented. In addition to A_1A_3 and A_1^4 points, the transition points include those points on A_3 curves, A_1^3 curves and A_1^2 surfaces where the distance to the boundary attains a critical value. Transitions on A_3 and A_1^3 curves depend on the direction of increasing distance to boundary along the curves, which we term as the curve's *flow direction*.

Let $S_i(u_i, v_i)$, $i = 1, 2, 3, 4$, denote representations of different regions of the boundary surface, and let $N_i = \frac{\partial S_i}{\partial u_i} \times \frac{\partial S_i}{\partial v_i}$ denote the unnormalized normals of S_i respectively. By treating the single boundary surface as if it were different regions, we present geometric equations for computing transition points. Roots of nonlinear geometric equations are computed using adaptations of robust subdivision based techniques [45, 43]. Extensions of these methods to improve efficiency of computing A_1^3 critical points and A_1^4 points are presented.

6.2.1 A_3 Critical Points

Although every point on $S(u, v)$ has two principal curvatures $\kappa_1 \geq \kappa_2$, for the purposes of medial axis computation, it is necessary to consider only the larger principal curvature κ_1 [9]. The transition points of the medial axis related to κ_1 correspond to points on $S(u, v)$ where κ_1 attains a critical value. These points are computed by solving for simultaneous roots of Equation 6.1.

$$\kappa_{1u}(u, v) = 0 \quad \kappa_{1v}(u, v) = 0 \quad (6.1)$$

Given Equation 6.1 is solved for a particular pair (u, v) , for those values, the sphere centered at the offset point $S(u, v) + \frac{1}{\kappa_1(u, v)}n(u, v)$ having radius $\frac{1}{\kappa_1(u, v)}$ must be maximal, by definition of the medial axis. The procedure for the maximal condition check is presented in Section 6.2.6. The transition points of A_3 type are the centers of such maximal spheres. An A_3 critical point is either a creation point or an intermediate transition point, depending on whether the two A_3 curve segments on either side of the critical point flow outward or inward.

6.2.2 A_1^2 Critical Points

The A_1^2 transition points correspond to A_1^2 points for which the distance to the boundary attains a critical value. An A_1^2 critical point at which the distance function has a local minimum corresponds to a creation event. When the distance function has a local saddle, the critical point corresponds to an intermediate transition event. Finally, the critical point corresponds to an annihilation event when the distance function attains a local maximum. This characterization follows the behavior of the transition points for the intersection of two surfaces under generalized offset flows [27].

$$\mathcal{D}(u_1, v_1, u_2, v_2) = \|S_1 - S_2\|^2 = \langle S_1 - S_2, S_1 - S_2 \rangle \quad (6.2)$$

Taking the partial derivatives of \mathcal{D} with respect to u_1, v_1, u_2, v_2 and solving Equation 6.3, we obtain critical points of \mathcal{D} .

$$\langle S_1 - S_2, \frac{\partial S_i}{\partial u_i} \rangle = 0, \quad \langle S_1 - S_2, \frac{\partial S_i}{\partial v_i} \rangle = 0, \quad i = 1, 2 \quad (6.3)$$

Since some solutions of Equation 6.3 may not correspond to maximal spheres, the maximality condition must be checked at all solutions of Equation 6.3 using the procedure in Section 6.2.6. Trivial solutions at which $(u_1, v_1) = (u_2, v_2)$ are ignored. Equation 6.3 implies that the surface normals at $S_1(u_1, v_1)$ and $S_2(u_2, v_2)$ must point exactly in opposite directions as illustrated in Figure 6.1(a).

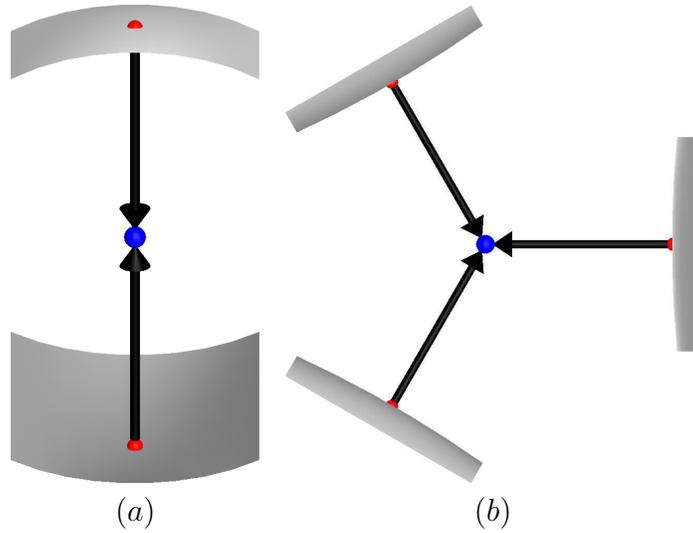


Figure 6.1. Critical point of type (a) A_1^2 , (b) A_1^3 , shown in blue. Different regions of the surface S are shown in gray and arrows point in the corresponding surface normal directions.

6.2.3 A_1A_3 Points

Figure 1.4 (d) shows an illustration of an A_1A_3 point. A_1A_3 points are computed as part of the algorithm for computing A_3 curves. See Section 6.3.1. A_1A_3 points correspond to intermediate transition events based on whether the A_3 curve and the A_1^3 curve flow inward or outward at the A_1A_3 point.

Techniques for computing the flow directions (called the shock structure) presented in [51] are summarized here. Let p and $q \in \mathbb{R}^3$ be the A_3 and A_1 type contact points, respectively, on S of the maximal sphere at the A_1A_3 point. Let $T_{i,p} \in \mathbb{R}^3$, be the unit length principal directions corresponding to $\kappa_{i,p}$, $i = 1, 2$, at p , and let n_p be the unit length surface normal at p . Consider a local coordinate system at p with $\{T_{1,p}, T_{2,p}, n_p\}$ as axes (Monge form). Let (x_q, y_q, z_q) be the coordinates of q in this coordinate system. Let $c_{ridge} = \kappa_{1,p} \langle \nabla \kappa_{1,p}, t_{2,p} \rangle ((\kappa_{1,p} - \kappa_{2,p})y_q - z_q \langle \nabla \kappa_{1,p}, t_{2,p} \rangle)$. The A_3 curve flows outward at the A_1A_3 point if $c_{ridge} > 0$ and vice versa [51]. Let $c_{junc} = \kappa_{1,p} y_q ((\kappa_{1,p} - \kappa_{2,p})y_q - z_q \langle \nabla \kappa_{1,p}, t_{2,p} \rangle)$. The A_1^3 curve flows outward at the A_1A_3 point if $c_{junc} > 0$ and vice versa [51].

6.2.4 A_1^3 Critical Points

The A_1^3 critical points correspond to A_1^3 points at which the distance to the boundary attains a critical value. At such points, the normals at the three surface points are coplanar as illustrated in Figure 6.1(b) [81]. A_1^3 critical points can be creation, annihilation or intermediate transition types based on the flow direction of the A_1^3 curve segments at the critical point. For a point $P \in \mathbb{R}^3$ to be an A_1^3 critical point, the following equations must be satisfied.

$$\langle P - S_i, \frac{\partial S_i}{\partial u_i} \rangle = 0, \quad \langle P - S_i, \frac{\partial S_i}{\partial v_i} \rangle = 0, \quad i = 1, 2, 3 \quad (6.4a)$$

$$\| P - S_1 \| = \| P - S_j \|, \quad j = 2, 3 \quad (6.4b)$$

$$\langle N_1 \times N_2, N_3 \rangle = 0 \quad (6.4c)$$

Since $P = (x, y, z)$ is unknown, Equations 6.4(a)-(c) form a system of nine nonlinear equations in nine unknowns. This system is reduced by adapting the simplification techniques presented for surface-surface bisectors in [45]. Suppose $P = S_1 + \alpha N_1$, and substitute for P in Equation 6.4(b) with $j = 2$ to obtain

$$\alpha = \frac{-\langle S_1 - S_2, S_1 - S_2 \rangle}{2\langle S_1 - S_2, N_1 \rangle} \quad (6.5)$$

Substituting for P and α in Equations 6.4(a)-(b) with $i = 2, 3$, $j = 3$, and simplifying yields a system of six equations in six variables. Denote $S_1 - S_i$ by $S_{1\mathcal{M}i}$, $i = 2, 3$.

$$2\langle S_{1\mathcal{M}2}, N_1 \rangle \langle S_{1\mathcal{M}i}, \frac{\partial S_i}{\partial u_i} \rangle - \| S_{1\mathcal{M}2} \|^2 \langle N_1, \frac{\partial S_i}{\partial u_i} \rangle = 0, \quad i = 2, 3 \quad (6.6a)$$

$$2\langle S_{1\mathcal{M}2}, N_1 \rangle \langle S_{1\mathcal{M}i}, \frac{\partial S_i}{\partial v_i} \rangle - \| S_{1\mathcal{M}2} \|^2 \langle N_1, \frac{\partial S_i}{\partial v_i} \rangle = 0, \quad i = 2, 3 \quad (6.6b)$$

$$\langle S_{1\mathcal{M}2}, N_1 \rangle \| S_{1\mathcal{M}3} \|^2 - \| S_{1\mathcal{M}2} \|^2 \langle N_1, S_{1\mathcal{M}3} \rangle = 0 \quad (6.6c)$$

$$\langle N_1 \times N_2, N_3 \rangle = 0 \quad (6.6d)$$

Note that Equations 6.4(a) for $i = 1$ are automatically satisfied. It is possible to solve the system of equations in 6.6(a)-(d) by symbolically representing the left hand

sides of the equations and using subdivision based methods directly on the 6-variate functions [45], but the large memory requirements rendered it infeasible on an 8GB RAM machine. The expression tree based method presented in [43] significantly reduced the memory requirements but was still computationally infeasible due to the large number of dimensions.

To make this approach practical, we have improved the computation time for this problem using hierarchical and parallel computation strategies. Equations 6.6(a)-(b) for $i = 2$ are dependent only upon u_1, v_1, u_2, v_2 . Therefore, the problem size is reduced by first finding subdomains that satisfy these two equations, and then finding subdomains that also satisfy Equations 6.6(a)-(d) for $i = 3$. Intuitively, Equations 6.6(a)-(b) $i = 2$ require two points on the surface that are equidistant from an offset point P in \mathbb{R}^3 . The remaining equations find a subset of those points for which there is a third surface point that is also equidistant to P .

The expression tree based approach is used to subdivide u_1, v_1, u_2, v_2 parametric domains until a user specified threshold is reached while using interval arithmetic to reject subdomains that do not satisfy Equations 6.6(a)-(b) with $i = 2$. Also, subdomains that do not correspond to regions that satisfy the maximal condition are pruned. The maximal condition is checked by computing ranges of $\rho = \alpha \| N \|$ with interval arithmetic and testing whether 1) the range contains a part of the positive real line, and 2) is bounded by the size of the region enclosed within S . Negative values of ρ correspond to points outside the region since the offset would be in the outward normal direction and thus cannot contribute to the solution. Each of the remaining subdomains along with variables u_3, v_3 is then tested with Equations 6.6(a)-(d) using subdivision along u_3, v_3 parametric directions, interval arithmetic and pruning. Moreover, this step is performed in parallel since the subdomains are independent. Trivial solutions are ignored. The centers of remaining subdomains of $u_i, v_i, i = 1, 2, 3$ are used in a Newton-Raphson refinement step [45] to obtain accurate solutions and nonmaximal solutions are ignored. This step is also performed using parallel processing techniques.

6.2.5 A_1^4 Points

A_1^4 points are equidistant to four different points on S and the corresponding sphere is maximal. Figure 1.4 (e) shows an illustration of an A_1^4 point. A_1^4 points can be annihilation or intermediate transition based on the direction of the flow of the four incident A_1^3 curves as presented in [51]. For a point $P \in \mathbb{R}^3$ to be an A_1^4 point, the following equations must be satisfied.

$$\langle P - S_i, \frac{\partial S_i}{\partial u_i} \rangle = 0, \quad \langle P - S_i, \frac{\partial S_i}{\partial v_i} \rangle = 0, \quad i = 1, 2, 3, 4 \quad (6.7a)$$

$$\| P - S_1 \| = \| P - S_j \|, \quad j = 2, 3, 4 \quad (6.7b)$$

Just as in the case for A_1^3 critical points, the system in Equation 6.7 of eleven equations is reduced using Equation 6.5 to a system in eight equations in eight variables $u_i, v_i, i = 1, 2, 3, 4$ shown in Equation 6.8(a)-(c).

$$2\langle S_{1M2}, N_1 \rangle \langle S_{1Mi}, \frac{\partial S_i}{\partial u_i} \rangle - \| S_{1M2} \|^2 \langle N_1, \frac{\partial S_i}{\partial u_i} \rangle = 0, \quad i = 2, 3, 4 \quad (6.8a)$$

$$2\langle S_{1M2}, N_1 \rangle \langle S_{1Mi}, \frac{\partial S_i}{\partial v_i} \rangle - \| S_{1M2} \|^2 \langle N_1, \frac{\partial S_i}{\partial v_i} \rangle = 0, \quad i = 2, 3, 4 \quad (6.8b)$$

$$\| S_{1Mj} \|^2 \langle S_{1M2}, N_1 \rangle - \| S_{1M2} \|^2 \langle N_1, S_{1Mj} \rangle = 0, \quad j = 3, 4 \quad (6.8c)$$

Intuitively, Equations 6.8(a)-(b) with $i = 2$ require two points on the surface that are equidistant from an offset point P in \mathbb{R}^3 . Equations 6.8(a)-(c) with $i = 3, j = 3$ find a third surface point that is also equidistant to P with the other two points. And finally, Equations 6.8(a)-(c) with $i = 4, j = 4$ find a fourth surface point that is equidistant to P with the other three points.

Equations 6.8(a)-(b) with $i = 2$ are dependent only upon u_1, v_1, u_2, v_2 ; Equations 6.8(a)-(c) with $i = 3, j = 3$ are dependent on $u_1, v_1, u_2, v_2, u_3, v_3$, and Equations 6.8(a)-(c) with $i = 4, j = 4$ are dependent on $u_1, v_1, u_2, v_2, u_4, v_4$. So this structure lends itself to a hierarchical technique similar to the A_1^3 critical point case. Equations 6.8(a)-(c) will provide exactly the same solutions for both $i = 3, j = 3$ and $i = 4, j = 4$, so this step is performed once and subdomains in the u_3, v_3 parametric directions are used for u_4, v_4 parametric directions, thereby effectively reducing the number of dimensions in

the subdivision search stage from eight to six. Moreover, Equations 6.8(a)-(c) with $i = 2, 3, j = 3$ are identical to Equations 6.6(a)-(c). So the subdivision stage for A_1^3 critical point and A_1^4 point computations are combined. The refinement stages for Equations 6.8(a)-(c) are performed in parallel.

In order to determine the flow directions of the the four A_1^3 curves at an A_1^4 point, we use the following rule as presented in [51]. Let S_1, S_2, S_3, S_4 denote the four contact points on the surface for the A_1^4 point. Let π_i be the plane through three of the four contact points $S_j, S_k, S_l, j, k, l \in \{1, 2, 3, 4\}, j \neq k \neq l, j \neq i, k \neq i, l \neq i$. If S_i and the A_1^4 point are on the same side of π_i , then the corresponding A_1^3 curve flows into the A_1^4 point. Otherwise, the A_1^3 curve flows outward from the A_1^4 point.

6.2.6 Maximal Condition Check

In several steps of the algorithms presented here, it is necessary to check whether a given point P corresponds to the center of a sphere with radius d that is maximally inscribed within the region bounded by \mathcal{B} . In this work, this condition is checked by first computing the closest point on $S(u, v)$ to P . Closest points are computed by first computing points on the surface where the distance to P , given by $\|S(u, v) - P\|$, attains a critical value. Such points are obtained by finding simultaneous roots of Equation 6.9 using robust B-spline equation solvers [45, 43].

$$\langle S - P, S_u \rangle = 0, \quad \langle S - P, S_v \rangle = 0 \quad (6.9)$$

All roots are then inspected to select the one with the smallest distance. For P to correspond to a maximally inscribed sphere, the smallest distance must equal d .

6.2.7 Catalog of Transition Events

At transition points, corresponding evolution curves are either created, annihilated or undergo intermediate transitions. We present a complete list of all generic transitions for the medial axis. This list presents consequences of the transitions on participating A_1^2 surfaces and has been compiled using theoretical results presented in [89], which considers local singularities of the distance field, as well as the shock structure of A_3 and A_1^3 curves presented in [51].

1. *Creation Events:*

- i) At an A_3 critical point where κ_1 has a local maximum. An intersection curve segment is created with the two end points on an A_3 curve (Figure 6.2 (a)).
- ii) At an A_1^2 critical point where the distance to S has a local minimum. An intersection curve loop of the evolving offset surfaces is created (Figure 6.2 (b)).

2. *Annihilation Events:*

- i) At an A_1^2 critical point where the distance to S has a local maximum. An intersection curve loop ends (Figure 6.3 (a)).
- ii) At an A_1^3 critical point where two A_1^3 curves flow inward, and the corresponding intersection curves disappear (Figure 6.3 (b)). The circumcenter of the triangle formed by the three contact points of the A_1^3 point lies inside the triangle.
- iii) At an A_1^4 point where four A_1^3 curves flow inward, and the corresponding intersection curves end (Figure 6.3 (c)).

3. *Intermediate Transition Events:*

- i) At an A_3 critical point where κ_1 has a local saddle point, at which two intersection curves join (Figure 6.4).

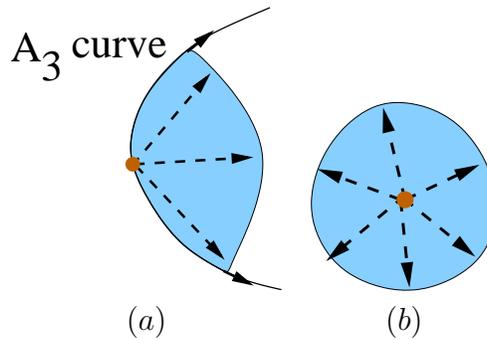


Figure 6.2. Creation events at (a) A_3 type critical point, (b) A_1^2 type critical point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in blue.

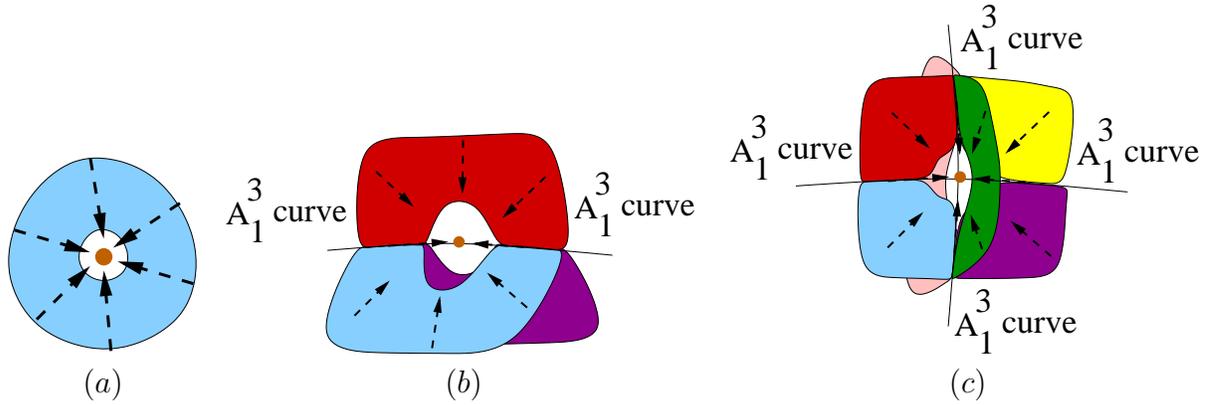


Figure 6.3. Annihilation events at (a) A_1^2 type critical point, (b) A_1^3 type transition point and (c) A_1^4 type transition point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.

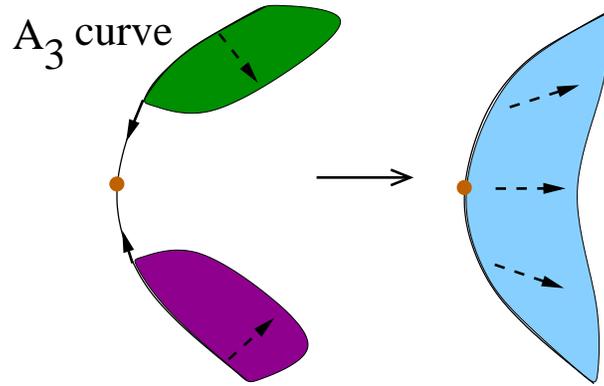


Figure 6.4. Intermediate transition event at an A_3 type critical point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in blue.

- ii) At an A_1^2 critical point where the distance to S has a local saddle so two intersection curves meet and exchange branches (Figure 6.5).
- iii) At an A_1A_3 point where the intersection curve from a smooth sheet passes and creates an A_1^3 point which evolves along an A_1^3 curve away from the A_1A_3 point (Figures 6.6, 6.7). Figure 6.6 corresponds to the case when the flow direction of the A_3 curve is into the A_1A_3 point and the flow direction of the A_1^3 curve is directed away from the A_1A_3 point. Figure 6.7 corresponds to the case when the flow directions of the A_3 and A_1^3 curves are directed away from the A_1A_3 point.

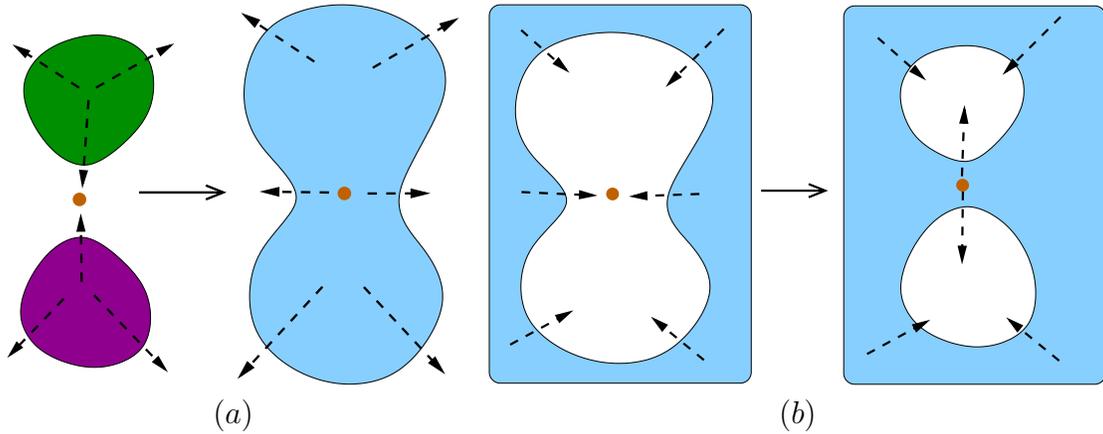


Figure 6.5. Intermediate transition events at an A_1^2 type critical point. The evolving intersection curves meet and exchange branches. This can result in a merge event (shown in (a)) or a split event (shown in (b)). Transition points are shown in brown and the evolving A_1^2 surfaces are shown in blue.

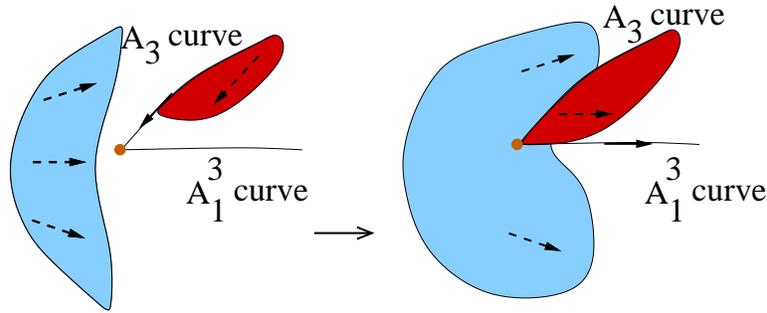


Figure 6.6. Intermediate transition event at an A_1A_3 type critical point. (a) One evolving intersection curve splits at the A_1A_3 point to create two segments belonging to the same A_1^2 surface. One of the ends of another intersection switches from an A_3 curve onto an outward flowing A_1^3 curve. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.

- iv) At an A_1A_3 point where an A_1^3 curve flows into it and the intersection curves meeting at the transition point transform into a smooth curve on a smooth medial sheet (Figure 6.8). This transition corresponds to the situation when the flow directions of the A_3 and A_1^3 curves are directed into the A_1A_3 point.
- v) At an A_1^3 critical point where three intersection curves on different A_1^2 surfaces meet and split into two triples of intersection curve segments which meet at the end points on A_1^3 curves flowing outward (Figure 6.9). The circumcenter

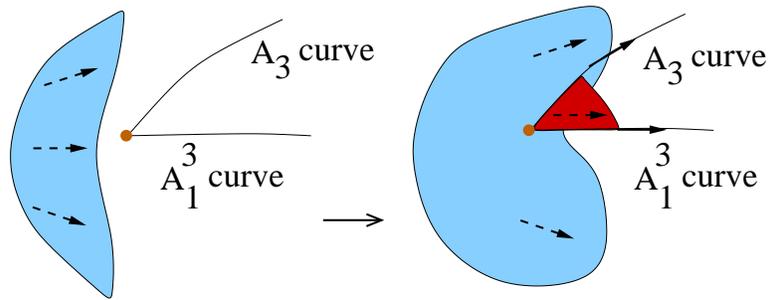


Figure 6.7. Intermediate transition event at an A_1A_3 type critical point. One evolving intersection curve splits at the A_1A_3 point to create two segments belonging to the same A_1^2 surface. A new A_1^2 surface is created with one end of the intersection curve on an outward flowing A_3 curve and the other end on an outward flowing A_1^3 curve. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.

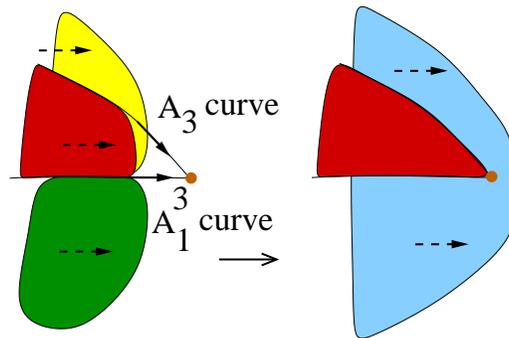


Figure 6.8. Intermediate transition event at an A_1A_3 type critical point. An evolving intersection curve with one end on an inward flowing A_3 curve and the other end on an inward flowing A_1^3 curve ends at the A_1A_3 point. Two other evolving intersection curve segments merge and the A_1^2 surface evolves smoothly beyond the A_1A_3 point. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.

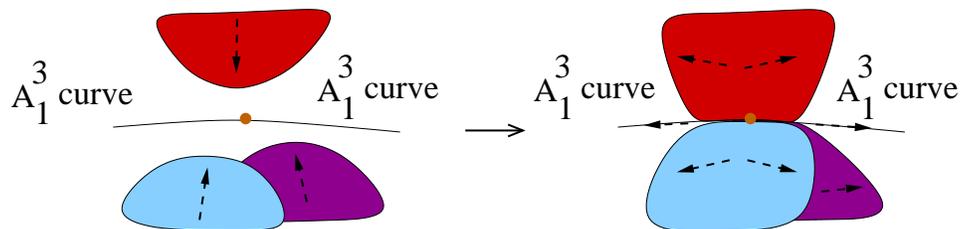


Figure 6.9. Intermediate transition event at an A_1^3 type critical point. Three evolving intersection curves meet, split and create two triples of intersection curves. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.

of the triangle formed by the three contact points of the A_1^3 point lies inside the triangle.

- vi) At an A_1^3 critical point where two intersection curves on different A_1^2 surfaces meet and split into two pairs of intersection curve segments meeting at the end points on A_1^3 curves flowing outward. Also, a new intersection curve on a third A_1^2 surface is created with the same end points on the A_1^3 curve (Figure 6.10). The circumcenter of the triangle formed by the three contact points of the A_1^3 point lies outside the triangle.
- vii) Two intersection curves on distinct A_1^2 surfaces meet at two regular A_1^3 points, where two intersection curves on a third A_1^2 surface also end. The two A_1^3 curves flow into the critical point, the first two intersection curves on different A_1^2 surfaces collapse to the critical point, and the other two intersection curves on the third A_1^2 surface merge into a single intersection curve (Figure 6.11). The circumcenter of the triangle formed by the three contact points of the A_1^3 point lies outside the triangle.
- viii) Three A_1^3 curves meet at an A_1^4 point with a transition to one A_1^3 curve flowing outward from the A_1^4 point. Intersection curves are modified accordingly (Figure 6.12).
- ix) Two A_1^3 curves meet at an A_1^4 point with a transition to the other two A_1^3 curves. Intersection curves are modified accordingly (Figure 6.13).

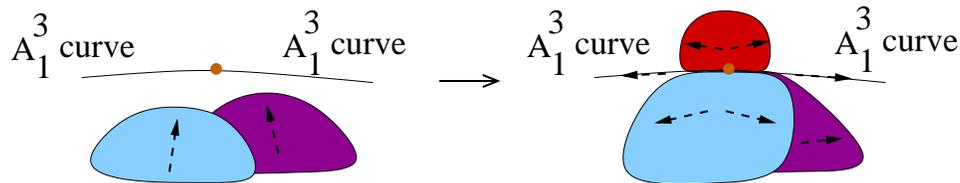


Figure 6.10. Intermediate transition event at an A_1^3 type critical point. Two evolving intersection curves meet and split. A third intersection curve on a new A_1^2 surface is created. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.

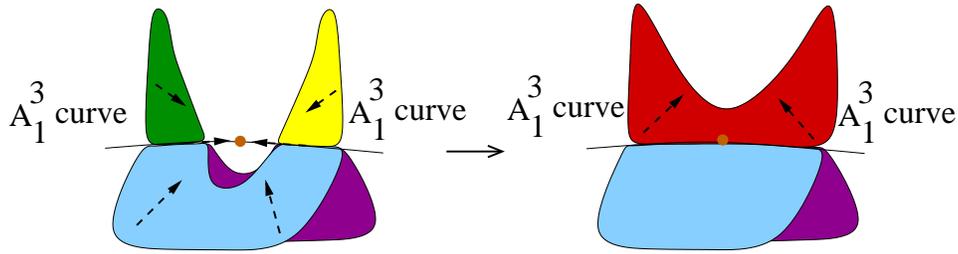


Figure 6.11. Intermediate transition event at an A_1^3 type critical point. Two evolving intersection curves annihilate. Another set of two intersection curves merge. Transition points are shown in brown and the evolving A_1^2 surfaces are shown in various colors.

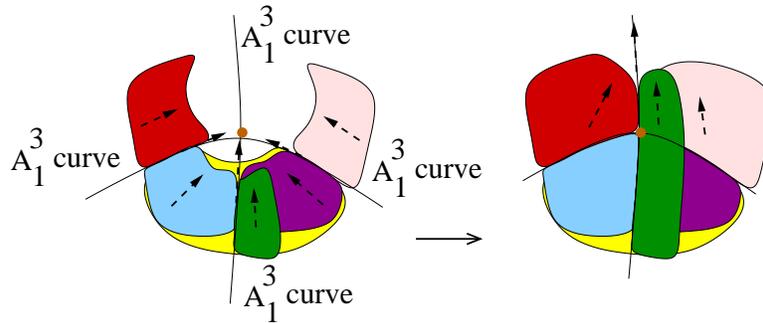


Figure 6.12. Intermediate transition event at an A_1^4 type transition point. Three evolving intersection curves annihilate. Each of three other intersection curves switch an end point from an inward flowing A_1^3 curve onto the outward flowing A_1^3 curve. Transition points are shown in brown and the A_1^2 surfaces are shown in various colors.

Proposition 6.2.1 *The list of transition events in Section 6.2.7 is complete for the generic case.*

Proof. Consider the function $D(y) = \min\{\|x - y\| : x \in S(u, v)\}$. $D(y)$ is the distance field giving the closest distance of any point $y \in \mathbb{R}^3$ to the surface S . The 3D medial axis corresponds to the singular set of $D(y)$. The set of all local normal forms of $D(y)$ for the generic case has been characterized in [89]. Computing the evolution of the self-intersection set of offsets of S is equivalent to following the level sets of $D(y)$. Therefore, the transition events of the intersection curves are given by the local normal forms of $D(y)$. Since the complete set of all local normal forms of $D(y)$ for the generic case is considered in our approach, the list of transition events is complete. ■

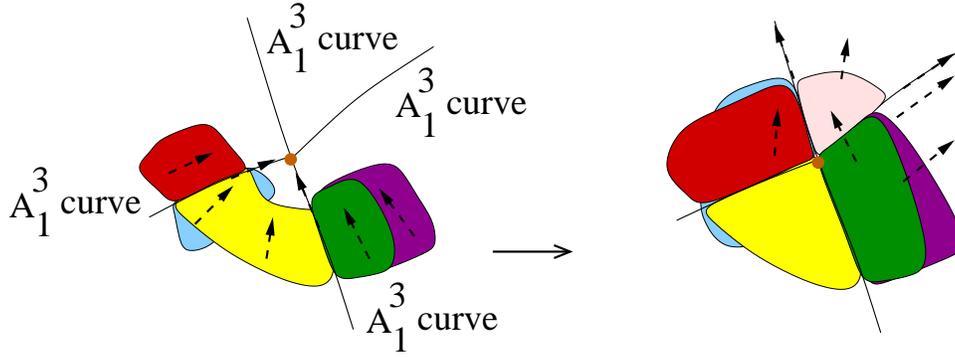


Figure 6.13. Intermediate transition event at an A_1^4 type transition point. One evolving intersection curve ends. Each of four other intersection curves switch an end point from an inward flowing A_1^3 curve onto the outward flowing A_1^3 curve. One new intersection curve with end points on the two outward flowing A_1^3 curves is created. Transition points are shown in brown and the A_1^2 surfaces are shown in various colors.

6.3 Curve Elements

6.3.1 A_3 Curves

Ridges are loci of points on a surface at which one of the principal curvatures attains a critical value along its principal direction [59]. A ridge point of κ_1 satisfies $\langle \nabla \kappa_1, t_1 \rangle = 0$, where principal direction t_1 is a 2D column vector whose two coordinates denote coefficients of S_u and S_v , respectively, at $S(u, v)$. A κ_1 ridge point is called *elliptic* if κ_1 attains a local maximum along t_1 [59].

A_3 curves on the medial axis correspond to loci of elliptic ridge points of κ_1 on the surface such that the sphere centered at the center of curvature corresponding to κ_1 is maximally inscribed within the region of interest. Ridges on $S(u, v)$ are computed using techniques presented in Chapter 5 that guarantee robust and accurate extraction of all ridges on B-spline surfaces. Ridge curves are output as polylines. Elliptic ridges of κ_1 are identified by testing the extremum type condition at each ridge point vertex of the polylines. Only those points that pass the maximal condition are retained resulting in segments or closed curve loops (See Section 6.2.6). The end points of each nonloop segment occur where the maximal condition first fails and corresponds to an A_1A_3 point. A_3 curves on the medial axis are then computed as offsets of elliptic ridges at distances $\frac{1}{\kappa_1}$ (radius of curvature) along the inward normal of the surface at each point. In Equation 6.10, \mathcal{R} is a collection of m ridge curves \mathcal{R}_j that correspond to A_3 curves.

Each \mathcal{R}_j is a connected set of elliptic ridge points on S .

$$\begin{aligned} \mathcal{R} &= \{\bigcup_{j=1}^m \mathcal{R}_j\}, \quad \mathcal{R}_j = \{(u, v) : \langle \nabla \kappa_1(u, v), t_1(u, v) \rangle = 0, \\ t_1^T \begin{bmatrix} \kappa_{1uu} & \kappa_{1uv} \\ \kappa_{1uv} & \kappa_{1vv} \end{bmatrix} t_1 < 0, \quad b_{max}(u, v) = (C(u, v), \frac{1}{\kappa_1(u, v)})\} \\ C(u, v) &= S(u, v) + \frac{1}{\kappa_1(u, v)} n(u, v) \end{aligned} \quad (6.10)$$

where $b_{max}(u, v)$ denotes a maximally inscribed sphere with center $C(u, v)$ and radius $\frac{1}{\kappa_1(u, v)}$. The A_3 curves on the medial axis are the loci of the sphere centers $C(u, v)$ for all \mathcal{R}_j .

6.3.1.1 Identifying A_1A_3 Points on A_3 Curves

By definition, A_1A_3 points are locations where the maximal sphere is tangent to the corresponding ridge point on the surface as well as to another point in a different region of the surface. For each end point of a nonloop segment where the maximal condition fails, the other surface point corresponding to the A_1 condition is determined by finding closest points on the surface with distance equal to the radius of curvature ($\frac{1}{\kappa_1}$) at the ridge point using Equation 6.9 in Section 6.2.6.

6.3.2 A_1^3 Curves

This section presents a tracing algorithm for computing A_1^3 curves. A_1^3 critical points, A_1A_3 points and A_1^4 points, the computation of which is presented in Section 6.2, are source and sink points for the proposed algorithm. The regular points of A_1^3 curves correspond to points where three different regions of the of the deforming boundary intersect transversely (i.e., the three tangent planes of the offset surface at the intersection point are different). Using this characterization, we construct evolution vector fields based on methods and ideas from [32] to follow the time trace of A_1^3 points. When these vector fields are integrated beginning at an A_1^3 point, we obtain the A_1^3 curve from that point.

6.3.2.1 Evolution Vector Field for A_1^3 Curves

Consider a triple intersection point $P \in \mathbb{R}^3$ at which the three offset surfaces intersect transversely. Denote the three different surface regions as S_i , with unit surface normal vectors n_i , and the corresponding offset surfaces as $\sigma_i(u_i, v_i, t)$, $i = 1, 2, 3$ respectively.

Let $\phi_i = \sigma_{i_{u_i}} \times \sigma_{i_{v_i}}$ (Note $\phi_i \parallel n_i$). Since the tangent planes at P are all different, n_i , $i = 1, 2, 3$ span \mathbb{R}^3 . $Z_{ij} = \phi_i \times \phi_j$, $i = 1, 2, 3$, $j = (i + 1) \bmod 3$, are tangent vectors to the intersection curves of the pair of offset surfaces σ_i and σ_j . A linear algebra argument shows that $\{Z_{12}, Z_{23}, Z_{13}\}$ also forms a basis for \mathbb{R}^3 at P . Also, because both Z_{ij} and Z_{jk} are orthogonal to n_j , $i = 1, 2, 3$, $j = (i + 1) \bmod 3$, $k = (i + 2) \bmod 3$, they span the tangent plane of σ_j at P . (See Figure 6.14). At each point in the neighborhood of P , we write

$$\begin{aligned} n_1 - n_2 &= a_1 Z_{12} + b_1 Z_{13} + c_1 Z_{23} \\ n_3 - n_2 &= a_2 Z_{12} + b_2 Z_{13} + c_2 Z_{23} \end{aligned} \quad (6.11)$$

From Equation 6.11, the evolution vector field η in a neighborhood of P in \mathbb{R}^3 is given by Equation 6.12, which is given in a representation in terms of the normal vector field and tangent vector fields to each offset surface.

$$\begin{aligned} \eta &= n_1 - ((a_1 - a_2)Z_{12} + b_1 Z_{13}) \\ &= n_2 + (c_1 Z_{23} + a_2 Z_{12}) \\ &= n_3 - (b_2 Z_{13} + (c_2 - c_1)Z_{23}) \end{aligned} \quad (6.12)$$

Proposition 6.3.1 η is tangent to the A_1^3 curve.

Proof. For this discussion, we temporarily consider separate time variables t_i for the three different surface regions σ_i . As σ_i ($i = 1, 2$ or 3) deforms under the eikonal flow

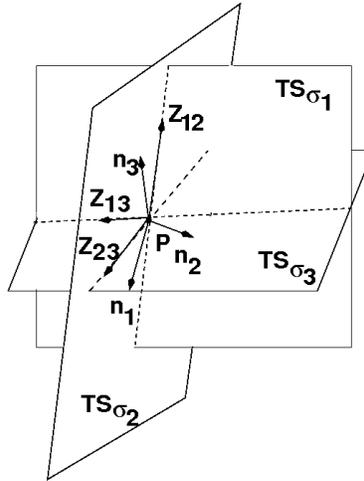


Figure 6.14. Tangent planes of offset surfaces σ_i with normals n_i , $i = 1, 2, 3$ at a point P . Pairwise intersections of offset surfaces are along Z_{12} , Z_{23} , Z_{13} .

with varying t_i , there will be a unique surface point for a given time t_i in the neighborhood of any point Q that lies within the volume swept out by the deforming surfaces. By the inverse function theorem, t_i is a differentiable function in the neighborhood of Q . Consider the values of t_i in the neighborhood of Q as a scalar field. Since each point on σ_i is deforming along the corresponding surface normal vector n_i , $\nabla t_i = n_i$. Therefore, the directional derivative of t_i along n_i , $\nabla_{n_i} t_i = \langle \nabla t_i, n_i \rangle = 1$. Further, the directional derivative of t_i along Z_{ij} and Z_{ik} are zero since they are in tangent plane of σ_i . Therefore, the directional derivative of t_i along η is $\nabla_{\eta} t_i = \langle \nabla t_i, \eta \rangle = 1$.

Let t_i be the time when σ_i reaches Q ($i = 1, 2, 3$). Define $\delta_{1j}(Q) = t_1 - t_j$, $j = 2, 3$. The directional derivative of δ_{1j} along η is $\nabla_{\eta} \delta_{1j} = \nabla_{\eta} t_1 - \nabla_{\eta} t_j = 1 - 1 = 0$. Thus, η is tangent to the level curves of δ_{12} and δ_{13} . Setting $t_1 = t_j$ gives $\delta_{1j} = 0$, $j = 2, 3$, that correspond to intersection points of the offset surfaces that are on the A_1^3 curve. Hence, η is tangent to the A_1^3 curve at an A_1^3 point. ■

The integral curve of η through P is a curve of triple intersection points and hence follows the evolution of the A_1^3 curve until a sink point is reached.

$$\frac{d\chi}{dt} = \eta(\chi), \quad \chi(0) = P, \quad \chi(t) \in \mathbb{R}^3 \quad (6.13)$$

Furthermore, define vector fields on the parameter-time space as

$$\nu_i = e_t + \alpha_i e_{u_i} + \beta_i e_{v_i}, \quad i = 1, 2, 3 \quad (6.14)$$

$$\begin{aligned} \alpha_1 &= (a_1 - a_2) \langle -\phi_2, \sigma_{1v_1} \rangle + b_1 \langle \phi_3, \sigma_{1v_1} \rangle \\ \beta_1 &= (a_1 - a_2) \langle \phi_2, \sigma_{1u_1} \rangle - b_1 \langle \phi_3, \sigma_{1u_1} \rangle \\ \alpha_2 &= -a_2 \langle \phi_1, \sigma_{2v_2} \rangle + c_1 \langle \phi_3, \sigma_{2v_2} \rangle \\ \beta_2 &= a_2 \langle \phi_1, \sigma_{2u_2} \rangle - c_1 \langle \phi_3, \sigma_{2u_2} \rangle \\ \alpha_3 &= b_2 \langle -\phi_1, \sigma_{3v_3} \rangle + (c_1 - c_2) \langle \phi_2, \sigma_{3v_3} \rangle \\ \beta_3 &= b_2 \langle \phi_1, \sigma_{3u_3} \rangle - (c_1 - c_2) \langle \phi_2, \sigma_{3u_3} \rangle \end{aligned} \quad (6.15)$$

and e_l denotes the unit vector in the parameter-time space direction l , $l = \{t, u_1, v_1, u_2, v_2, u_3, v_3\}$. Then, $\eta = d\sigma_i(\nu_i)$, $i = 1$ or 2 or 3 . This implies that the integral curves of ν_i are mapped by σ_i to integral curves of η . The corresponding integral curves of ν_i will trace the evolution of the intersection curves in the parameter space.

6.3.2.2 Tracing Algorithm for A_1^3 Curves

We first classify the critical points and end points as source and sink points. Starting points are computed from each type of source point using local geometric properties of the medial axis [51] and Algorithm 5 is used to trace A_1^3 curves from all starting points.

For each starting point, three initial points corresponding to the tangency points of an A_1^3 point are computed. The parameter values of the three initial points are then refined to correspond to a point on the A_1^3 curve using a Newton's method for Equation 6.16.

$$\begin{aligned} S_1 + rn_1 &= S_2 + rn_2 \\ S_1 + rn_1 &= S_3 + rn_3 \end{aligned} \quad (6.16)$$

where $r = r_{src} + \delta_{src}$ is the radius of the maximal sphere at the starting A_1^3 point, r_{src} is the radius of the maximal sphere at the source point and δ_{src} is a user-specified step size ($\delta_{src} = 10^{-3}$ was used in our examples). The projection also computes the parameter values of the three surface points corresponding to the starting A_1^3 point.

There are three types of source points:

1. An A_1A_3 point provides one start A_1^3 point. Let p and $q \in \mathbb{R}^3$ be the A_3 and A_1 type contact points, respectively, of the maximal sphere on S . Let $T_{1,p} \in \mathbb{R}^3$, be the unit length principal direction corresponding to $\kappa_{1,p}$ at p . $T_{1,p}$ is perpendicular to the A_1^3 curve at the A_1A_3 point [51]. Therefore, initial points $f_1 = p + \delta_{src}T_{1,p}$, $f_2 = p - \delta_{src}T_{1,p}$ are computed and projected onto S . The corresponding parameter values of the projected points and the parameter values of q are used to determine a start point.
2. Source A_1^3 critical points provide two start A_1^3 points computed on either side of n_{src} , the normal to the plane containing the three surface points. The three initial points are computed as projections of $S_i + \delta_{src}n_{src}$ for one of the start points, and as projections of $S_i - \delta_{src}n_{src}$ for the other start point.
3. A_1^4 source points provide one or two start A_1^3 points [51]. Each start point corresponds to a three point subset of the four A_1 type contact points of the A_1^4 point. Let S_i , $i = 1, 2, 3$, be such a subset on S . The vector, d_{src} , from the A_1^4 point to the circumcenter of the triangle formed by S_i , $i = 1, 2, 3$, is tangent to

the A_1^3 curve [51]. The three initial points required for a start point are computed as projections of $S_i + \delta_{src}d_{src}$.

Algorithm 5 Tracing A_1^3 curves

```

INPUT SRC, SINK
OUTPUT  $C_{all}$ , the set of all  $A_1^3$  curves
 $C_{all} := \emptyset$ 
for  $p := (u_1, v_1, u_2, v_2, u_3, v_3, t) \in \text{SRC}$  do
   $C := \{p\}$ 
  while  $p \notin \text{SINK}$  do
     $u_i := u_i + \alpha_i dt; v_i := v_i + \beta_i dt, i = 1, 2, 3$ 
     $p := \text{refine}(u_1, v_1, u_2, v_2, u_3, v_3, t + dt)$ 
     $C := C \cup \{p\}$ 
  end while
   $C_{all} := C_{all} \cup C$ 
end for

```

In Algorithm 5, α_i, β_i are computed by first solving for $a_j, b_j, c_j, j = 1, 2$ from Equation 6.11 and substituting in Equation 6.15. In order to avoid numerical errors accumulating over time, a Newton's refinement step for Equation 6.16 with $r = t + dt$ is performed to project points accurately onto A_1^3 curves [45].

6.3.2.3 Alternate Approach for Computing A_1^3 Curves

We also present an alternate formulation of a vector that is tangent to A_1^3 curves away from transition points. This tangent vector is defined in euclidean (world) space and can be used in place of the evolution vector field to trace A_1^3 curve segments.

Suppose $S_i(u_i, v_i), i = 1, 2, 3$ are the three surface points corresponding to the triple intersection point P . Let O be the circumcenter of the triangle formed with S_1, S_2, S_3 as vertices. Then the vector $T = \vec{OP}$ is tangent to the A_1^3 curve through P [51]. By computing a new tangent vector, T , at every step, A_1^3 curve segments can then be traced by solving the differential equation

$$\frac{dx}{dt} = T(x), \quad x(0) = P, \quad x(t) \in \mathbb{R}^3 \quad (6.17)$$

Algorithm 6 is used to trace all A_1^3 curves from source to sink points. Transition points are first computed as presented in Section 6.2 and classified as source (creation)

and sink (annihilation) points. Numerical improvement is performed at each step in order to avoid errors from numerical integration using a refinement step for Equation 6.16 [45]. This numerical improvement step also computes the parameter values of the three surface points corresponding to the new A_1^3 point.

Algorithm 6 Tracing A_1^3 curves

INPUT SRC, SINK
 OUTPUT m_{all} , the set of all A_1^3 curves
 $m_{all} := \emptyset$
for $p := (x, y, z) \in \text{SRC}$ **do**
 $m := \{p\}$
 $q := p$
 while $q \notin \text{SINK}$ **do**
 Integrate Equation 6.17 to obtain new point q
 $m := m \cup \{q\}$
 end while
 $m_{all} := m_{all} \cup m$
end for

6.4 Medial Surfaces

This section presents evolution vector fields to sweep out intersection curves of offset surfaces under the eikonal flow. An algorithm to compute the surfaces of the medial axis using the evolution method is then presented. Between transition points, the evolution vector fields are integrated to compute a time trace of the evolving intersection curves that, together with transition points, forms the medial axis A_1^2 surfaces. The construction of the evolution vector fields will follow the same ideas used in the Section 4.3 for the 2D medial axis case. It is an adaptation of that given in [27] for the evolution of intersection curves of two different surfaces under generalized offset flows. In our case, it is applied to the self-intersections of the offsets of a single surface under the eikonal flow.

6.4.1 Evolution Vector Field for Intersection Curves

Consider two separate surface regions S_i with unit normals n_i and the offset surfaces under the eikonal flow by $\sigma_i(u_i, v_i, t)$, $i = 1, 2$, respectively, at a point P on

an intersection curve. If P is not a transition point, then σ_1 and σ_2 are not tangent and hence, n_1 and n_2 are independent. $n_1 \times n_2$ is tangent to the intersection curve. $W = n_2 \times (n_1 \times n_2)$ is in the tangent plane of σ_2 . W is also orthogonal to the intersection curve (Figure 6.15). Since n_1 and n_2 are independent, $\{\sigma_{1u_1}, \sigma_{1v_1}, W\}$ are independent vector fields, and hence a basis for \mathbb{R}^3 . Thus, in the neighborhood of P , we write

$$n_2 - n_1 = a_1 \sigma_{1u_1} + b_1 \sigma_{1v_1} - W \quad (6.18)$$

Since W lies in the tangent plane of σ_2 ,

$$W = a_2 \sigma_{2u_2} + b_2 \sigma_{2v_2} \quad (6.19)$$

Combining Equations 6.18 and 6.19, we define an evolution vector field, ξ , in the neighborhood of P in \mathbb{R}^3 given by two equivalent representations

$$\xi = n_1 + a_1 \sigma_{1u_1} + b_1 \sigma_{1v_1} = n_2 + a_2 \sigma_{2u_2} + b_2 \sigma_{2v_2} \quad (6.20)$$

Proposition 6.4.1 ξ is tangent to the A_1^2 medial surface formed from the union of the evolving intersection curves of σ_1 and σ_2 .

Proof. For this discussion, we temporarily consider separate time variables t_i for the two different surface regions σ_i . As σ_i ($i = 1$ or 2) deforms under the eikonal flow with

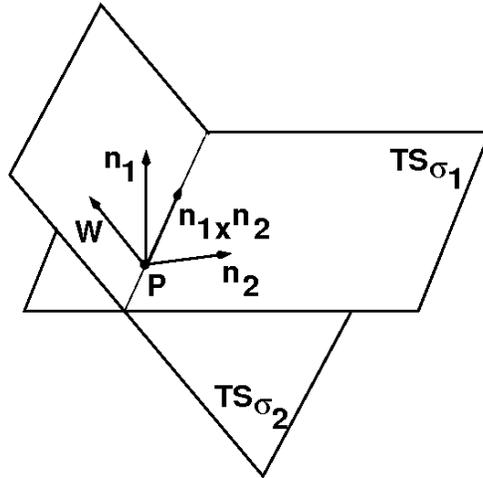


Figure 6.15. Tangent planes of offset surfaces σ_i with normals n_i , $i = 1, 2$, intersecting along $n_1 \times n_2$ at a point P . $W \in TS_{\sigma_2}$.

varying t_i , there will be a unique surface point for a given time t_i in the neighborhood of any point Q that lies within the region covered by the deforming surfaces. By the inverse function theorem, t_i is a differentiable function in the neighborhood of Q . Consider the values of t_i in the neighborhood of Q as a scalar field. Since each point on σ_i is deforming along the corresponding surface normal vector n_i , $\nabla t_i = n_i$. Therefore, the directional derivative of t_i along n_i , $\nabla_{n_i} t_i = \langle \nabla t_i, n_i \rangle = 1$. Further, the directional derivative of t_i along $\sigma_{i_{u_i}}$ and $\sigma_{i_{v_i}}$ are zero since they are in tangent plane of σ_i . Therefore, the directional derivative of t_i along ξ is $\nabla_{\xi} t_i = \langle \nabla t_i, \xi \rangle = \langle \nabla t_i, n_i + a_i \sigma_{i_{u_i}} + b_i \sigma_{i_{v_i}} \rangle = 1$.

Define $\phi(Q) = t_1 - t_2$, where t_i is the time when σ_i reaches Q ($i = 1, 2$). The directional derivative of ϕ along ξ is $\nabla_{\xi} \phi = \nabla_{\xi} t_1 - \nabla_{\xi} t_2 = 1 - 1 = 0$. Thus, ξ is tangent to the level curves of ϕ . Setting $t_1 = t_2$ gives $\phi = 0$, that correspond to intersection points of the offset surfaces that are on the medial axis. Hence, ξ is tangent to the A_1^2 surface at an A_1^2 point on an evolving intersection curve. ■

Thus, we can follow the evolution of the intersection curve (where the evolving curves remain transverse) by integrating the vector field ξ with initial conditions as the points on the intersection curve.

$$\frac{d\chi}{dt} = \xi(\chi), \quad \chi(0) = P, \quad \chi(t) \in \mathbb{R}^3 \quad (6.21)$$

Furthermore, define vector fields on the parameter-time space as $\zeta_i = e_t + a_i e_{u_i} + b_i e_{v_i}$, where e_l denotes the unit vector in the parameter-time space direction l , $l = t, u_1, v_1, u_2, v_2$. Then, $\xi = d\sigma_1(\zeta_1) = d\sigma_2(\zeta_2)$. This implies that the integral curves of the ζ_i are mapped by σ_i to integral curves of ξ . The corresponding integral curves of ζ_i will trace the evolution of the intersections curves in the parameter space.

6.4.2 Algorithm for Computing A_1^2 Surfaces

Given a connected set of samples representing an intersection curve at time t , a discrete marching algorithm is used to trace each point onto a new intersection curve at time $t + dt$ using Equation 6.21. Suppose, $P = \sigma_i(u_i^p, v_i^p, t)$ on an intersection curve evolves to a point $Q = \sigma_i(u_i^q, v_i^q, t + dt)$ after a small time dt , then $(u_i^q, v_i^q) = (u_i^p + a_i dt, v_i^p + b_i dt)$. a_i and b_i can be obtained by solving Equations 6.18 and 6.19. In order to avoid numerical errors accumulating over time, the middle point algorithm [8]

is used to project points accurately onto intersection curves and refine parameter values simultaneously. Points on an intersection curve are resampled at every time step by adaptively inserting or removing points such that they are approximately uniformly spaced in \mathbb{R}^3 .

Starting from $t = 0$, Algorithm 7 is used to compute A_1^2 surfaces. Intersection curves can have A_3 or A_1^3 points at curve ends after certain transitions. Since A_3 and A_1^3 curves are computed in a prior step, the evolution of such end points is performed by tracking points on corresponding A_3 and A_1^3 curves. A_1A_3 and A_1^4 points occur at transition events only. Such points are added to intersection curve ends during the transition to maintain topological structure of the medial axis. Every medial axis point is associated with a set of parameter values, one for each contact point on S . Each parameter designates a distinct region of S , where the intersection of the offsets at time given by the radius of the maximal sphere results in the medial axis point. During transitions, the correspondences between parameter values of the end points of two interacting intersection curves are obtained using distance in parameter space to consistently identify distinct regions of S . For example, let p and q be the end A_1^2 points of two interacting intersection curves. Let the parameter values of the two tangency points for p and q be $(u_i^{(p)}, v_i^{(p)})$, $i = 1, 2$ and $(u_j^{(q)}, v_j^{(q)})$, $j = 1, 2$ respectively. $(u_i^{(p)}, v_i^{(p)})$ and $(u_j^{(q)}, v_j^{(q)})$ are matched based on the proximity in parameter space.

Algorithm 7 Computing surface sheets

1. Sort transition points in order of increasing time.
 2. Increment t by small timestep dt .
 3. If no transition points are encountered, evolve all current intersection curves to $t + dt$.
 4. Otherwise, perform transition for each transition point encountered in increasing order of t and evolve intersection curves not involved in the transition event.
 5. If all transitions have been completed, then stop. Otherwise, repeat Step (ii).
-

6.5 Results and Discussion

This section presents examples of 3D medial axis computation using the proposed approach. In each example, the region of interest is bounded by a single tensor product parametric biquintic B-spline surface. As explained in Section 6.1, the medial axis singular set and transition points are computed prior to computing surface sheets of a medial axis. Figures 6.16, 6.17 and 6.18 show several steps of the evolution of the (self-)intersecting set of the eikonal flow for various shapes. In these figures, A_1A_3 points are indicated by purple spheres, A_1^4 points by dark blue spheres, A_3 curves by thick blue curves, A_1^3 curves by thick yellow curves and A_1^2 surface sheets in dull violet. Transition points are shown as green spheres and evolving intersection curve fronts are shown as thick red curves.

Figure 6.16 shows several steps of the A_1^2 surface computation for a deformed ellipsoid shape. The medial axis singular set for this example is a single A_3 curve. The evolution starts at the left and right extremities of the shape that correspond to curvature maximum creation points. The intersection curves merge into a single loop at two simultaneous curvature saddle points that then annihilate at a distance maximum point. The medial axis therefore consists of a single A_1^2 surface bounded by a closed A_3 curve.

Figure 6.17 shows several steps of the A_1^2 surface computation for a deformed ellipsoid shape with a fin. The medial axis singular set for this example consists of a closed A_3 curve, another A_3 curve segment with an A_1A_3 point at both ends and an A_1^3 curve

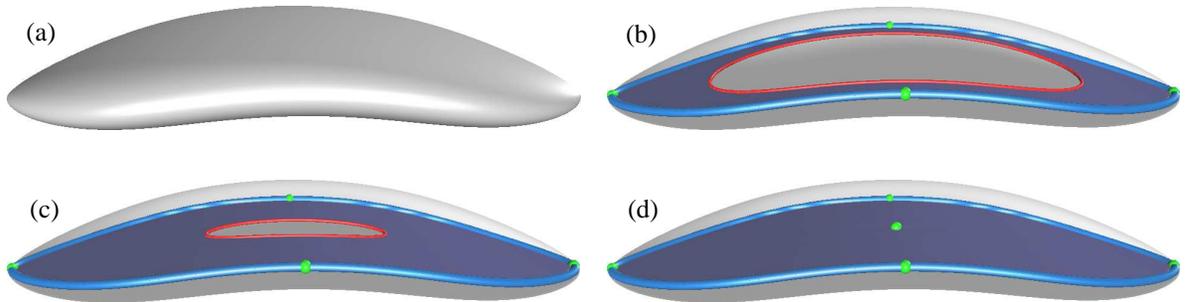


Figure 6.16. Several stages of evolution of medial axis of a deformed ellipsoid. Transition points are shown in green, evolving intersection curves are shown in red.

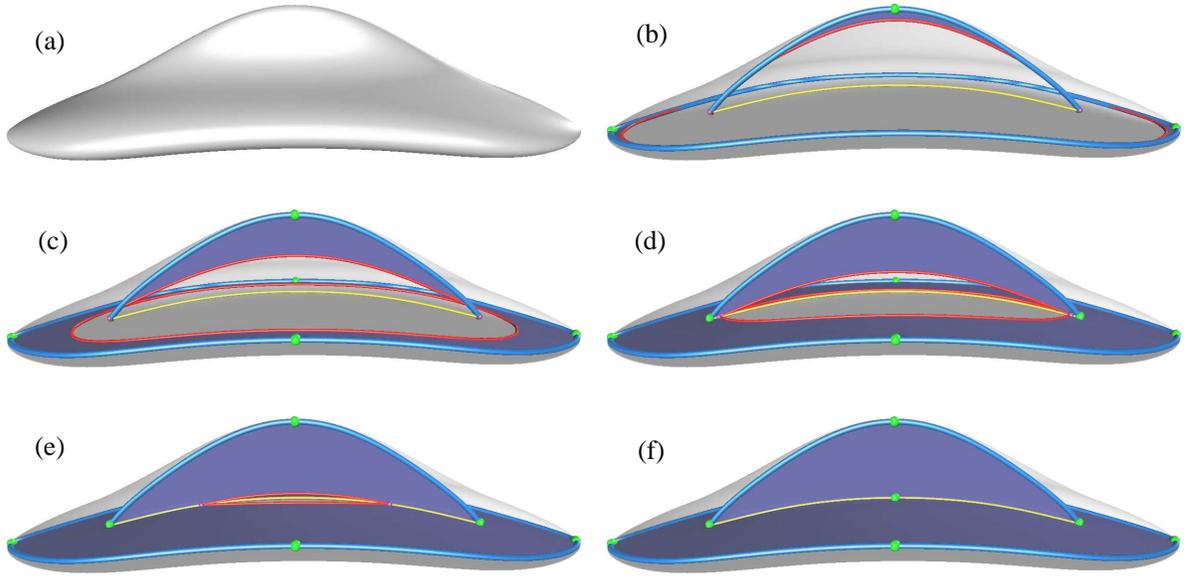


Figure 6.17. Several stages of evolution of medial axis of a deformed ellipsoid with a fin. Transition points are shown in green, evolving intersection curves are shown in red.

connecting the two A_1A_3 points. Intersection curves are created at curvature maximum points at the left and right extremities of the shape, as well as at the top of the model corresponding to the fin. The intersection curves corresponding to the bottom surface merge into a single loop at two simultaneous curvature saddle points. The intersection curves then transition at the two A_1A_3 points and annihilate at a critical point on the A_1^3 curve.

Figure 6.18 shows several steps of the A_1^2 surface computation for a more complicated shape. Intersection curves are created at curvature maximum points (Figure 6.18(b)). An A_3 curve loop is formed after corresponding intersection curve segments merge at curvature saddle points (right most A_3 curve shown in Figure 6.18(b)). The intersection curves undergo further transitions at fin points (Figure 6.18(c) and (d)) and then evolve through the A_1^4 point where three A_1^3 curves flow into the A_1^4 point and a fourth one flows outward (Figure 6.18(e)). The intersection curves finally annihilate at a critical point on the A_1^3 curve (Figure 6.18(f)). The medial axis for this shape exhibits all the generic structural elements presented in Section 1.2.2. There are two A_3 curve segments each having an A_1A_3 fin point at both ends. There is also one A_3 curve loop. Four A_1^3 curve segments meet at an A_1^4 point shown as a dark blue sphere (Figure 6.18(b)). The

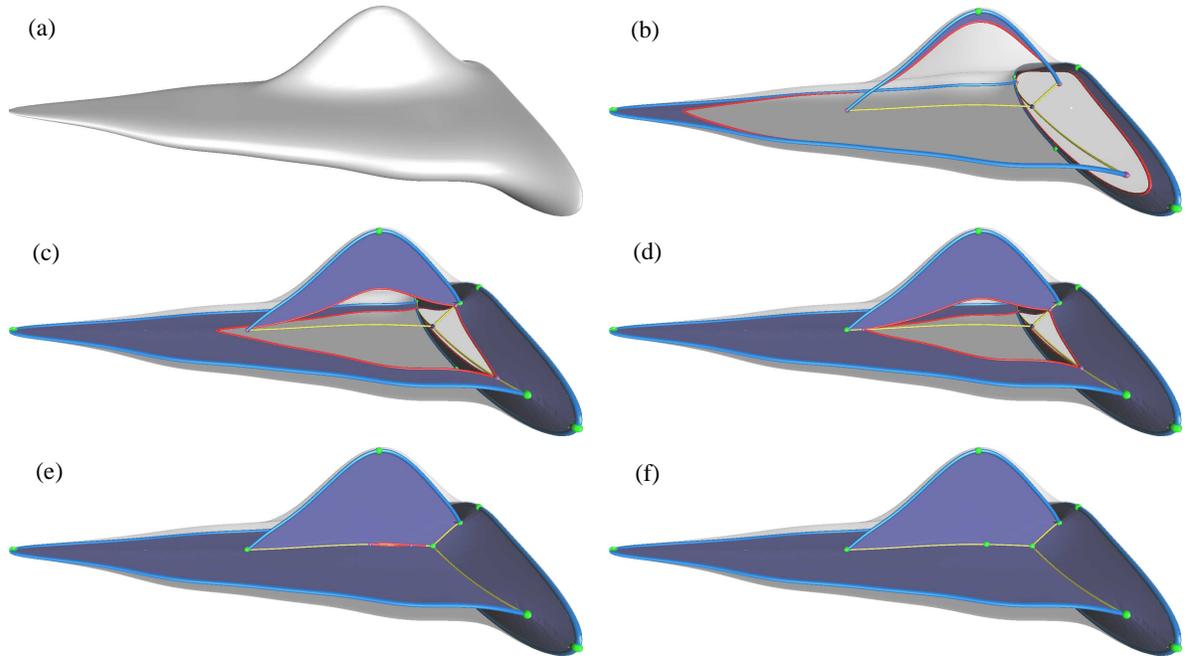


Figure 6.18. Several stages of evolution of medial axis of an object with multiple interconnected medial surfaces. Transition points are shown in green, evolving intersection curves are shown in red.

other ends of the four A_1^3 curve segments correspond to the A_1A_3 fin points where they meet A_3 curve segments.

In addition to accurate points on the medial axis, the proposed approach also accurately captures the topological structure of the medial axis. Figure 6.19 indicates all entity types on the medial axis of the region shown in Figure 6.18. The medial axis consists of surfaces (shown in orange, brown, bluish violet) bounded by edge curves (thick blue), and branch curves (thick yellow) where surfaces meet. The medial axis also contains fin points (purple spheres) where edge curves and branch curves meet, and a six junction point (green sphere) where six surfaces (and four branch curves) meet at a point. Since the evolution tracing is performed in parametric space, the boundary locations corresponding to each medial axis point are known. Therefore, the distance from each medial axis point to its corresponding boundary location is known, giving the complete medial axis transform. Further, the method also indicates the direction of increasing distance at all transition points, giving the shock structure [80].

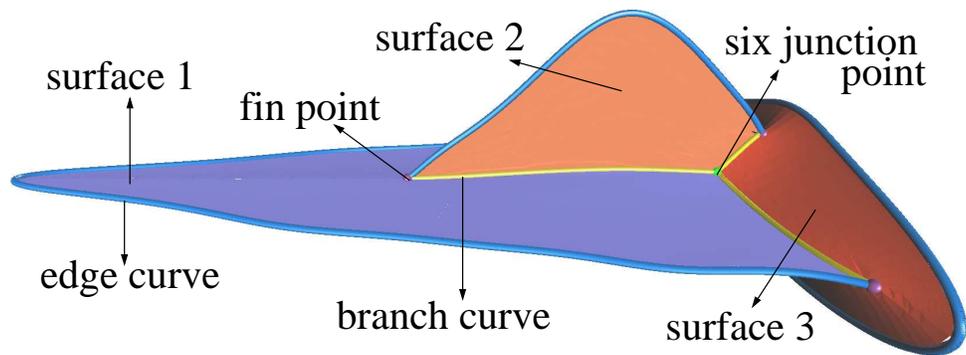


Figure 6.19. Medial axis of region shown in Figure 6.18(a) with all entities labeled.

Figure 6.20 shows the result of using a discrete approach for computing the 3D medial axis. The tight cocone [37] software was used to generate this result using a surface sampling consisting of roughly 8600 points. Although the result can be improved by using denser sampling of the object, this result illustrates typical issues with using discrete approaches. These issues include holes in the medial surfaces and degenerate, overlapping, and disconnected floating components. A_3 curves, A_1^3 curves, A_1A_3 points and A_1^4 points are not clearly identified and neither is their topological connection. This information must be inferred in a manual postprocess after clean up or fixing operations on the generated triangle meshes are performed.

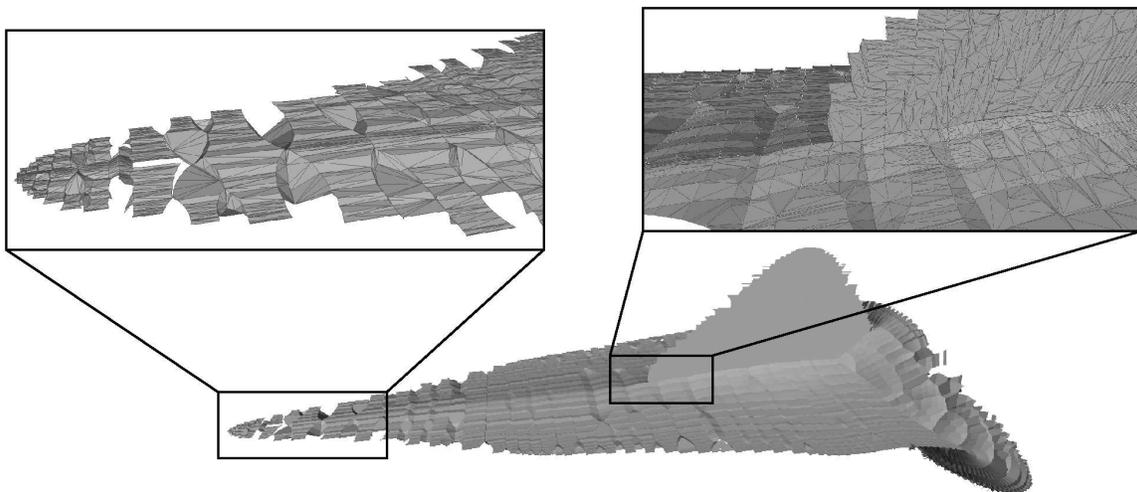


Figure 6.20. Medial axis computed using a discrete approach [37]

In Figure 6.21, the medial axis of the shown object consists of a single A_1^2 surface that is bounded by an A_3 curve and an A_1^3 curve. The A_3 and A_1^3 curves meet at both ends at common A_1A_3 points. The A_1^3 curve is designated as an *inessential fin curve* in [34]. The term arises from the fact that if the A_1^3 curve is collapsed and the corresponding A_1A_3 points are merged the surface will be a 2-manifold with a single boundary (A_3 curve) that is homeomorphic to a disk. Figure 6.22 shows a more complicated example where some of the A_1^3 curves form two loops that are connected at an A_1^4 point in the middle. Each loop is also connected to an A_1^4 point at the other end.

Although the approach is presented for the generic situation, the examples presented above do have certain nongenericities. For example, in the examples shown in Figure 6.16 and Figure 6.17, there are two simultaneously occurring curvature saddle point transitions that modify the same set of evolving intersection curves. Further, in the example shown in Figure 6.17, there are two simultaneously occurring A_1A_3 points. These situations are addressed as special cases in the implementation. The examples shown in Figure 6.16 and Figure 6.17 also have degenerate points that are sharp corners

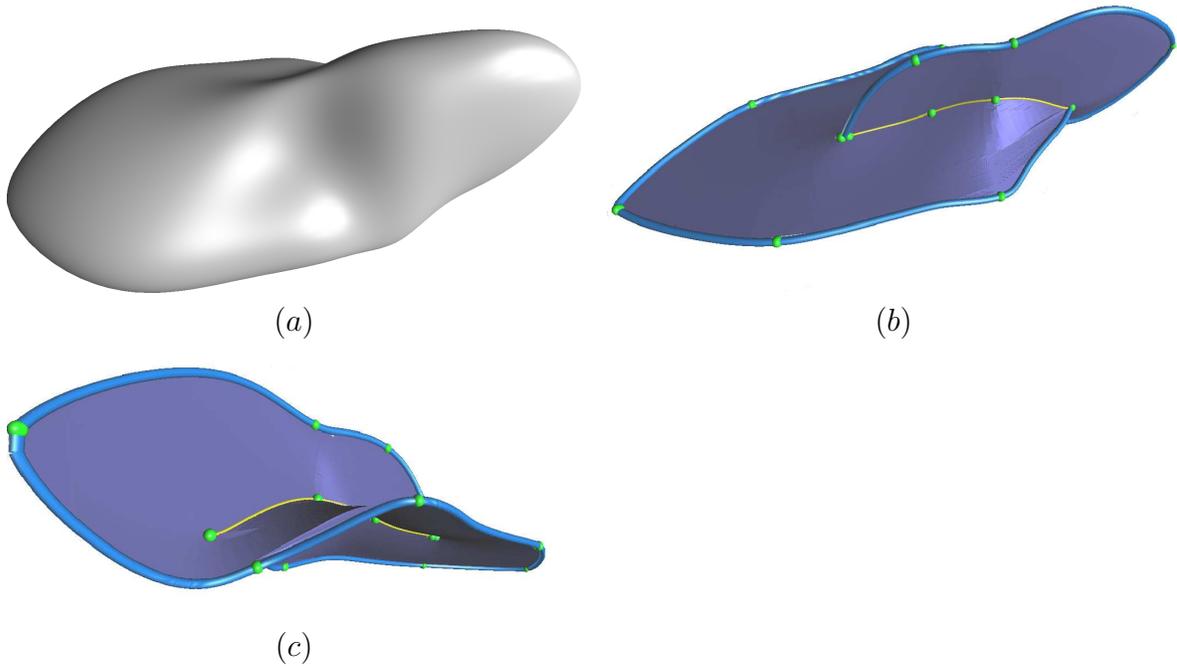


Figure 6.21. Two views of the medial axis of a region consisting of a single A_1^2 surface with an inessential fin type A_1^3 curve, where the A_1^2 surface turns into a fin onto itself [34].

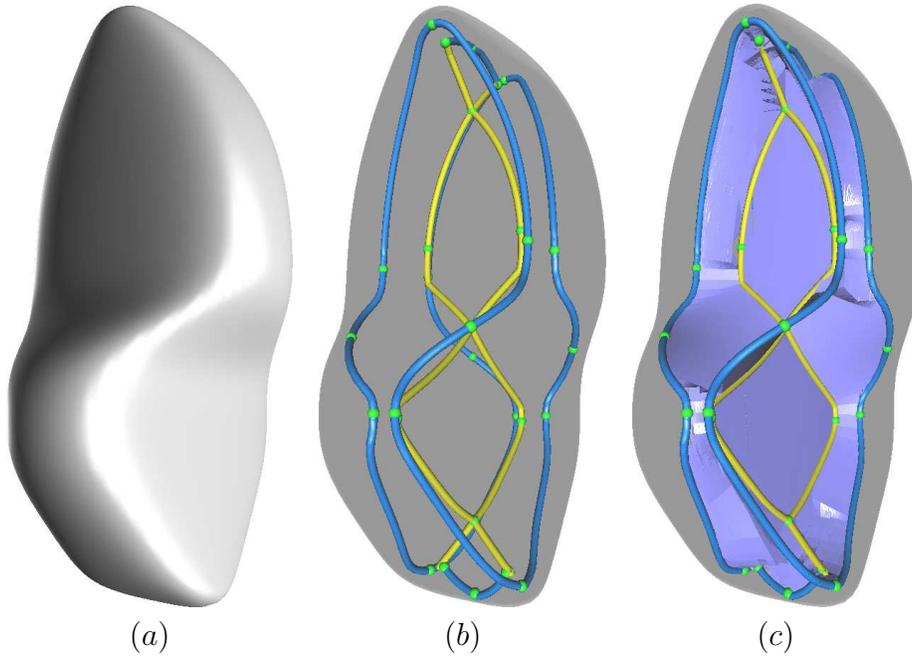


Figure 6.22. Medial axis of a region where a subset of the A_1^3 curves form two loops connected at an A_1^4 point in the middle, and an A_1^4 point at either end of the loops. (a) Input object. (b) A_3 curves shown in blue, A_1^3 curves in yellow and transition points in green. (c) A_1^2 surfaces shown in dull violet.

at the left and right extremities. These points are computed as curvature maximum points having a very small radius. Symmetrical regions typically result in nongeneric situations. For example, the medial axis of a sphere is a single point and that of a cylindrical region is a single curve. We are currently looking at extending the proposed approach to address such situations.

The computational complexity of the proposed algorithm is directly proportional to the number of transition points since it bounds the total number of intersection curves over the entire evolution for a given model. Likewise, the number of critical points, A_1A_3 points and A_1^4 points bound the number and hence computational complexity of A_3 and A_1^3 curve tracing. Table 6.1 presents running times on an Intel x64 machine with four cores and 8GB RAM for the examples shown here. It is evident from the table that most of the time was spent in computing the transition points since that involves B-spline root finding. We are currently investigating further enhancements to the root finding algorithms to reduce computation time. Although the total running time is longer than

Table 6.1. Computation Times (TP = Transition Point, C = Curves)

Biquintic surfaces (control points)	A_3 TP (min)	A_1^2 TP (sec)	$A_1^3 + A_1^4$ TP (min)	A_3 C, A_1A_3 (min)	A_1^3 C (sec)	Surfaces (sec)
Fig. 6.16 : 13×9	18	35	40	4.4	-	2
Fig. 6.17: 15×9	23	36	40	5.3	1	3
Fig. 6.18: 17×15	55	59	90	40	1	7

is desirable, each stage of the proposed approach is automatic, presenting accurate solutions with topology. Therefore, time consuming manual pruning and structure extraction steps are not required.

The techniques for computing transition points of A_3 , A_1^2 and A_1A_3 types provide all solutions. In order to reduce computation time for A_1^3 and A_1^4 transition points, they were required to be at least further apart than 5% of the size of the parametric domain in the examples shown. Therefore, no two transition events could occur within a region of this size. The topology of the computed medial axis is correct up to this specified accuracy. Between transition events, a much higher accuracy (10^{-4} of the size of the model in \mathbb{R}^3) was used to locate the medial axis points since the evolution of the intersection curves is smooth.

6.6 Multisurface Models

We now consider the case when a region of interest is bounded by more than one parametric B-spline surface. Multisurface models occur frequently in the area of computer-aided design (CAD). The theoretical results are quite general and extend to regions bounded by multiple surface patches stitched together since we present results considering different regions of a single surface. In order to address such types of models, computation of transition points involves a combinatorial search with subsets of curves or surfaces. The implementation of algorithms for computing curve elements and medial surfaces involves keeping track of the boundary components involved along with corresponding parameter values. Further, designed or sculptured models typically contain sharp edges and corners where surfaces meet. Sharp edges and corners introduce nongeneric behavior that are addressed in our approach. Surfaces joined at a sharp edge

will introduce a medial edge curve that is a degenerate A_3 curve along the edge. Three degenerate A_3 curves corresponding to sharp edges meet at a corner. An A_1^3 curve is also incident at the corner point.

Figure 6.23 shows a model bounded by six smooth parametric B-spline surfaces and its medial axis. The medial axis consists of thirteen A_1^2 surfaces that are bounded by twelve A_1^3 curves and the sharp edges of the model. Some of the A_1^3 curves have a sharp corner of the model as an end point. The medial axis also contains four A_1^4 points. Figure 6.24 shows several stages of the medial axis computation for this model. Figure 6.24 (a) shows all the sharp edges and corners of this model. Figure 6.24 (b) shows all the A_1^4 points and A_1^3 curves of its medial axis. Figures 6.24 (c) - (g) shows intermediate stages of the evolution of the A_1^2 surfaces of its medial axis. Figure 6.24 (h) shows the complete medial axis.

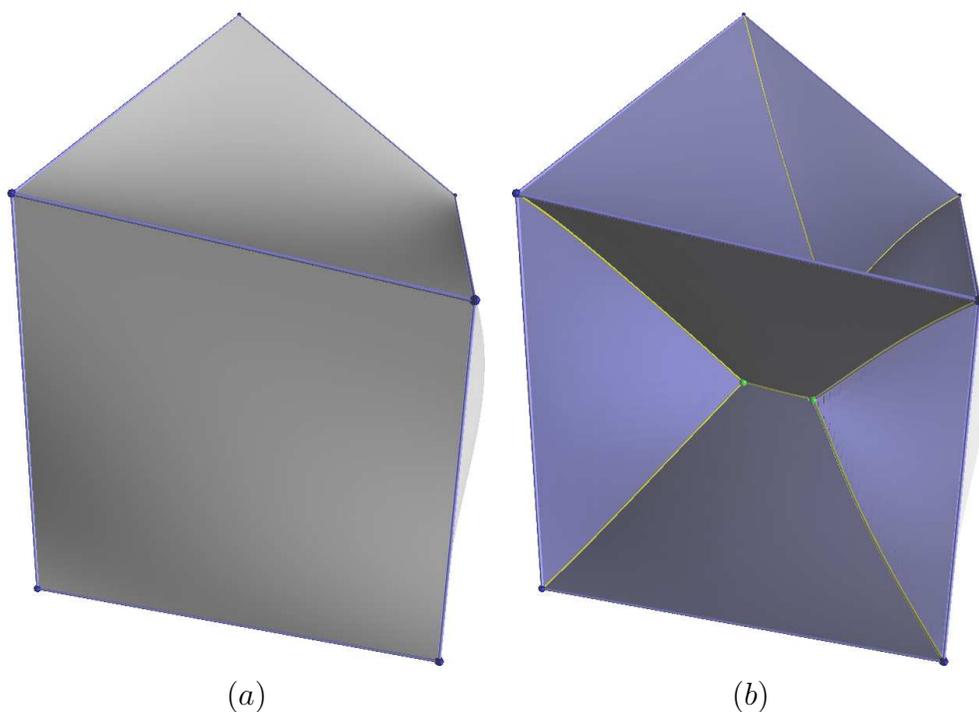


Figure 6.23. Result of medial axis computation on a multisurface model. (a) A model consisting of six parametric B-spline surfaces and (b) its medial axis. Medial surfaces are shown in dull violet, A_1^3 curves in yellow, A_1^4 points as green spheres, sharp edge curves in dull violet and corners as dark blue spheres.

6.7 Summary

A new algorithm for computing the medial axis of regions in \mathbb{R}^3 bounded by tensor product parametric B-spline surfaces is presented. The generic structure of the 3D medial axis consists of A_1^2 surfaces along with a singular set of A_3 curves, A_1^3 curves, A_1A_3 points and A_1^4 points. The proposed approach is based on the eikonal or grassfire flow of the bounding surfaces along the inward surface normal direction. The eikonal flow results in special transition points that create, modify or annihilate evolving (self-) intersection curves of the corresponding offset surfaces. The transition points as well as A_1A_3 and A_1^4 points are computed by solving geometric equations using B-spline based root finding techniques. The geometric equations are of high degree and dimensions and several strategies for reducing the computation time are provided. A_3 curves are computed using the technique presented in 5. A_1^3 curves are computed using a new evolution based tracing approach. A_1^2 surfaces of the medial axis are computed as a time trace of the evolving (self-) intersection set under the eikonal flow using an adaptation of the technique presented in [27] in conjunction with techniques for addressing special transition events of the eikonal flow.

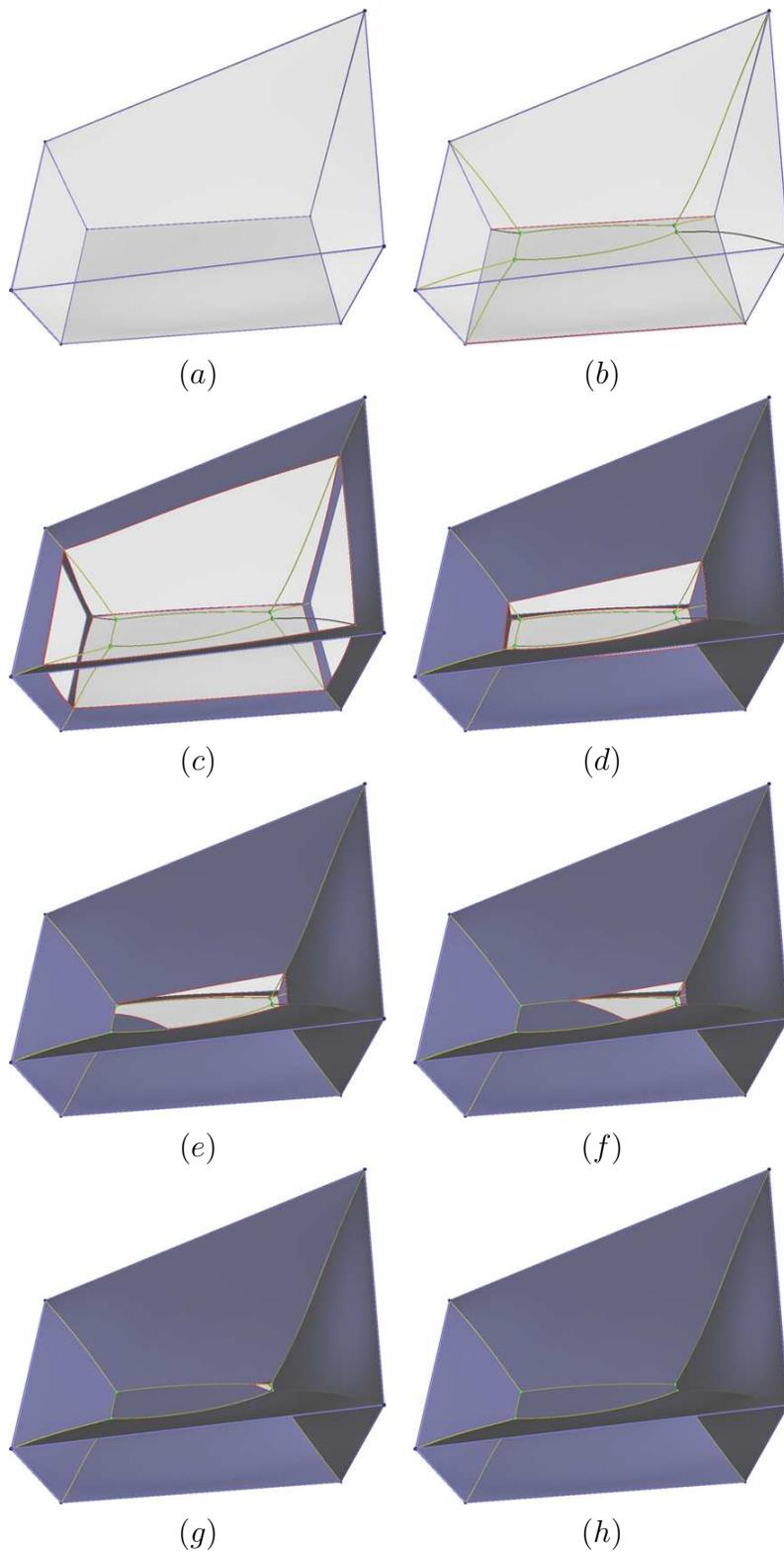


Figure 6.24. Several stages of evolution of medial surfaces of a multisurface model. A_1^4 points are shown in green, evolving intersection curves are shown in red.

CHAPTER 7

SHAPE ANALYSIS USING MEDIAL AXES AND RIDGES

This chapter presents several new techniques to augment computed ridges and medial axes to present enriched geometric and structural information. Some applications of the augmented structures are presented. The following sections are presented:

- Section 7.1 introduces a graph-theoretic approach for analysing the 3D medial axes to infer structural relationships between entities of medial axes and thereby their interpretation on the boundary surfaces.
- Section 7.2 introduces a new approach for shape analysis using 2D medial axes of level sets of scalar functions defined on surfaces.
- Section 7.3 extends the approach presented in Chapter 5 to compute ridges directly from volumetric data using smooth representations and demonstrates the benefits of such an approach.
- Section 7.4 introduces a new type of region-based shape analysis technique in the neighborhood of ridges.

7.1 Structural Analysis of 3D Medial Axes Using Graph Cycle Bases

The 3D medial axis consists of a collection of surfaces bounded by medial edge curves and junction curves. There is a natural desire to automatically segment the medial axis into its constituent pieces to enable localized modification of corresponding surface regions and selection of suitable subsets to obtain simplified object representations. With a suitable method to determine salient subsets of the medial axis, this will enable simplification of the object representation to retain only the subparts corresponding to the salient pieces of the medial axis. Constructing a graph representation has been the goal of several works in order to enable its use in computational shape matching and analysis applications. A review of existing approaches for graph-based analysis of medial axes is presented in Section 7.1.1.

We present a new method to compute salient features of an object from its medial axis. First, we propose a discrete representation of the 1-complex of a medial axis (Section 7.1.2) called the medial representation graph (MRG) (Section 7.1.3). A feature corresponds to a cycle in the MRG. The set of cycles in a graph forms a group under Z_2 addition. Then, a basis for this group (the cycle basis) corresponds to a set of features that we term as being salient. The definition of cycle bases is presented in Section 7.1.4. Since the cycle basis spans the cycle group, the features corresponding to the basis cycles span the set of all possible cycle features of a medial axis with the union operator. Therefore a cycle basis of the MRG is a representation of salient features for a new type of shape analysis framework. Section 7.1.5 presents computed examples of the cycle basis of the MRG for medial axes and their application in surface segmentation.

7.1.1 Related Work on Graph-based Analysis of Medial Axes

The 2D medial axis has a straightforward graph structure, which is not the case for the 3D medial axis. Giblin and Kimia [51] suggest a hypergraph structure of the medial axis consisting of nodes (fin and six junction points), links between nodes (boundary and junction curves) and hyperlinks between groups of links corresponding to medial surfaces. Inspired by the natural graph structure of 2D medial axes, the goal of their work was to characterize generic behavior of 3D medial axes so that a

similar graph structure may be obtained. Based on this idea, Leymarie [47] presents a directed graph representation called the *medial shock scaffold* consisting of shock nodes connected by a network of shock links. Shock nodes are points that generate, propagate or annihilate entities of the medial axis. Combinatorial geometric techniques are presented to estimate shock scaffolds from 3D point clouds [47] and polygonal meshes [80]. An initial set of *shock sources* of medial surfaces is computed by considering pairs of generators (points or polygons) and an iterative algorithm for linking shock nodes in increasing order of the corresponding radius function values at the nodes is presented. New shock nodes may be created during the linking step of the algorithm that are used to link nodes in later steps. However, fin points and medial boundary curves are not considered in their algorithm. Graph matching techniques are then used for shape registration applications using shock scaffolds [23].

Zhang et al. [119] construct directed acyclic graphs (DAG) from the medial axes of voxelized objects. Vertices of a DAG correspond to components (surfaces) of the medial axis and edges are inserted between adjacent components (those sharing a common junction curve). Voxels corresponding to different medial entity types are distinguished based on the average outward flux of the gradient of the distance (to object surface) function. Connected components are extracted and ranked based on the number of contributing voxels in each of them where components contributing more voxels are deemed more salient. A DAG is then created by arranging components in order of their saliency and inserting edges between components that share junction curves. This graph structure presents a hierarchical representation of the components of a medial axis that is then used for shape matching as presented in [119].

Damon [34, 35] presents a theoretical framework for organizing the components of medial axes of generic objects of arbitrary genus into multilevel graphs. Irreducible components of the medial axis are first identified based on topological slide operations. The first-level graph structure is then created based on adjacencies of the irreducible components. The irreducible components are then cut along junction curves to obtain the *Y-network* which is the second level graph structure.

7.1.2 1-complex of the 3D Medial Axis

We consider the 1-complex of a medial axis consisting of A_1A_3 points, A_1^4 points, A_3 curves and A_1^3 curves [51]. In the generic situation, the 1-complex of 3D medial axes forms a graph-like structure consisting of points (A_1A_3 fin points, A_1^4 six junction points) connected by curves (A_1^3 junction curves, A_3 edge curves), or closed curve loops, that bound medial surfaces. We define a feature of a medial axis as the connected set of medial surface points bounded by a loop in the 1-complex since it corresponds intuitively to a component or part of an object. We present a new approach for analyzing the structure of the 1-complex by decomposing it into a set of constituent loops (that are minimal in some sense) from which salient features of a medial axis are reconstructed. Salient features computed using the proposed approach form a set of building blocks for the object.

7.1.3 The Medial Representation Graph

In this section we define the *medial representation graph* (MRG) that will serve as a discrete representation of the geometry and topology of the 1-complex of a medial axis. The MRG is a multigraph with one vertex for each entity of the medial representation, and edges connecting adjacent entities. There are four different types of entities (of differing dimension) in the 1-complex that we denote using the terms in Table 7.1. Figure 7.1 illustrates the MRG for the medial axis shown in Figure 1.2 (b).

Remark 1 *The MRG of a medial representation is unique.*

We note some basic properties of vertex neighborhoods in the MRG.

Table 7.1. Entities and Their Labels

Entity	Vertex
A_1A_3 fin point	Ⓕ
A_1^4 6-junction point	Ⓖ
A_3 edge curve	Ⓔ
A_1^3 junction curve	Ⓙ

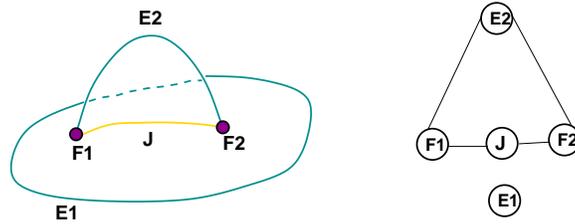


Figure 7.1. An illustration of the 1-complex of the medial axis from Figure 1.2 (b) and its MRG.

1. An A_3 curve \textcircled{E} is adjacent two A_1A_3 points when they are open. When closed, they are isolated vertices.
2. An A_1A_3 point \textcircled{F} is always adjacent to one A_1^3 curve \textcircled{J} and one A_3 curve.
3. An A_1^4 point \textcircled{S} is adjacent to four A_1^3 curves. Two of the A_1^3 curves could be the same forming a loop. The MRG in this case is a multigraph.
4. An A_1^3 curve is adjacent to two point type entities (A_1A_3 or A_1^4). The curve can also be a closed loop or loop at an A_1^4 point. It is an isolated vertex in the former case and has multiedges in the MRG in the latter case.

7.1.4 Cycle Basis of a Graph

We review the definition of the *cycle basis* of a graph as presented in [72]. Let $G = (V, E)$ be an undirected graph. A *cycle* in G is a subgraph of G in which each vertex has degree exactly 2. Note that this subgraph might have more than one component. We can represent each cycle by a characteristic vector over E (a 1 in coordinate i indicating that edge e_i is present, and a 0 indicating absence), in which case the set of cycles forms a vector space over \mathbb{Z}_2 . A *cycle basis* for G is a basis of this vector space, and consists of a set of cycles. The cycle basis is a compact way of describing all cycles in a graph; while there may be exponentially many cycles in G , the cycle basis has exactly $|E| - |V| + CC(G)$ elements (where $CC(G)$ is the number of connected components in G). We note that these definitions extend to the case when G is a multigraph; in this case, each parallel edge is treated as a distinct edge (and thus a distinct dimension in the associated vector space).

Let $w : E \rightarrow \mathbb{R}^+$ be a weight function on E , and let the *weight* of a cycle as the sum of its edge weights. The *minimum weight cycle basis* is then a cycle basis of minimum total weight. Such a basis can be computed in polynomial time; the first such algorithm was given by Horton in 1987 [65].

Graph cycle bases have been applied to geometric problems such as extracting topological information from point clouds [55] and meshing genus-1 point clouds [125]. Conceptual relationships between cycle bases and generators of the homotopy group and the first homology group of 2-manifolds have been exploited in various geometric algorithms [46, 36, 25].

7.1.5 Application of the Cycle Basis of the MRG for Shape Analysis

The central observation here is that a cycle basis of the MRG determines salient features of the medial surface. We illustrate this with the example shown in Figure 7.1. The cycle basis for this example is trivial. It consists of the two constituent cycles. Each cycle corresponds to a medial surface component shown by the blue and orange colored surfaces in Figure 1.2 (b). An isolated \textcircled{E} vertex in this example corresponds to an A_3 curve bounding a single medial sheet. An isolated \textcircled{J} vertex will correspond to an A_1^3 curve bounding three medial sheets. Note that in general multiple isolated vertices might correspond to a single sheet (the medial surface of a multihanded torus is one example).

Using the medial axis computation approach presented in Chapter 6, the MRG and cycle basis can be automatically computed for B-spline models. Figure 7.2 presents the MRG and a cycle basis for the model presents in Figure 6.19. Each component of the model corresponding to its basis cycle is mapped onto the surface as shown by the orange, brown and dull violet colors in Figure 7.2 (c) and (d). This presents a medial axis based surface segmentation.

Figure 7.3 presents an example of a computer-aided design application using this information. The surface region corresponding to one of the basis cycles is trimmed (and separated from the model) as shown in Figure 7.2 (a). The trimmed region is replaced with a different surface in the same region. In Figure 7.2 (b), points on the

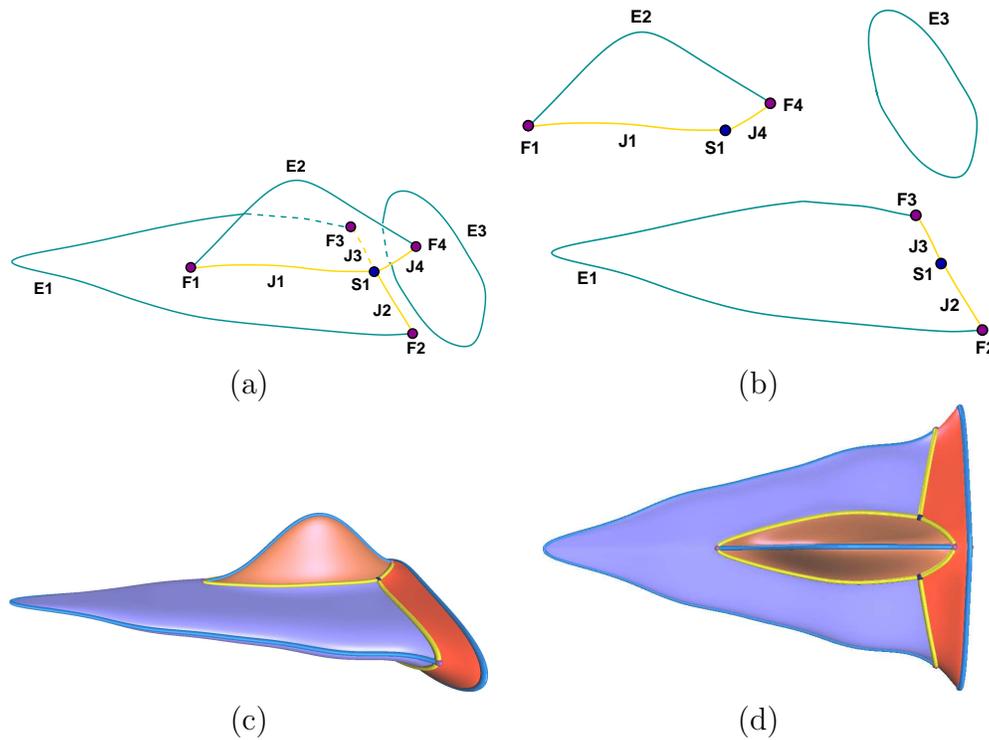


Figure 7.2. Cycle basis analysis of a B-spline model shown in Figure 6.19. (a) a sketch of the medial axis with MRG vertex labels. (b) basis cycles of the MRG. (c) cycle basis features of medial axis mapped onto surface to obtain segmentation (side view). (d) top view of the same result.

replacement surface are offset by an amount proportional to the magnitude of the larger principal curvature of corresponding points on the original surface.

The cycle basis approach can be applied when the medial axis is computed using a discrete approach as well. However, manual analysis of the structure of the 1-complex is a tedious process for complicated medial axes. Figure 7.4 presents an example of the cycle basis approach using a polygonal surface model of a hip bone (Figure 7.4 (a)) and its medial axis (Figure 7.4 (b)) computed using the tight cocone algorithm [37]. A sketch of its MRG showing all entities of the 1-complex of its medial axis is shown in Figure 7.4 (c). The medial axis contains all types of generic entities of the 1-complex of a medial axis. The medial axis also contains several inessential fins where a surface sheet turns into a fin onto itself [34] (See Figure 7.4 (c)). The cycle basis for the MRG mapped to geometry on the medial axis is shown in Figure 7.4 (d) where each basis cycle is shown in a different color. There are six basis cycles for this medial axis. The

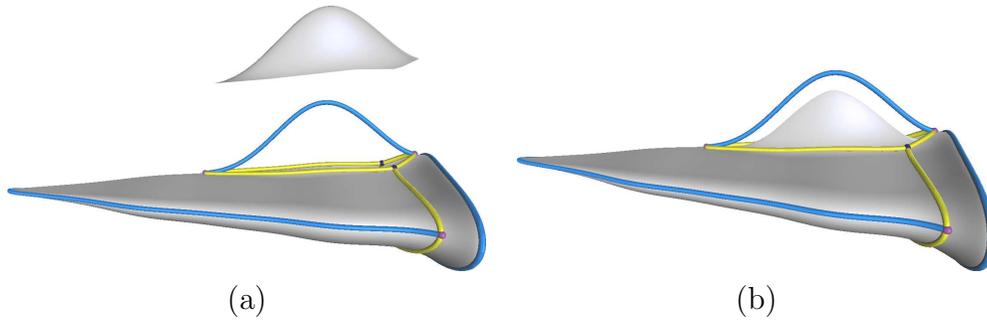


Figure 7.3. Structural component-based design. (a) The surface region corresponding to one basis cycle component is trimmed. (b) The corresponding surface region is shrunk using a curvature-based variable offset.

medial axis feature associated with each basis cycle is shown in Figure 7.4 (e).

When mapped to the object surface, the cyan, magenta and green features seem to have a close correspondence to the ilium, ischium and pubis anatomical regions of the hip bone, respectively (Figure 7.4 (f)). This observation is a strong indication that medial axis features computed using the cycle basis identify a set of sub parts of which an object is composed. Figure 7.4 (f) also presents a new type of surface segmentation based on the structural composition of an object.

The above examples demonstrate the cycle basis approach to compute a set of salient features using medial axes. Equal weights were used for all edges in the MRGs for computing cycle bases. Under this condition, the segmentation of the medial axis arising from the computed cycle basis given results similar to [109]. However, weights can be assigned to edges in the MRG based on geometric properties of the medial entities that may result in other features. For example, saliency measures such as separation angle [49, 123], fire-front meet angle [100], object angle [39] and weak feature size [24] may be used. The cycle basis approach therefore enables a generalized framework for computing salient features.

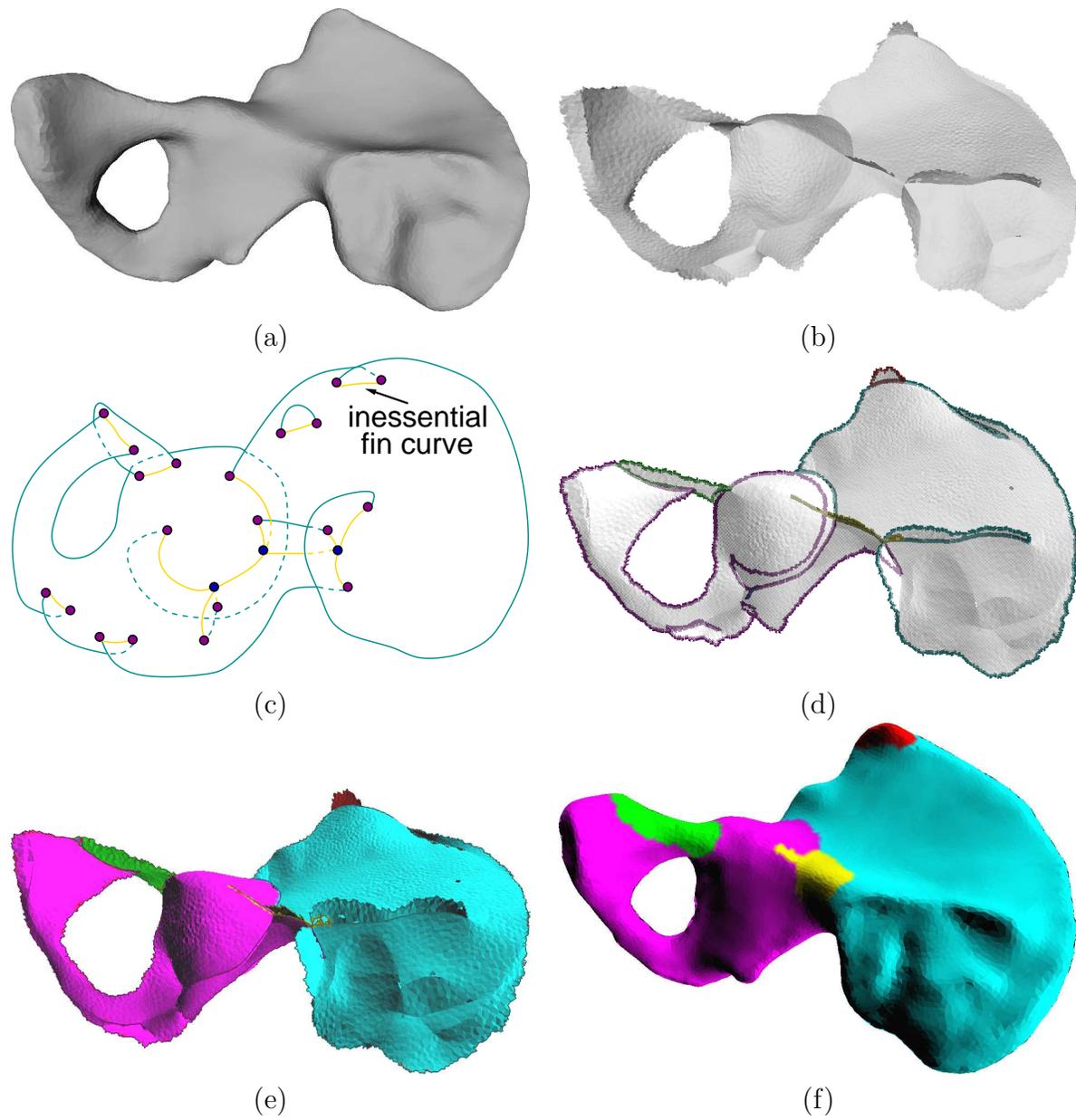


Figure 7.4. Cycle basis analysis of a hip bone model. (a) polygonal surface mesh of a hip bone. (b) medial axis of the surface mesh. (c) a sketch of the corresponding MRG. (d) basis cycles computed using the proposed approach shown in different colors. (e) salient features corresponding to basis cycles (features shown with color of the basis cycle). (f) medial basis features mapped on surface mesh to obtain segmentation.

7.2 Shape Analysis Using Scalar Functions on Surfaces

This section presents a new method for shape analysis using scalar functions defined on surfaces. This is joint work with Tobias Martin and Guoning Chen.

Several techniques for structural analysis of shapes using topological analysis of scalar functions defined on the boundaries of 3D objects (polygonal) are surveyed in [11]. The level set diagram [78] is one such example where structural information is computed by extracting geometric information at several level sets of a scalar function defined on the surface. The level set diagram presents structure as a skeleton consisting of curve segments that is constructed by connecting barycenters of isocontours corresponding to the level sets.

We present a generalization of this concept, where structural information is extracted by computing 2D medial axes of regions bounded by the isocontours at level sets of a scalar function. Evolution of the medial axes along level sets with monotonically varying isovalues provides information about structure of the shape. When the medial axes are aggregated in a sweeping manner across all level sets, it presents a medial-like structure for the entire object. Accurate 2D medial axes are computed using the approach presented in Chapter 4 of this dissertation. Automatic creation of graph structures of 2D medial axes that is a result of correct topology conforming to generic structure computed using our approach, as well as accurate computation directly from smooth representations that do not introduce extraneous artifacts, enable the sweeping operation to generate consistently connected curve and surface elements of the medial structure.

The computational procedure for this approach begins with a specification of a Morse function on the object. The object is then decomposed into a sequence of nonplanar slices. Each slice is a level set of the Morse function. Each slice is flattened using an algorithm such as LSCM [79] or ABF++ [114] and its boundary is approximated with a closed parametric B-spline curve. A medial axis is computed for each component of every slice using the approach presented in Chapter 4. The computed medial axes are then mapped back onto the respective nonplanar slice (unflattened). Starting from the

first slice, the medial structure for the object is iteratively constructed by matching and connecting the medial axes of two adjacent slices until the last slice is reached. For smoothly changing geometry of the boundary, there are only two generic transitions of the 2D medial axis [50]. In our approach, adjacent slices are created close enough so that their boundaries are assumed to change smoothly. This enables a simple graph matching algorithm that addresses both topological changes to match two successive medial axes. A triangulation is created between matching pairs of medial curve segments.

Figure 7.5 (a) shows an object represented by a polygonal mesh. Figure 7.5 (b) visualizes isocontours of a user-specified Morse function on the object. Figure 7.5 (c) shows several slices of the object that are level sets of the Morse function. Figure 7.5 (d) shows the 2D medial axes of the level sets. Figure 7.5 (e) shows the result of connecting successive pairs of 2D medial axes. Colored surface sheets indicate structural components. Different surfaces are distinguished by tracking segments of the 2D medial axes as they grow and die over the range of isovalues used to generate them.

7.2.1 Examples and Discussion

We present examples of this approach for shape analysis of 3D polygonal objects using user-defined scalar Morse functions on the surface. Figure 7.6 presents examples for several objects and a comparison of a medial axis computed for each object using a discrete algorithm [91]. The results indicate that the swept medial structure presents the global structure of an object. For the chosen set of Morse functions, the swept medial structures are similar to the actual medial axis.

The iterative construction of the swept medial structure allows us to track topological changes of the medial axes of slices along the user desired cutting orientation. As shown in Figure 7.6 (left column), differently colored sheets represent the evolution of their individual feature components of the 2D medial axes. This result must be performed manually in the case of a 3D medial axis computed using a discrete approach, and is therefore tedious and time-consuming. Simplification of the medial axes of the level sets results in contraction to single points. This leads to a hybrid structure where curve segments are created in tubular regions and surfaces are generated in other more general regions shown in Figure 7.6 (c) and (e).

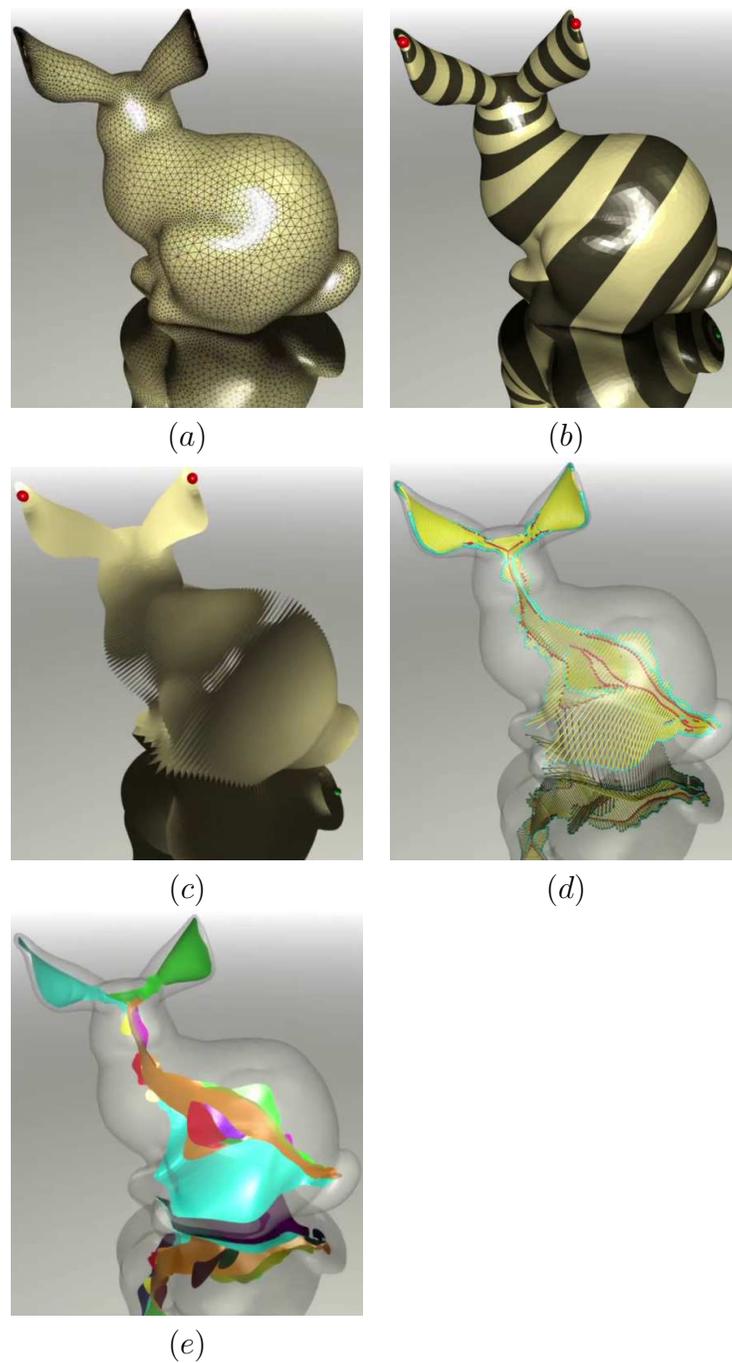


Figure 7.5. Structural analysis using 2D medial axes. (a) Input polygonal mesh. (b) Scalar Morse function defined on surface. (c) Level sets of scalar function. (d) 2D medial axes of level sets. (e) Connected medial structure indicating structural features.

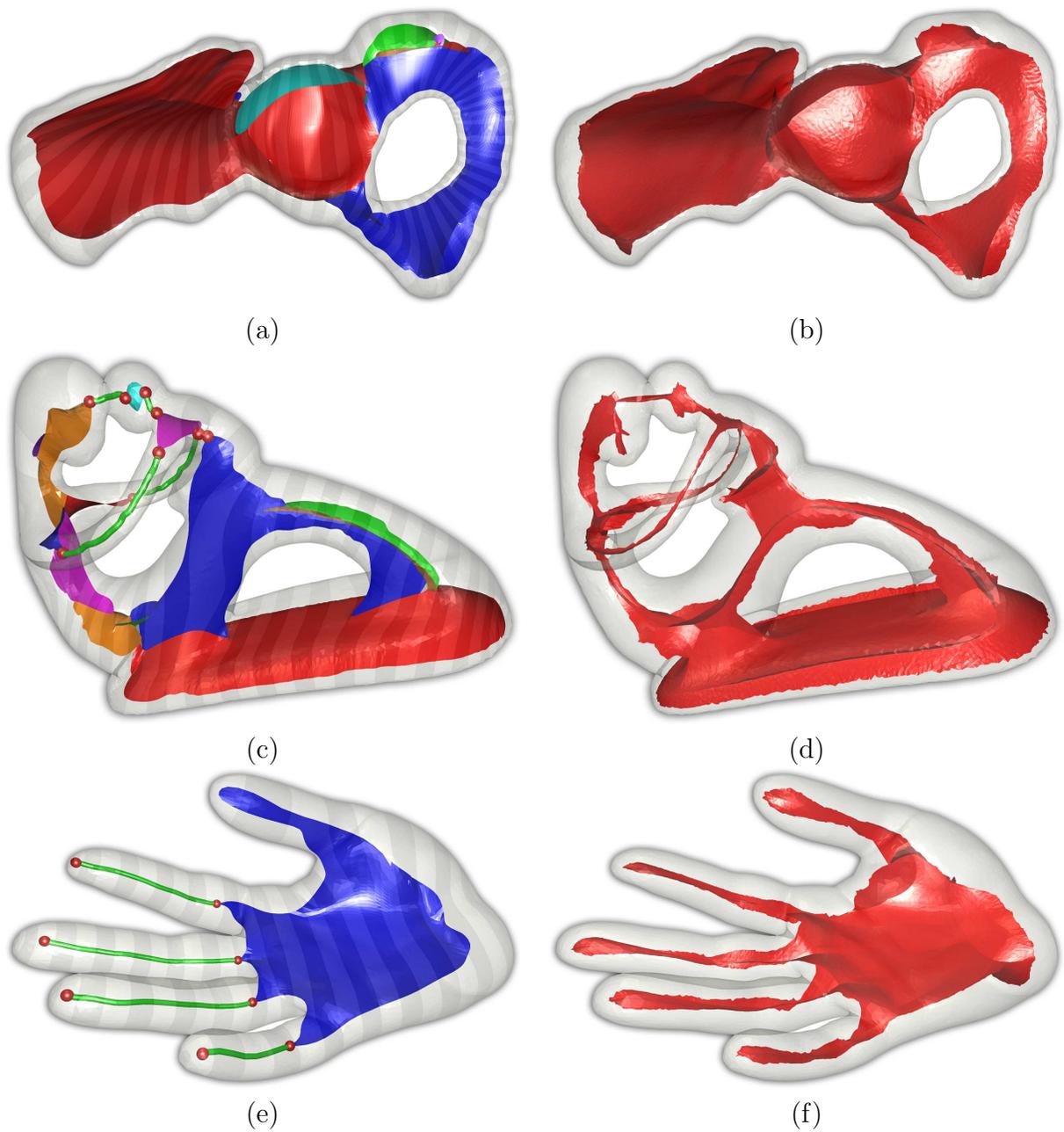


Figure 7.6. Swept medial structure (left) computed using our approach, and medial axis (right) computed using a discrete algorithm [91]. Different surface sheets in our results are shown in different colors. The scalar functions used for computing the swept medial structures are also visualized (left).

7.3 Ridges from Isosurfaces of Volumetric Data

Ridge curves have been found to be useful as *landmarks* for shape matching [56, 98, 122]. Such applications typically have input in the form of volumetric MRI or CT images (3D scalar-value grids), and the surface of interest a level-set or isosurface at a given scalar value. Three-dimensional data grids are also abundantly available from simulation data where the resulting grids approximate solutions of discretized partial differential equations, and in the field of graphics and visualization in the form of level set models [18].

Using implicit trivariate B-spline representations of the isosurface of interest, we present a generalization of the tracing algorithm for parametric B-spline surfaces to extract all generic ridges of the isosurface directly from the smooth representation. As presented in Section 2.3, existing approaches for extracting ridges from isosurfaces of volumetric data tend to result in sets of disconnected ridge segments or tend to have undesirable undulations. By tracing on a smooth representation, the extracted ridge curves conform to generic behavior and are therefore continuous connected curves. B-splines act as low pass filters on the data grid and tends to smooth out high frequency characteristics such as noise and thus the ridges do not have unexpected undulations. In addition, a smooth representation allows robust detection of isolated umbilics, and thus, ridges around umbilics, to present a complete solution.

7.3.1 Implicit B-spline Representation of Isosurfaces of Volumetric Data

Consider a parallelepiped region $\Omega \subset \mathbb{R}^3$, where $\Omega = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$. Let $f : \Omega \rightarrow \mathbb{R}$ be a $C^{(4)}$ trivariate function that maps a point $(x_1, x_2, x_3) \in \Omega$ to a scalar value. Given a specific isovalue $\hat{a} \in \mathbb{R}$,

$$\mathcal{I} = \{(x_1, x_2, x_3) : f(x_1, x_2, x_3) = \hat{a}\} \quad (7.1)$$

forms an implicit surface also called an isosurface or level set at isovalue \hat{a} . If $\nabla f \neq [0 \ 0 \ 0]^T \forall (x_1, x_2, x_3) \in \mathcal{I}$, the isosurface is guaranteed to be a 2-manifold [15].

The implicit function theorem states that around every point on the isosurface there exists a neighborhood in which the isosurface can be represented as a Monge surface

using at least one of (x_1, x_2) , (x_2, x_3) , (x_3, x_1) as parameter variables. For example, when $\frac{\partial f}{\partial x_3} \neq 0$, there exists a scalar field $g(x_1, x_2)$ such that the isosurface can be represented as $S(x_1, x_2) = (x_1, x_2, g(x_1, x_2))$ where $f(x_1, x_2, g(x_1, x_2)) = \hat{a}$. The first, second and third order partials of $S(x_1, x_2)$ are computed using this framework which is in turn required to evaluate principal curvatures, principal directions and curvature gradients at any point $(\hat{x}_1, \hat{x}_2, \hat{x}_3) \in \mathcal{I}$. The reader is referred to [120, 127] for derivation of the formulae for computing principal curvatures, principal directions and curvature gradients defined on the implicit surface.

In this work, $f(x_1, x_2, x_3)$ is a trivariate B-spline defined as

$$f(x_1, x_2, x_3) = \sum_{\mathbf{i}=1}^{\mathbf{n}} c_{\mathbf{i}} \mathcal{B}_{\mathbf{i}, \mathbf{d}, \mathbf{\Gamma}}(x_1, x_2, x_3), \quad (7.2)$$

where $c_{\mathbf{i}} \in \mathbb{R}$ are the coefficients of a $n_1 \times n_2 \times n_3$ control grid and $\mathbf{i} = (i_1, i_2, i_3)$ and $\mathbf{n} = (n_1, n_2, n_3)$ are multiindices. Every coefficient has an associated piecewise polynomial basis function

$$\mathcal{B}_{\mathbf{i}, \mathbf{d}, \mathbf{\Gamma}}(x_1, x_2, x_3) := \prod_{j=1}^3 B_{i_j, d_j, \Gamma_j}(x_j), \quad (7.3)$$

where $B_{i_j, d_j, \Gamma_j}(x_j)$, $j = 1, 2, 3$ are linearly independent B-spline basis functions. $B_{i_j, d_j, \Gamma_j}(x_j)$ as defined in [29] is a piecewise polynomial of degree d_j with knot vector $\Gamma_j = \{t_k^j\}_{k=1}^{n_j+d_j}$ that has local support and is $C^{(d_j-1)}$. In order for the ridge functions ϕ_1 and ϕ_2 to be continuous, third order derivative smoothness is required ($d_j = 4$). In this case, Γ_j is a uniform and open knot vector where the first five and last five knots of Γ_j are a_j and b_j respectively. To distinguish crests from other types of ridges, fourth order derivative smoothness ($d_j = 5$) is required to compute second derivatives of curvatures. In this case, Γ_j is a uniform and open knot vector where the first and last six knots of Γ_j are a_j and b_j respectively.

Given a scalar-valued volumetric grid, we represent the region using a trivariate B-spline by using the scalar values as coefficients $c_{\mathbf{i}}$ in Equation 7.3. The corresponding B-spline basis functions can be viewed as smoothing low pass reconstruction filters [87] of the samples $c_{\mathbf{i}}$ that does not introduce additional geometric features on the isosurface.

7.3.2 Generalization of Tracing Algorithm

The tracing algorithm for implicit trivariates follows the same framework of advancing and sliding from seed points. This section presents techniques for addressing new challenges that arise with computing seed points and tracing with the implicit trivariate representation. Curvatures, principal directions and curvature gradients required for evaluating the ridge function are computed using a local parameterization of the isosurface given by the implicit function theorem [120].

7.3.2.1 Seed Points

Extremal points of κ_i and umbilics are computed as simultaneous roots of three equations in three unknowns as given in Equations (7.4) and (7.5), respectively,

$$\begin{aligned} f(x_1, x_2, x_3) &= \hat{a} \\ \frac{\partial \kappa_i(x_1, x_2, x_3)}{\partial x_1} &= 0 \end{aligned} \tag{7.4}$$

$$\frac{\partial \kappa_i(x_2, x_2, x_3)}{\partial x_2} = 0$$

$$\begin{aligned} f(x_1, x_2, x_3) &= \hat{a} \\ \frac{\partial Q(x_1, x_2, x_3)}{\partial x_1} &= 0 \end{aligned} \tag{7.5}$$

$$\frac{\partial Q(x_1, x_2, x_3)}{\partial x_2} = 0$$

wherein the isosurface is locally parameterized using (x_1, x_2) and coefficients of the LHS are computed symbolically. When using the (x_1, x_2) parameterization, it is assumed that $\frac{\partial f}{\partial x_3} \neq 0$ so that the implicit function theorem is valid. However it is possible that $\frac{\partial f}{\partial x_3} = 0$ within Ω . Therefore, similar equations are derived for (x_2, x_3) and (x_3, x_1) parameterizations and seed points are computed using these parameterizations as well.

In practice, obtaining roots of these equations using techniques presented in [43, 45] is computationally very demanding in terms of both time and memory even for reasonably sized trivariate B-splines. We have developed several optimizations to reduce compute time.

First, subregions of the trivariate that potentially contain the isosurface are extracted. The domain of the trivariate in Section 7.3.1 is $[t_{d_1+1}^1, t_{n_1-1}^1] \times [t_{d_2+1}^2, t_{n_2-1}^2] \times [t_{d_3+1}^3, t_{n_3-1}^3]$. Every knot span subdomain $\{[t_{k_1}^1, t_{k_1+1}^1] \times [t_{k_2}^2, t_{k_2+1}^2] \times [t_{k_3}^3, t_{k_3+1}^3]\}$, $k_1 = d_1 + 1 \dots n_1 - 2$, $k_2 = d_2 + 1 \dots n_2 - 2$, $k_3 = d_3 + 1 \dots n_3 - 2$, is extracted as a Bézier trivariate and retained if the range of $f(x_1, x_2, x_3)$ within the subdomain contains the isovalue of interest. The convex hull property of the trivariate Bézier representation enables an efficient test of checking the 1D AABB of the coefficients of $f(x_1, x_2, x_3)$ of the region representing the subdomain for this purpose. Since the subdomains are typically very small for a reasonably high resolution data set, a large number of subdomains are rejected in this step. Table 7.2 compares the percentage of subdomains retained for seed point computation for some of the examples presented here.

Second, computing coefficients of the LHS of the equations symbolically is still computationally expensive even though they are in Bézier form. An expression tree approach was presented in [43] to reduce computational demands of multivariate B-spline constraint solvers especially when the different terms in an equation are functions of different independent variables. In our work, the high degree terms in the equations for computing seed points are functions of the same independent variables x_1 , x_2 and x_3 . We have developed a variant of the expression tree approach to address this situation. The equations are represented as expression trees as in [43] and coefficients are computed only for $f(x_1, x_2, x_3)$ and its partial derivatives up to third order, which are low degree terms. It should be noted that a term involving a partial derivative of f appears multiple times in the LHS of the equations. In the expression tree approach of [43], a copy of this term is stored in every repeated leaf node. This can lead to redundant subdivisions of each copy. In our method only one global copy of each term is stored and only the global copies of the terms are subdivided during the subdivision step. This

Table 7.2. Isosurface Subdomains

Model	Source	Size of trivariate control grid	Subdomains with isosurface	% of subdomains with isosurface
Skull	CT scan	128 x 128 x 128	73447	3.5
Silicium	Simulation data	34 x 34 x 98	15231	13.4

approach is similar in spirit to the idea presented in [43] for efficient data structure management using reference counted pointers but were proposed for cases when the leaf nodes of an expression tree are functions of different independent variables. The AABBs for a subdomain are computed using interval arithmetic as presented in [43]. In our experiments, we have found that this reduces computation time by over an order of magnitude. Constraint solving by computing the coefficients for a Bézier trivariate subdomain took about four minutes on average per parameterization. The optimized approach presented here required a few seconds.

In addition, in some cases, seed points may have to be computed for all three possible parameterizations of the isosurface (Section 7.3.1). However, in our experiments we noticed that most of the time a single parameterization is sufficient for a given subdomain. In order to determine an appropriate parameterization, we first select one such parameterization $x_i x_{i \oplus 1}$. If the range of $\frac{\partial f}{\partial x_{i \oplus 2}}$ potentially does not have a zero within the subdomain, then this parameterization is the only one used for computing seed points. If $\frac{\partial f}{\partial x_{i \oplus 2}}$ does potentially have a zero within the subdomain, then one of the other two candidate parameterizations are similarly tested. It is possible that $\frac{\partial f}{\partial x_1}$, $\frac{\partial f}{\partial x_2}$ and $\frac{\partial f}{\partial x_3}$ all potentially have a zero (but not at the same point) in which case all three parameterizations are used.

Further, the constraint solver for different subdomains are executed in parallel since they are independent. These optimizations significantly reduce time and thus enable seed point computation for large data sets.

7.3.2.2 Tracing

The following additional issues are addressed for tracing:

1. At each step of the trace, it is imperative that the orientation of the normal of the isosurface is globally consistent since the convention of which curvature, κ_1 or κ_2 , is the larger one is dependent on it. Since ∇f is the normal of the global implicit representation at any point and is oriented consistently, the normal computed using the local parametrization of the isosurface ($S_{x_1} \times S_{x_2}$) is compared with it.

If the directions of the two vectors are opposite, then $\kappa_1, t_1, \nabla\kappa_1$ and $\kappa_2, t_2, \nabla\kappa_2$ are swapped and the signs of $\kappa_i, \nabla\kappa_i$ $i = 1, 2$ are changed.

2. It is assumed that $\nabla f \neq [0 \ 0 \ 0]$. However, it is possible that up to two of the quantities $f_{x_1}, f_{x_2}, f_{x_3}$ are zero at a point. At every advance and slide step of the trace, the algorithm selects the most numerically appropriate parameterization for evaluating the ridge function based on which of $f_{x_1}, f_{x_2}, f_{x_3}$ has the largest magnitude. This enables tracing of ridges passing through such points and even lying exactly on such points. All the ridges of the ellipsoid shown in Figure 7.10 lie along curves where either f_{x_1}, f_{x_2} or f_{x_3} are zero since $f_{x_i} = 0$ at $x_i = 0$, $i = 1, 2, 3$.
3. At every step, the trace moves off the isosurface (along one of the principal directions) and is projected back onto the isosurface using a standard technique of iteratively marching along ∇f until the isosurface is reached.

7.3.3 Results and Discussion

We demonstrate the method presented here on a CT scan, a 3D data grid arising from simulation results, on implicit B-spline representations of isosurfaces resulting from isogeometric analyses on a volumetric B-spline model, and on algebraic surfaces. Ridges of κ_1 are shown in blue and ridges of κ_2 are shown in red. Crests are shown as thicker curves. $|\phi_i|$ is used as a measure of the accuracy of the ridges extracted. A user specified accuracy (typically 10^{-2} or 10^{-3}) is used as an input parameter for the tracing algorithm and all ridges extracted in generic regions using our method satisfy the accuracy requirement.

Our algorithm has been implemented in the Irit modeling environment [41]. The results presented here have been generated on an Intel Xeon X7350 processor with 32 cores and computation times are shown in Table 7.3. Since seed point computations for different subregions are independent processes, a speedup roughly equal to the number of threads is achieved using a parallel implementation. Similarly, ridge tracing from different start points are also independent processes and a significant speedup could

Table 7.3. Computation Time

Model	Seed points (32 threads) (minutes)	Tracing (1 thread) (minutes)
Silicium	31	52
Skull	122	167
Cube (isogeometric)	12	2

be achieved using a parallel implementation. However, this has not been currently implemented.

Figure 7.7 shows ridges and crests extracted from a CT scan where the skull isosurface is identified at intensity value 69.5. This result can be compared to the results in Figure 15 of [127] and Figures 12, 13, 19 and 20 of [92] that show crests extracted from volumetric images of skulls. Figure 7.7 shows that our method captures a very high level of detail with smooth crest curves whereas previous grid based methods result in a sparse collection of fragmented crest segments. The crests on the top and side of the skull correspond to scanning artifacts of the data set and are accurately captured by our method. We attempted to create a polygonal mesh representation of the isosurface using marching cubes but due to the geometric and topological complexity, the mesh failed to be suitable for use with the ridge extraction method in CGAL even after considerable manual effort to correct the errors. Afront [111] generated a suitable mesh for the algorithm in CGAL after several hours of computation but many of the features presented in the original data set were missing. Our method avoids issues related to mesh generation for complex data sets and extracts ridges directly from B-spline filtered smooth representations of the 3D grids.

Figure 7.8 compares ridges extracted on a 3D grid resulting from the Silicium simulation¹ using our method (Figure 7.8.(a)) with the method of [22] on a high resolution isosurface mesh (335,000 triangles) extracted using marching cubes (Figure 7.8.(b)) and by isocontouring zeros of the Gaussian extremality on the isosurface mesh using ParaView [1] based on the method presented in [126](Figure 7.8.(c)). However, the isocontouring method has the drawback that the ridge type is unknown (See Sec-

¹available at <http://www.volvis.org>

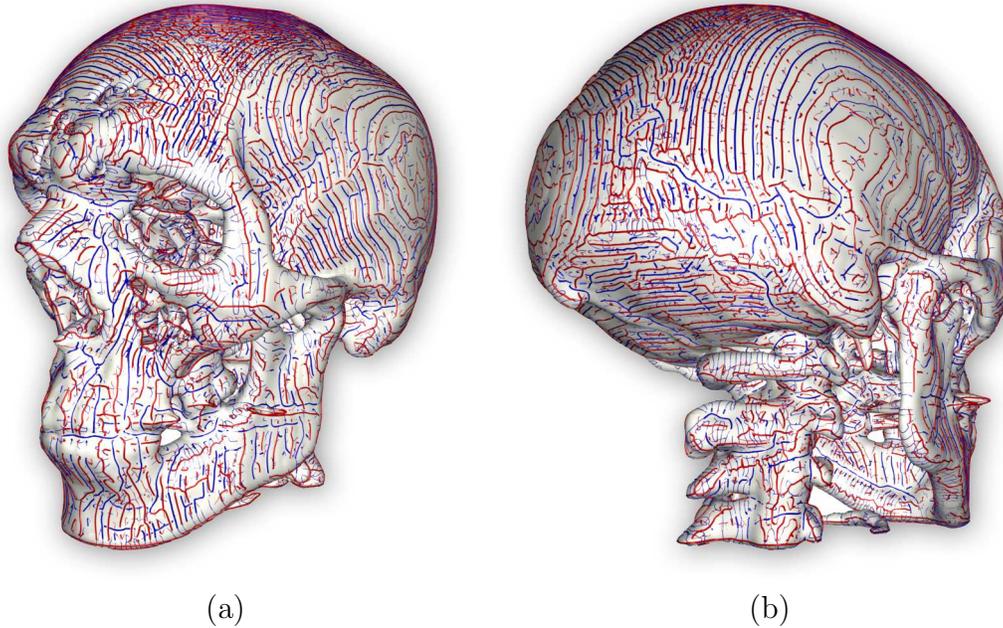


Figure 7.7. Ridges and crests extracted from an isosurface corresponding to a skull from a CT scan. κ_1 ridges are in blue and κ_2 ridges are in red. Crests are indicated by thicker curves.

tion 2.3). As shown in the close up views of part of the data set, ridges extracted using the proposed method are smoother than the ridges extracted using the method of [22], which are fragmented, have undesirable undulations and do not capture many ridges. The ridges extracted using the isocontouring method also have large undulations. In addition, the topology of the ridges extracted using the isocontouring approach is incorrect in many areas where κ_1 and κ_2 ridges cross each other, as noted in [22].

There has been recent impetus in the area of isogeometric engineering analysis [67], where partial differential equations are solved directly on CAD representations of objects by avoiding any conversion into conventional finite element representations such as hexahedral meshes that only approximate the CAD models. Simulation results are obtained directly on the CAD representations and are therefore more reliable. As an example, we solve linear elasticity equations on a cube represented as a trivariate B-spline to examine the vertical displacements resulting from loads applied at the top of the cube. The isosurface at a particular displacement value (that is now an implicit B-spline) identifies all locations within the volume that have the same vertical displacement.

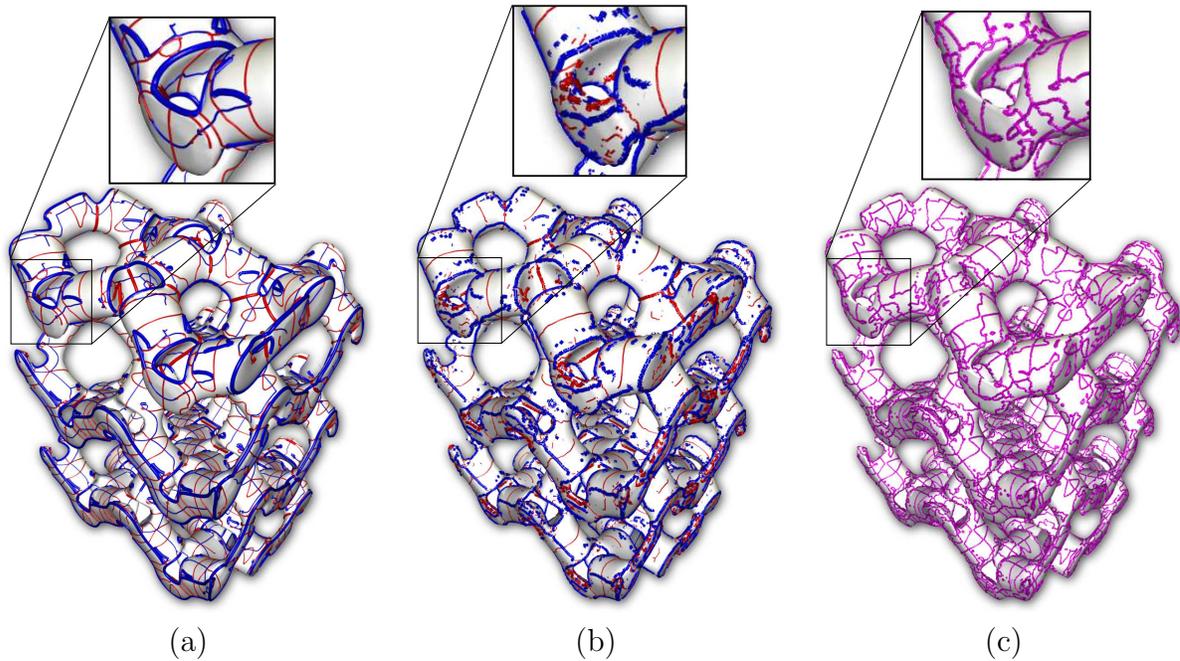


Figure 7.8. Ridges extracted from the level set of simulation data of the Silicon model using (a) the proposed approach; (b) method of [22] on isosurface mesh; (c) isocontouring method of [126] using Gaussian extremality. In (c) all ridges are shown in purple since the ridge type is unknown.

Ridges and crests are extracted directly from the higher order trivariate implicit B-spline representation of the isosurface to reveal additional structural information about the distribution of this vertical displacement within the cube. Crests indicate areas where there is a sharp change in the stress-strain relationship within the cube and may provide better insight for engineering analysis. Figure 7.9 (a) and (b) shows the results for the cube under slightly different vertical loads and the variation in geometry of the corresponding isosurfaces. While crests indicate major variations in the geometry of the two isosurfaces, the noncrest ridges also indicate higher order structural differences in the stress-strain relationships.

Figures 7.10 and 7.11 show all ridges extracted on an ellipsoid, a tangle surface and a smooth dodecahedron represented as algebraic functions. Exact trivariate Bézier representations of the algebraic functions are determined using the multivariate version of Marsden's identity [88]. The exact structure of ridges and umbilics on ellipsoids are well documented in the existing literature [102] and Figure 7.10 validates that our

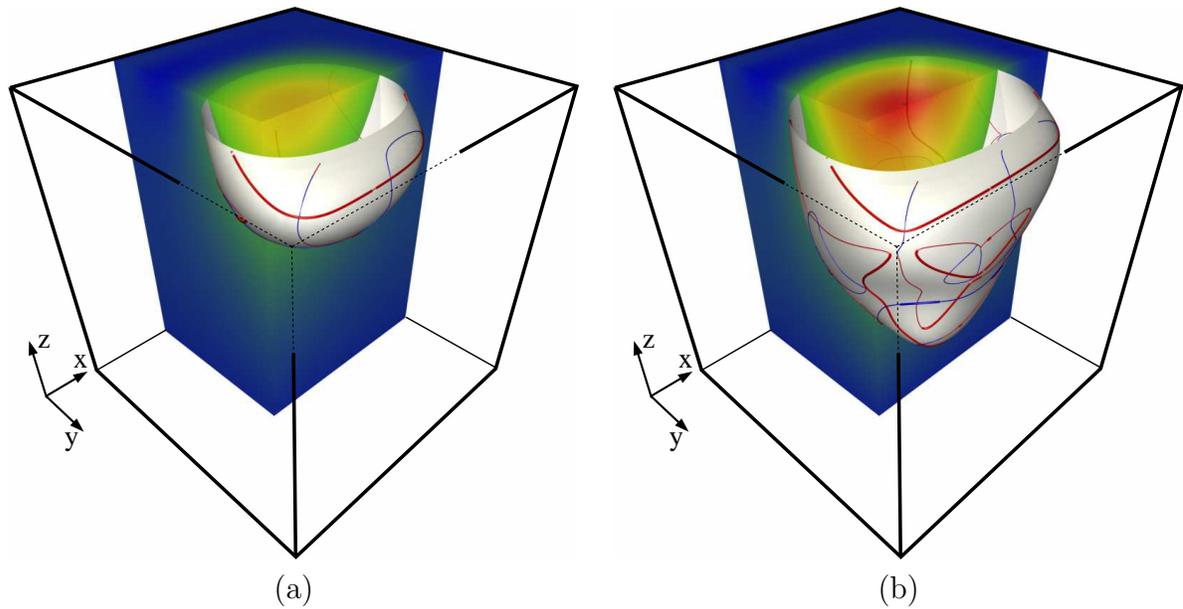


Figure 7.9. Ridges extracted from isosurfaces resulting from linear elasticity analysis on a cube represented as a trivariate B-spline under vertical loads. The loads in (a) and (b) vary slightly. The color map on part of the volume indicates the z displacement where blue indicates almost no displacement and red indicates larger displacement. The isosurfaces in both images correspond to z displacement value of $4e^{-5}$. The ridges and crests indicate regions where the interior of the volume with a specific z displacement changes sharply.

approach accurately extracts all ridges and umbilics on the ellipsoid. One of the partial derivatives of the algebraic function $f(x_1, x_2, x_3)$ is zero along each ridge and two of the partial derivatives are zero at the six poles. Similar issues are present on the tangle function and the smooth dodecahedron as well. In addition, the surface normals for both examples computed using local parameterizations given by the implicit function theorem do not always agree with the function gradient direction. The results show that our method is robust to both situations. Figure 7.11 (b) shows the different types of ridges on a smooth dodecahedron including crests, noncrest elliptic ridges and hyperbolic ridges. Figure 5 of Ohtake et al. [95] shows only the crests extracted from a polygonal representation of this surface which are similar to the crests shown here.

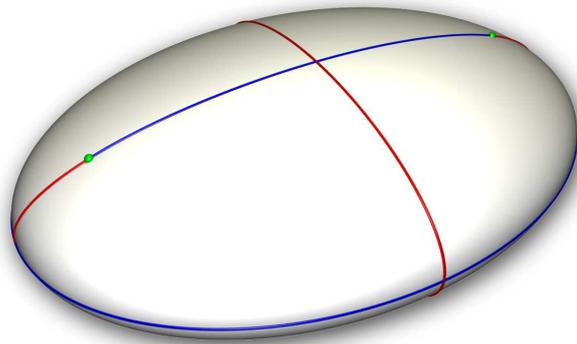


Figure 7.10. Ridges extracted using the proposed approach on an ellipsoid (algebraic function) represented with a single trivariate Bézier patch. Umbilic points are shown in green.

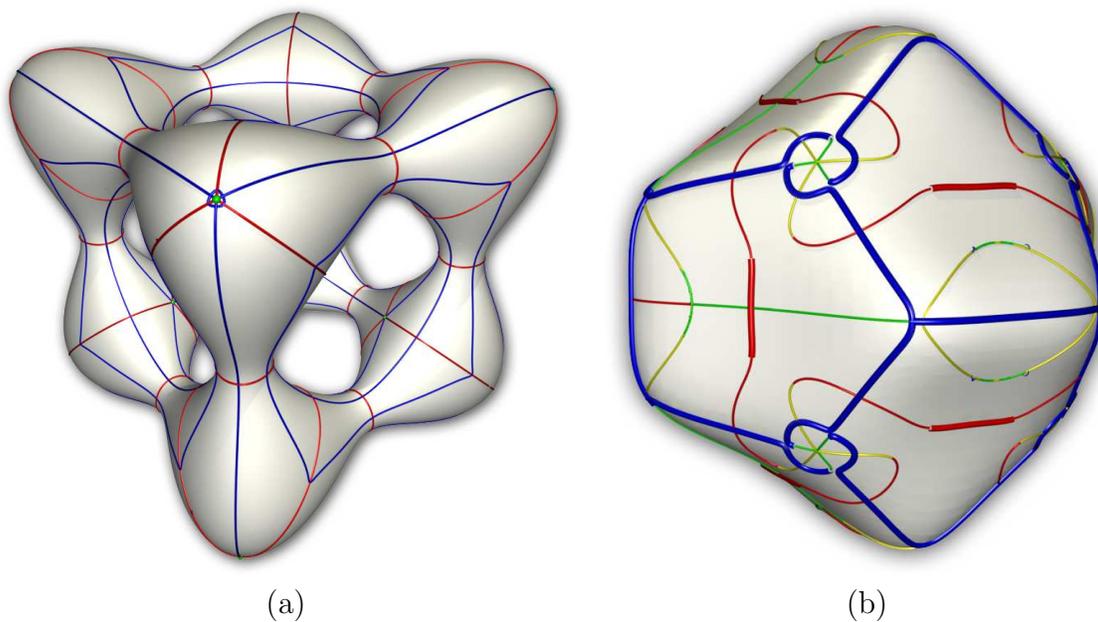


Figure 7.11. Ridges extracted using the proposed approach on algebraic functions represented with a single trivariate Bézier patch: (a) Tangle: $x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 11.8 = 0$. (b) Smooth dodecahedron: $x^6 + y^6 + z^6 + 20(x^4y^2 + y^4z^2 + z^4x^2) = 1$. Umbilic points are shown in green. In (b) thick blue and red curves correspond to crests, thin blue and red curves correspond to noncrest elliptic ridges, thin green and yellow correspond to hyperbolic ridges of κ_1 and κ_2 , respectively.

7.4 Geometrically Significant Regions Associated with Ridges

There can be a large number of ridge curves on a surface indicating complicated variations in curvature, although not all of them may be useful for a particular application. Therefore it is necessary to be able to distinguish between those that may be important or not depending on the target application. Typically, ridges that indicate stronger local geometric variation are more stable with respect to small surface deformations than others and may hence be more useful for shape analysis applications. Several approaches to defining ridge strength exist in the current literature. For example, it has been defined as the integral of the magnitude of the curvature along the ridge lines in [95]. The integral of the magnitude of the ridge functions were considered in [129] to create similar strength measures. Ridge sharpness has been defined as integral of the second derivative of the curvature in the principal direction in [22].

All of the earlier methods for quantifying the importance of ridges account for geometric properties only at ridge points, partly due to the fact that their computational framework required discrete surface representations. We present a new method, enabled by direction computation with smooth surface representations, that also considers neighborhoods of ridges to create measures of ridge significance. We define *salient regions* associated with major ridges (Section 7.4.1) to indicate geometrically significant regions on surfaces. Salient regions provide additional information for studying geometric variation of similar shapes, and are especially useful when the ridges themselves and geometric properties at ridge points do not provide sufficient information. We present an example of such a situation and show how the salient regions can be used for analyzing geometric variation.

7.4.1 Salience Boundaries and Salient Regions

Elliptic ridges (defined in Section 1.5) are identified in our work as major ridges and hyperbolic ridges are identified as secondary ridges. At a major ridge point, where κ_1 has a local maximum that is not a turning point, the corresponding principal curve is transverse to the ridge. By following the principal curve away from the ridge point, eventually a point is reached where κ_1 has a local minimum that is a point on a secondary

κ_1 -ridge. Along the curve between these points there is a point where the function κ_1 changes from concave downward to concave upward where κ_1 has an inflection point on the curve between the local maximum and minimum (at $t_1^T H_{\kappa_1} t_1 = 0$). There is a second inflection point reached by flowing along the principal curve from the major ridge point in the negative principal direction. We term these inflection points the *salience boundary points* of the corresponding major ridge point and we term the flow along the principal direction from the major ridges to the salience boundary points the *salience flow*. Salience boundary points of major ridges of κ_2 are identified in a similar manner by flowing along principal curves of κ_2 .

The *salience boundary for a major ridge* then is the collection of salience boundary points identified from the flow along the corresponding principal curves from the major ridges. The regions surrounding such a ridge curve and bounded by the salience boundaries defines *salient regions* associated with the major ridges. In order to compute salience boundaries and salient regions, principal curves are traced from each major ridge point on either side of the ridge using the method presented in [86]. The first inflection points on both sides, identified as the location where the sign of $t_i^T H_{\kappa_i} t_i$ changes, are marked as salience boundary points.

Figure 7.12 indicates major and secondary ridges on a surface. Figure 7.13 illustrates salient regions on the surface determined using the proposed definition. For elliptic ridges of κ_1 , salient regions are shown in green, while for elliptic ridges of κ_2 , the salient regions are shown in yellow. Narrower salience boundaries correspond to ridge points and surface regions that are more salient than others.

It is possible for multiple inflection points to exist between a major ridge and a secondary ridge along a principal curve. In this case, the first inflection points reached by flowing from the major ridge points are treated as the salience boundary points. In Figure 7.14 there are other inflection points that occur on the κ_2 principal curves between the magenta and red ridges and the number of such points is consistent with the number of sign changes of $t_2^T H_{\kappa_2} t_2$ between major and secondary ridges. Note that there are no additional ridges between the additional inflection points. This is validated in Figure 7.14 b) where the sampled ridge function for κ_2 is shown. In this image, the

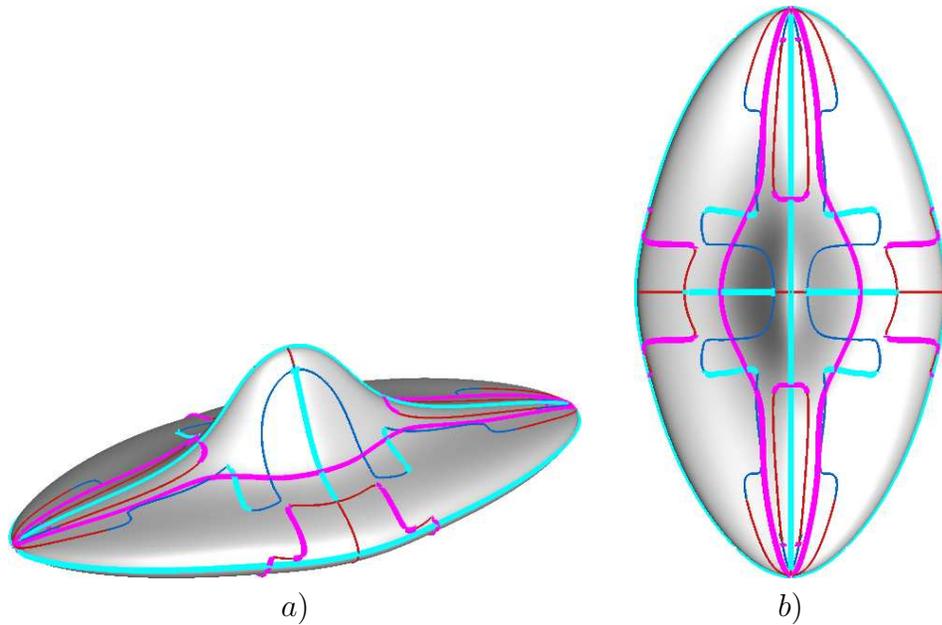


Figure 7.12. Major and secondary ridges of an object; a) side view, b) top view. Major ridges of κ_1 are shown in cyan and those of κ_2 are shown in magenta. Secondary ridges of κ_1 are shown in blue and those of κ_2 are shown in red.

intensity of the red color is higher in the regions closer to a κ_2 ridge.

7.4.2 Properties of Saliency Boundaries

With the exception of turning points on a ridge, the principal curve p_i corresponding to a principal curvature κ_i for that ridge is transverse to the corresponding ridge. In a small neighborhood of a nonturning point x_0 , generically, the principal curvature does not have critical points at inflection points, so the implicit function theorem implies that the inflection points form a regular differentiable curve. Generically this curve is transverse to the corresponding principal curves and is disjoint from the corresponding major ridge curve except at isolated points. These properties can fail in two distinct ways at isolated points. One is when the inflection point is degenerate (Figure 7.13), and then the curve of inflection points meets the corresponding major ridge (which ends there). The other is when the curve of inflection points is tangent to the principal curve at a point disjoint from the corresponding major ridge.

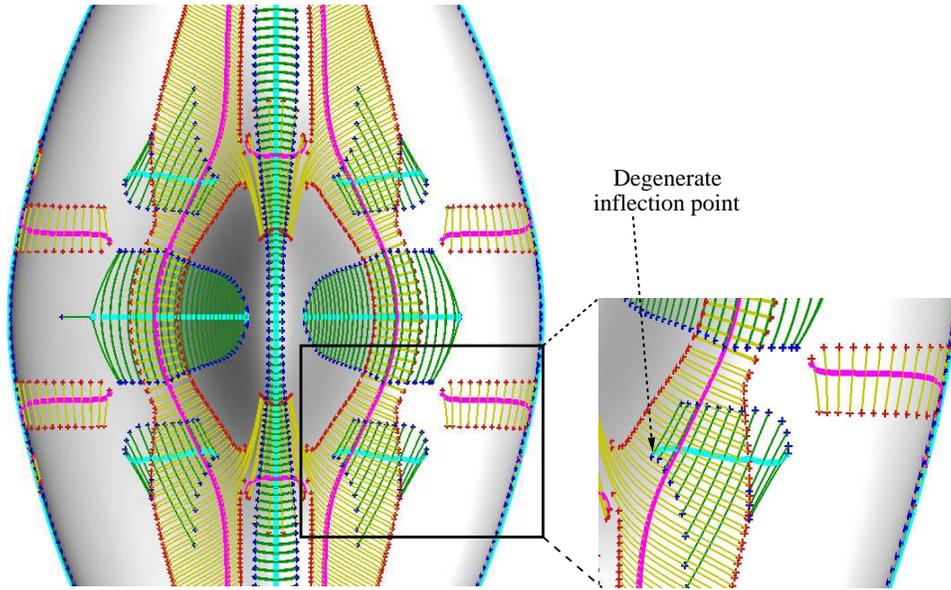


Figure 7.13. Saliency boundaries and salient regions associated with major ridges. For κ_1 , major ridges are shown in cyan, principal curve segments in salient regions are shown in green and inflection points are shown in dark blue. For κ_2 , major ridges are shown in magenta, principal curve segments in salient regions are shown in yellow and inflection points are shown in dark red. Also shown is a closeup of a region that indicates a degenerate inflection point.

7.4.3 Properties of Salient Regions

Consider a convex elliptic ridge. Suppose the principal curvature κ_1 is large. A large curvature corresponds to a small radius of curvature. If the ridge is part of a larger region, then this high curvature can only be maintained for a short time along the corresponding principal curve. Hence, the decrease must be rapid initially which then begins to decrease more gradually. This is where an inflection point occurs. Hence, the salient region is concentrated in a small region about the major ridge curve, as illustrated in Figure 7.13 at the cyan ridge in the center of the image where the surface is sharply curving. If instead the curvature is much smaller, then the decrease can be more gradual so the inflection point occurs much farther along the principal curve. Then, the salient region is much larger but changes more gradually. The cyan ridges on either side of the image center of Figure 7.13 illustrate this behavior. Also, in Figure 7.14 a) the saliency boundaries of κ_2 are further away from the magenta ridge since the curvature change is more gradual.

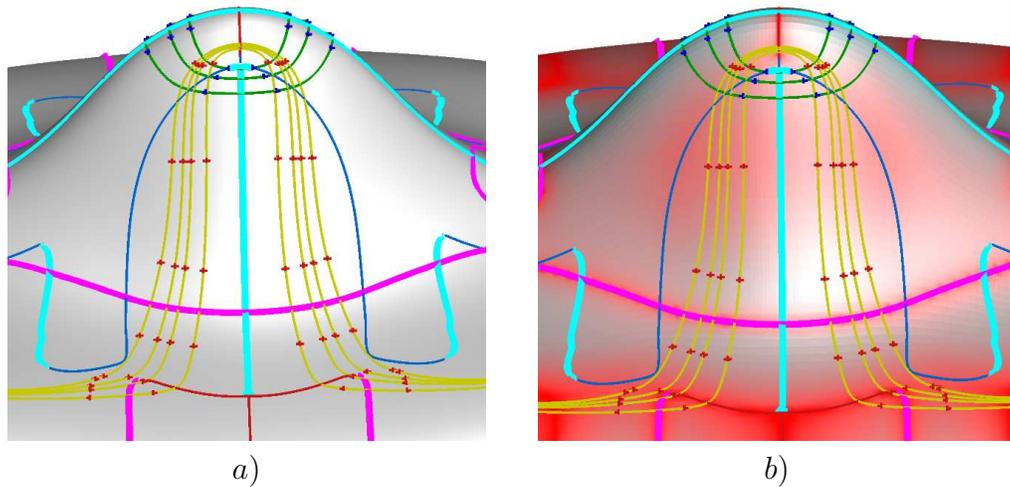


Figure 7.14. Occurrence of multiple inflection points between a major and secondary ridge even without existence of additional ridges in between them. a) Salience boundary points and other inflection points on principal curves traced from a few major ridge points. b) Results superimposed on surface indicating sampling of κ_2 ridge function. For κ_1 , major ridges are shown in cyan, principal curve segments in salient regions are shown in green and inflection points are shown in dark blue. For κ_2 , major ridges are shown in magenta, principal curve segments in salient regions are shown in yellow and inflection points are shown in dark red.

7.4.4 Analyzing Geometric Variation Using Salient Regions

Salient regions are an effective visualization tool for analyzing higher order geometric properties of surfaces. They are especially useful in distinguishing geometric properties of a population of similar objects when ridges on the surfaces occur at similar locations. Figure 7.15 shows front and back views of an object that is slightly asymmetric. The major and secondary ridges are slightly different on the front and back sides of the model but do not clearly distinguish the differences. In particular, the major ridge running along the bump of the model is almost identical on both sides. In this case the salient regions clearly indicate the geometric differences and enable quantitative evaluation of the differences. The differences are clearly visible from the top view of the object as shown in Figure 7.16.

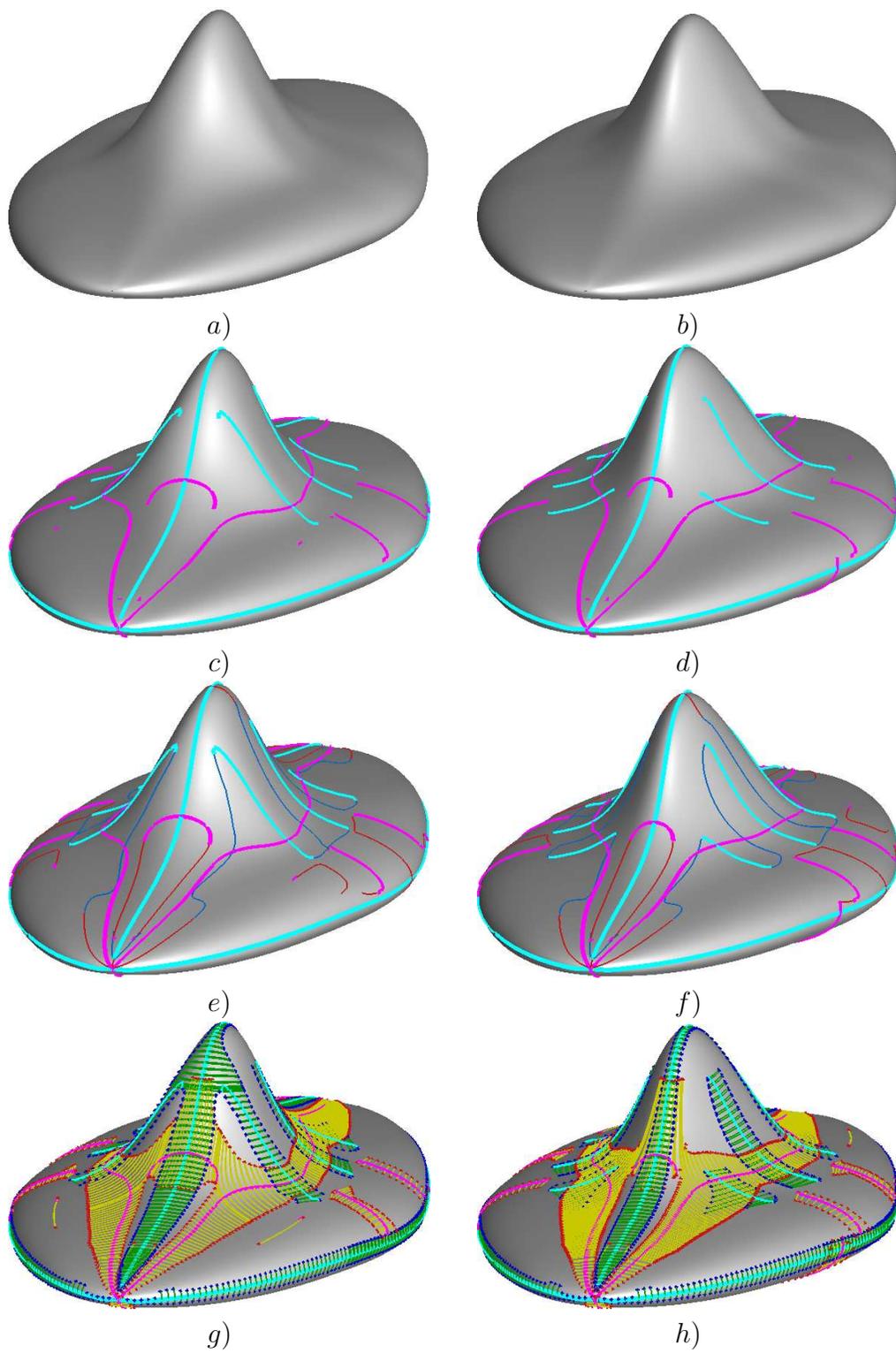


Figure 7.15. Visualizing geometric differences using salient regions. a) front view and, b) back view of a slightly asymmetric object, c) and d) major ridges only, e) and f) all ridges, g) and h) salient regions.

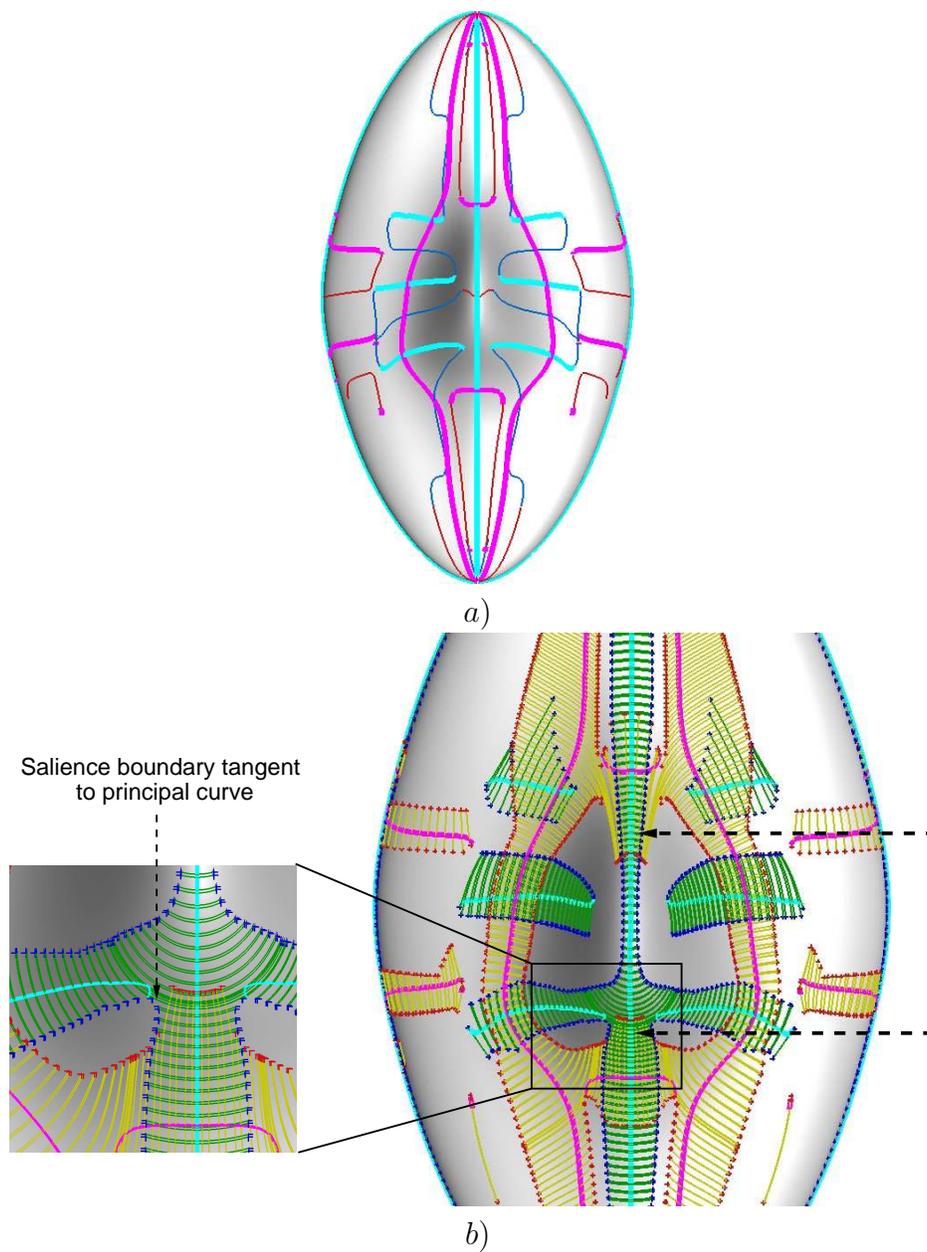


Figure 7.16. Visualizing geometric differences using salient regions. a) the major ridge corresponding to the bump in the surface lies in the center on both asymmetric regions, other major ridges are slightly different across plane of symmetry, b) salient regions clearly indicate geometric differences in the asymmetric regions pointed by the arrows. Also shown is a close up view of a region that contains a point where the salience boundary is tangent to the principal curve.

CHAPTER 8

CONCLUSIONS

The medial axis was conceptualized by Blum in the late 1960s and has since become an important geometric structure for the analysis of objects in a variety of geometric processing applications. Although mathematical properties of the medial axis are well documented in the existing literature, before completion of this body of research, state of the art techniques had been able to compute only pieces of medial representations for discrete approximations of objects, and without topological structure. This can be attributed in part due to the complexity in structure and inherent nonlinearity of the medial axis [113]. Considerable human interaction is required in order to infer topological information. Discrete approximations of smooth geometry introduce artifacts that are not part of the medial axis of regions bounded by smooth surfaces and considerable effort is required to remove them. The few techniques that use smooth representations only compute partial solutions for a restricted set of objects.

This dissertation proposes a new higher order methodology, based in part on results from both B-spline theory and singularity theory, that automatically computes medial axes of objects, in 2D as well as 3D, directly from smooth geometry representations. The computed medial axes are accurate up to user specified accuracy and include correct structural information. Since smooth geometry representations are used, the method does not generate nonmedial artifacts. In 2D, this dissertation presents an approach to compute medial axes of planar regions bounded by closed parametric B-spline curves. In 3D, this dissertation presents an approach to compute medial axes of volumetric regions bounded by closed parametric B-spline surfaces. B-splines are extensively used for representing a large variety of geometric data. The B-spline representation is the de-facto standard for designed models in the field of computer-aided design

(CAD). Therefore, the computational techniques presented in this dissertation are widely applicable.

Accurate and automated computation of topological structure opens new avenues and methods for analyzing geometry and structure. We present a graph-theoretic approach for analyzing the structure of 3D medial axes, that in turn is useful for shape analysis of objects by enriching the surface representation with structural and inherent geometric information. We also present a new technique for structural analysis of 3D objects using scalar functions defined on their surfaces. We compute 2D medial axes for an ordered set of slices of an object obtained as level sets of the defined scalar function. When the medial axes of the slices are connected, they form a skeleton-like structure that presents structural and geometric information. We present a computational procedure for this approach for 3D polygonal objects and demonstrate several examples. Several applications that utilize geometric information could benefit from the proposed approaches.

Edge curves of the medial axis correspond to a subset of ridges on a surface. Ridges have been found to be an important geometric feature in their own right, and have been used in several shape analysis applications. We present a new algorithm to extract ridges directly from smooth parametric surfaces. We also extended that approach to compute ridges from isosurfaces of volumetric data (3D medical images, simulation results) using smooth implicit B-spline representations. The resulting ridges are shown to have guaranteed accuracy and improved quality. Computing with smooth representations also enables a new method for further analysis of intrinsic geometric properties of surface not only at ridge points, but also in their neighborhoods. We introduce a new definition of salient regions and demonstrate its usefulness in shape analysis. Region-based shape descriptors are gaining emphasis for anatomical shape analysis to overcome the limitations of point-based descriptors [6] and the salient regions defined here may be used for such applications.

We present several examples where the genericity assumption is relaxed and address the manifestation of nongeneric behavior in their 3D medial axes. The examples include objects with sharp corners, sharp crease edges and situations where transition

points occur simultaneously. Other types of nongeneric situations can also occur for multisurface models. Regions bounded by symmetrical entities such as cylindrical, spherical or pairwise planar and opposite surfaces will create degenerate situations in the medial axis. It is necessary to effectively address such nongeneric situations to apply this method to such models.

The computation time is limited by the root finding steps in the presented techniques. There is still much scope for improving the performance of equation solvers in general. A parallel processing approach is one option that may have potential in addressing high dimensional systems. In this dissertation, we present specialized hierarchical and parallel processing approaches to make the approach more practical. These ideas could be extended further for addressing B-spline root solving in general.

This dissertation presents a dynamic computational framework, in conjunction with powerful B-spline based tools, for geometric problems whose solutions are higher dimensional entities, such as curves or surfaces, instead of isolated points. By identifying special critical points and tracking the evolution of the solution between them enables computation of the topological structure, thereby giving connected higher dimensional elements. This approach can be applied to other types of geometric problems whose solutions are higher dimensional entities.

REFERENCES

- [1] ParaView. <http://www.paraview.org>.
- [2] AICHHOLZER, O., AIGNER, W., AURENHAMMER, F., HACKL, T., JUTTNER, B., AND RABL, M. Medial axis computation for planar free-form shapes. *Computer-Aided Design* 41, 5 (2009), 339–349.
- [3] AMENTA, N., CHOI, S., AND KOLLURI, R. The power crust. In *Proceedings of 6th ACM Symposium on Solid Modeling* (2001), pp. 249–260.
- [4] AMENTA, N., CHOI, S., AND KOLLURI, R. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications* 19, 2-3 (2001), 127–153.
- [5] ATTALI, D., BOISSONNAT, J., AND EDELSBRUNNER, H. Stability and computation of medial axes: A state-of-the-art report. *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration. Springer-Verlag, Mathematics and Visualization* (2007).
- [6] AWATE, S., YUSHKEVICH, P., SONG, Z., LICHT, D., AND GEE, J. Cerebral cortical folding analysis with multivariate modeling and testing: Studies on gender differences and neonatal development. *NeuroImage* (2010).
- [7] BAJA, G., AND SVENSSON, S. Surface Skeletons Detected on the D6 Distance Transform. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition* (2000), Springer-Verlag, pp. 387–396.
- [8] BARNHILL, R., FARIN, G., JORDAN, M., AND PIPER, B. Surface/surface intersection. *Computer Aided Geometric Design* 4, 1-2 (1987), 3–16.
- [9] BELYAEV, A., BOGAEVSKI, I., AND KUNII, T. Principal direction ridges. Tech. rep., Technical Report 96-4-001, Center for Mathematical Sciences, The University of Aizu, Japan, 1996.
- [10] BELYAEV, A., PASKO, A., AND KUNII, T. Ridges and ravines on implicit surfaces. In *Computer Graphics International, 1998. Proceedings* (1998), pp. 530–535.
- [11] BIASOTTI, S., ATTALI, D., BOISSONNAT, J., EDELSBRUNNER, H., ELBER, G., MORTARA, M., BAJA, G., SPAGNUOLO, M., TANASE, M., AND VELTKAMP, R. Skeletal structures. *Shape Analysis and Structuring* (2008), 145–183.

- [12] BLANDING, R., BROOKING, C., GANTER, M., AND STORTI, D. A skeletal-based solid editor. In *Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications* (1999), ACM, p. 150.
- [13] BLUM, H., ET AL. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form* 19, 5 (1967), 362–380.
- [14] BOGAEVSKI, I., LANG, V., BELYAEV, A., AND KUNII, T. Color ridges on implicit polynomial surfaces. *GraphiCon 2003, Moscow, Russia* (2003).
- [15] BOOTHBY, W. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press Inc, 1986.
- [16] BORGEFORS, G., NYSTRÖM, I., AND DI BAJA, G. S. Computing skeletons in three dimensions. *Pattern Recognition* 32, 7 (1999), 1225–1236.
- [17] BRANDT, J., AND ALGAZI, V. Continuous skeleton computation by Voronoi diagram* 1. *CVGIP: Image Understanding* 55, 3 (1992), 329–338.
- [18] BREEN, D., AND WHITAKER, R. A Level-Set Approach for the Metamorphosis of Solid Models. *IEEE Transactions on Visualization and Computer Graphics* 7, 2 (2001), 192.
- [19] CAZALS, F., FAUGÈRE, J., POUGET, M., AND ROUILLIER, F. Topologically certified approximation of umbilics and ridges on polynomial parametric surface. INRIA Technical Report 2005.
- [20] CAZALS, F., FAUGÈRE, J., POUGET, M., AND ROUILLIER, F. The implicit structure of ridges of a smooth parametric surface. *Computer Aided Geometric Design* 23, 7 (2006), 582–598.
- [21] CAZALS, F., FAUGÈRE, J., POUGET, M., AND ROUILLIER, F. Ridges and umbilics of polynomial parametric surfaces. *Geometric Modeling and Algebraic Geometry* (2007), 141–159.
- [22] CAZALS, F., AND POUGET, M. Topology driven algorithms for ridge extraction on meshes. INRIA Technical Report 2005.
- [23] CHANG, M., LEYMARIE, F., AND KIMIA, B. 3D Shape Registration using Regularized Medial Scaffolds. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium* (2004), IEEE Computer Society, p. 994.
- [24] CHAZAL, F., AND LIEUTIER, A. Stability and homotopy of a subset of the medial axis. In *SM '04: Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 243–248.
- [25] CHEN, C., AND FREEDMAN, D. Measuring and computing natural generators for homology groups. *Computational Geometry* (2009).

- [26] CHEN, X. *An Application of Singularity Theory to Robust Geometric Calculation of Interactions among Dynamically Deforming Geometric Objects*. PhD thesis, University of Utah, 2008.
- [27] CHEN, X., RIESENFELD, R. F., COHEN, E., AND DAMON, J. Theoretically-based algorithms for robustly tracking intersection curves of deforming surfaces. *Computer-Aided Design* 39, 5 (2007), 389–397.
- [28] CHOU, J. Voronoi Diagrams for Planar Shapes. *IEEE Computer Graphics and Applications* 15, 2 (1995), 52–59.
- [29] COHEN, E., RIESENFELD, R., AND ELBER, G. *Geometric Modeling with Splines: An Introduction*. AK Peters, Ltd., 2001.
- [30] COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ, S. Where do people draw lines? *ACM Transactions on Graphics (Proc. SIGGRAPH)* 27, 3 (Aug. 2008).
- [31] CULVER, T., KEYSER, J., AND MANOCHA, D. Accurate computation of the medial axis of a polyhedron. In *Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications* (1999), ACM New York, NY, USA, pp. 179–190.
- [32] DAMON, J. Smoothness and geometry of boundaries associated to skeletal structures I: Sufficient conditions for smoothness. *Annales de l’Institut Fourier* 53, 6 (2003), 1941–1985.
- [33] DAMON, J. Determining the geometry of boundaries of objects from medial data. *International Journal of Computer Vision* 63, 1 (2005), 45–64.
- [34] DAMON, J. The global medial structure of regions in \mathbb{R}^3 . *Geometry and Topology* 10 (2006), 2385–2429.
- [35] DAMON, J. Tree Structure for Contractible Regions in \mathbb{R}^3 . *International Journal of Computer Vision* 74, 2 (2007), 103–116.
- [36] DE VERDIÈRE, É., AND LAZARUS, F. Optimal system of loops on an orientable surface. *Discrete and Computational Geometry* 33, 3 (2005), 507–534.
- [37] DEY, T., AND GOSWAMI, S. Tight cocone: a water-tight surface reconstructor. *Journal of Computing and Information Science in Engineering* 3 (2003), 302.
- [38] DEY, T., AND ZHAO, W. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica* 38, 1 (2003), 179–200.
- [39] DEY, T., AND ZHAO, W. Approximate medial axis as a voronoi subcomplex. *Computer-Aided Design* 36, 2 (2004), 195–202.
- [40] EBERLY, D. *Ridges in Image and Data Analysis*. Kluwer Academic Pub, 1996.
- [41] ELBER, G. The IRIT modeling environment, version 10.0, 2008.

- [42] ELBER, G., COHEN, E., AND DRAKE, S. MATHSM: Medial axis transform toward high speed machining of pockets. *Computer-Aided Design* 37, 2 (2005), 241–250.
- [43] ELBER, G., AND GRANDINE, T. Efficient solution to systems of multivariate polynomials using expression trees. In *IEEE International Conference on Shape Modeling and Applications, 2008. SMI 2008* (2008), pp. 163–169.
- [44] ELBER, G., AND KIM, M. Computing rational bisectors. *IEEE Computer Graphics and Applications* (1999), 76–81.
- [45] ELBER, G., AND KIM, M. Geometric constraint solver using multivariate rational spline functions. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications* (2001), ACM New York, NY, USA, pp. 1–10.
- [46] ERICKSON, J., AND WHITTLESEY, K. Greedy optimal homotopy and homology generators. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2005), Society for Industrial and Applied Mathematics, p. 1046.
- [47] FOL-LEYMARIE, F. *Three-dimensional Shape Representation via Shock Flows*. PhD thesis, Brown University, Providence, RI, USA, 2003.
- [48] FOSKEY, M., GARBER, M., LIN, M., AND MANOCHA, D. A Voronoi-based hybrid planner. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2001), vol. 1, pp. 55–60.
- [49] FOSKEY, M., LIN, M., AND MANOCHA, D. Efficient computation of a simplified medial axis. *Journal of Computing and Information Science in Engineering* 3 (2003), 274.
- [50] GIBLIN, P., AND KIMIA, B. On the local form and transitions of symmetry sets, medial axes, and shocks. *International Journal of Computer Vision* 54, 1 (2003), 143–157.
- [51] GIBLIN, P., AND KIMIA, B. A formal classification of 3D medial axis points and their local geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 2 (2004), 238–251.
- [52] GIBLIN, P., KIMIA, B., AND POLLITT, A. Transitions of the 3D medial axis under a one-parameter family of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2009), 900–918.
- [53] GIESEN, J., MIKLOS, B., AND PAULY, M. Medial axis approximation of planar shapes from union of balls: A simpler and more robust algorithm. In *Proceedings of the Canadian Conference on Computational Geometry* (2008), pp. 105–108.
- [54] GOLDMAN, R. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design* 22, 7 (2005), 632–658.

- [55] GOTSMAN, C., KALIGOSI, K., MEHLHORN, K., MICHAIL, D., AND PYRGA, E. Cycle bases of graphs and sampled manifolds. *Computer Aided Geometric Design* 24, 8-9 (2007), 464–480.
- [56] GUÉZIEC, A. Large deformable splines, crest lines and matching. In *Proceedings of the Fourth International Conference on Computer Vision* (1993), pp. 650–657.
- [57] GÜRSOY, H., AND PATRIKALAKIS, N. An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part II implementation. *Engineering with Computers* 8, 4 (1992), 179–196.
- [58] GÜRSOY, H., AND PATRIKALAKIS, N. Automatic coarse and fine surface mesh generation scheme based on medial axis transform: part I algorithms. *Engineering with Computers* 8, 3 (1992), 121–137.
- [59] HALLINAN, P., GORDON, G., YUILLE, A., GIBLIN, P., AND MUMFORD, D. *Two and Three-dimensional Patterns of the Face*. AK Peters, Ltd. Natick, MA, USA, 1999.
- [60] HANNIEL, I., MUTHUGANAPATHY, R., ELBER, G., AND KIM, M. Precise Voronoi cell extraction of free-form rational planar closed curves. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling* (2005), ACM, p. 59.
- [61] HASTINGS, D., DUNBAR, P., ELPHINGSTONE, G., BOOTZ, M., MURAKAMI, H., MARUYAMA, H., MASAHARU, H., HOLLAND, P., PAYNE, J., BRYANT, N., ET AL. The global land one-kilometer base elevation (GLOBE) digital elevation model. Version 1.0. National Oceanic and Atmospheric Administration, National Geophysical Data Center, Boulder, Colorado. *National Geophysical Data Center, Digital data base on the World Wide Web (URL: <http://www.ngdc.noaa.gov/mgg/topo/globe.html>) and CD-ROMs.* (1999).
- [62] HELD, M. VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Computational Geometry* 18, 2 (2001), 95–123.
- [63] HILDEBRANDT, K., POLTHIER, K., AND WARDETZKY, M. Smooth feature lines on surface meshes. In *Symposium on geometry processing* (2005), pp. 85–90.
- [64] HOFFMAN, C. How to construct the skeleton of CSG objects. In *Proceedings of the Mathematics of Surfaces* (1990), Oxford Univ. Press.
- [65] HORTON, J. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing* 16 (1987), 358.
- [66] HOSAKA, M. *Modeling of Curves and Surfaces in CAD/CAM*. Springer, 1992.
- [67] HUGHES, T., COTTRELL, J., AND BAZILEVS, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194, 39-41 (2005), 4135–4195.

- [68] INTERRANTE, V., FUCHS, H., AND PIZER, S. Enhancing transparent skin surfaces with ridge and valley lines. In *Proceedings of the 6th Conference on Visualization* (1995), IEEE Computer Society Washington, DC, USA.
- [69] JEFFERIES, M. Extracting Crest Lines from B-spline Surfaces. Master's thesis, Arizona State University, U.S.A, 2002.
- [70] JOHNSON, D., AND COHEN, E. An improved method for haptic tracing of sculptured surfaces. In *Symp. on Haptic Interfaces, ASME International Mechanical Engineering Congress and Exposition, Anaheim, CA* (1998).
- [71] JOSHI, S., PIZER, S., FLETCHER, P., YUSHKEVICH, P., THALL, A., AND MARRON, J. Multiscale deformable model segmentation and statistical shape analysis using medial descriptions. *IEEE Transactions on Medical Imaging* 21, 5 (2002), 538–550.
- [72] KAVITHA, T., LIEBCHEN, C., MEHLHORN, K., MICHAIL, D., RIZZI, R., UECKERDT, T., AND ZWEIG, K. Cycle bases in graphs characterization, algorithms, complexity, and applications. *Computer Science Review* 3, 4 (2009), 199–243.
- [73] KENT, J., MARDIA, K., AND WEST, J. Ridge curves and shape analysis. In *The British Machine Vision Conference 1996* (1996), pp. 43–52.
- [74] KIM, S., AND KIM, C. Finding ridges and valleys in a discrete surface using a modified MLS approximation. *Computer-Aided Design* 37, 14 (2005), 1533–1542.
- [75] KO, K., MAEKAWA, T., PATRIKALAKIS, N., MASUDA, H., AND WOLTER, F. Shape intrinsic fingerprints for free-form object matching. In *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications* (2003), ACM New York, NY, USA, pp. 196–207.
- [76] KOENDERINK, J. *Solid Shape*. MIT Press Cambridge, MA, USA, 1990.
- [77] LAM, L., LEE, S., AND SUEN, C. Thinning methodologies—a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 9 (1992), 869–885.
- [78] LAZARUS, F., AND VERROUST, A. Level set diagrams of polyhedral objects. In *Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications* (1999), ACM, pp. 130–140.
- [79] LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLO T, J. Least squares conformal maps for automatic texture atlas generation. In *ACM SIGGRAPH Conference Proceedings* (Jul 2002), ACM, Ed.
- [80] LEYMARIE, F., AND KIMIA, B. Computation of the shock scaffold for unorganized point clouds in 3D. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2003), vol. 1, IEEE Computer Society; 1999.

- [81] LEYMARIE, F., AND KIMIA, B. The medial scaffold of 3d unorganized point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2007), 313–330.
- [82] LIEUTIER, A. Any open bounded subset of \mathbb{R}^n has the same homotopy type as its medial axis. *Computer-Aided Design* 36, 11 (2004), 1029–1046.
- [83] LITTLE, J., AND SHI, P. Structural lines, TINs, and DEMs. *Algorithmica* 30, 2 (2001), 243–263.
- [84] MA, K., AND INTERRANTE, V. Extracting feature lines from 3D unstructured grids. In *Proceedings of the 8th Conference on Visualization* (1997), pp. 285–292.
- [85] MAEKAWA, T., AND PATRIKALAKIS, N. Interrogation of differential geometry properties for design and manufacture. *The Visual Computer* 10, 4 (1994), 216–237.
- [86] MAEKAWA, T., WOLTER, F., AND PATRIKALAKIS, N. Umbilics and lines of curvature for shape interrogation. *Computer Aided Geometric Design* 13, 2 (1996), 133–161.
- [87] MARSCHNER, S. R., AND LOBB, R. J. An evaluation of reconstruction filters for volume rendering. In *Proceedings of the Conference on Visualization* (Los Alamitos, CA, USA, 1994), IEEE Computer Society Press, pp. 100–107.
- [88] MARSDEN, M. J. An identity for spline functions with applications to variation diminishing spline approximation. *J. Approx. Theory* 3 (1970), 7–49.
- [89] MATHER, J. Distance to a manifold in euclidean space. In *Proceedings of the Symposium on Pure Mathematics 40 (part II)* (1983), American Mathematical Society, pp. 199–216.
- [90] MAXWELL, J. L. On hills and dales. *Philosophical Magazine Series 4* 40, 269 (1870), 421–427.
- [91] MIKLOS, B., GIESEN, J., AND PAULY, M. Discrete scale axis representations for 3d geometry. *ACM Trans. Graph.* 29 (July 2010), 101:1–101:10.
- [92] MONGA, O., AND BENAYOUN, S. Using partial derivatives of 3D images to extract typical surface features. *Computer Vision and Image Understanding* 61, 2 (1995), 171–189.
- [93] MORRIS, R. *Symmetry of Curves and the Geometry of Surfaces*. PhD thesis, University of Liverpool, 1990.
- [94] OGNIEWICZ, R., AND ILG, M. Voronoi skeletons: Theory and applications. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on* (1992), IEEE, pp. 63–69.
- [95] OHTAKE, Y., BELYAEV, A., AND SEIDEL, H. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics* 23, 3 (2004), 609–612.

- [96] O'NEILL, B. *Elementary Differential Geometry*, 2nd ed. Academic Press, 2006.
- [97] PATRIKALAKIS, N., AND MAEKAWA, T. *Shape Interrogation for Computer-aided Design and Manufacturing*. Springer, 2002.
- [98] PENNEC, X., AYACHE, N., AND THIRION, J.-P. Landmark-based registration using features identified through differential geometry. In *Handbook of Medical Image Processing and Analysis - New edition*, I. Bankman, Ed. Academic Press, Dec. 2008, ch. 34, pp. 565–578.
- [99] PIZER, S., FLETCHER, P., JOSHI, S., THALL, A., CHEN, J., FRIDMAN, Y., FRITSCH, D., GASH, A., GLOTZER, J., JIROUTEK, M., ET AL. Deformable m-reps for 3D medical image segmentation. *International Journal of Computer Vision* 55, 2 (2003), 85–106.
- [100] PIZER, S., SIDDIQI, K., SZEKELY, G., DAMON, J., AND ZUCKER, S. Multiscale Medial Loci and Their Properties. *International Journal of Computer Vision* 55, 2/3 (2003), 155–179.
- [101] PORTEOUS, I. The normal singularities of a submanifold. *Journal of Differential Geometry* 5 (1971), 543–564.
- [102] PORTEOUS, I. *Geometric Differentiation for the Intelligence of Curves and Surfaces*. Cambridge University Press, 2001.
- [103] RAMAMURTHY, R., AND FAROUKI, R. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. Theoretical foundations. *Journal of Computational and Applied Mathematics* 102, 1 (1999), 119–141.
- [104] RAMAMURTHY, R., AND FAROUKI, R. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries II: Detailed algorithm description. *Journal of Computational and Applied Mathematics* 102, 2 (1999), 253–277.
- [105] RAMANATHAN, M., AND GURUMOORTHY, B. Constructing medial axis transform of planar domains with curved boundaries. *Computer-Aided Design* 35, 7 (2003), 619–632.
- [106] RAMANATHAN, M., AND GURUMOORTHY, B. A tracing algorithm for constructing medial axis transform of 3D objects bound by free-form surfaces. In *Proceedings of the International Conference on Shape Modeling and Applications 2005* (2005), IEEE Computer Society, p. 237.
- [107] RAMANATHAN, M., AND GURUMOORTHY, B. Constructing medial axis transform of extruded and revolved 3D objects with free-form boundaries. *Computer-Aided Design* 37, 13 (2005), 1370–1387.
- [108] RENIERS, D., AND TELEA, A. Patch-type segmentation of voxel shapes using simplified surface skeletons. In *Computer Graphics Forum* (2008), vol. 27, Blackwell Science Ltd, Osney Mead, Oxford, OX 2 0 EL, UK,, pp. 1837–1834.

- [109] RENIERS, D., AND TELEA, A. Segmenting simplified surface skeletons. *Lecture Notes in Computer Science 4992* (2008), 262.
- [110] SCHMITT, M. Some examples of algorithms analysis in computational geometry by means of mathematical morphological techniques. *Geometry and Robotics* (1989), 225–246.
- [111] SCHREINER, J., SCHEIDEGGER, C., AND SILVA, C. High quality extraction of isosurfaces from regular and irregular grids. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2006)* 12, 5 (September-October 2006), 1205–1212.
- [112] SEONG, J., COHEN, E., AND ELBER, G. Voronoi diagram computations for planar NURBS curves. In *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling* (2008), ACM, pp. 67–77.
- [113] SHEEHY, D., ARMSTRONG, C., AND ROBINSON, D. Shape Description By Medial Surface Construction. *IEEE Transactions on Visualization and Computer Graphics* 2, 1 (1996), 62–72.
- [114] SHEFFER, A., LÉVY, B., MOGILNITSKY, M., AND BOGOMYAKOV, A. Abf++: fast and robust angle based flattening. *ACM Trans. Graph.* 24 (April 2005), 311–330.
- [115] SHERBROOKE, E., PATRIKALAKIS, N., AND BRISSON, E. An algorithm for the medial axis transform of 3d polyhedral solids. *IEEE Transactions on Visualization and Computer Graphics* (1996), 44–61.
- [116] SHERBROOKE, E., PATRIKALAKIS, N., AND WOLTER, F. Differential and topological properties of medial axis transforms. *Graphical Models and Image Processing* 58, 6 (1996), 574–592.
- [117] SIDDIQI, K., BOUIX, S., TANNENBAUM, A., AND ZUCKER, S. Hamilton-Jacobi skeletons. *International Journal of Computer Vision* 48, 3 (2002), 215–231.
- [118] SIDDIQI, K., AND PIZER, S. *Medial Representations: Mathematics, Algorithms and Applications*. Springer Verlag, 2008.
- [119] SIDDIQI, K., ZHANG, J., MACRINI, D., SHOKOUFANDEH, A., BOUIX, S., AND DICKINSON, S. Retrieving articulated 3-D models using medial surfaces. *Machine Vision and Applications* 19, 4 (2008), 261–275.
- [120] SOLDEA, O. Global segmentation and curvature analysis of volumetric data sets using trivariate b-spline functions. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 2 (2006), 265–278. Member-Elber, Gershon and Member-Rivlin, Ehud.
- [121] STYLIANOU, G., AND FARIN, G. Crest lines extraction from 3D triangulated meshes. *Hierarchical and Geometrical Methods in Scientific Visualization* (2003), 269–281.

- [122] SUBSOL, G. Crest lines for curve-based warping. *Brain Warping* (1999), 241–262.
- [123] SUD, A., FOSKEY, M., AND MANOCHA, D. Homotopy-preserving medial axis simplification. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling* (2005), ACM, pp. 39–50.
- [124] TASDIZEN, T., AND WHITAKER, R. Feature preserving variational smoothing of terrain data. In *Proceedings of the Second IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision (VLSM'03)(Institute of Electrical and Electronics Engineers, 2003)*, pp. 121–128.
- [125] TEWARI, G., GOTSMAN, C., AND GORTLER, S. Meshing genus-1 point clouds using discrete one-forms. *Computers & Graphics* 30, 6 (2006), 917–926.
- [126] THIRION, J. The extremal mesh and the understanding of 3D surfaces. *International Journal of Computer Vision* 19, 2 (1996), 115–128.
- [127] THIRION, J., AND GOURDON, A. The 3D marching lines algorithm. *Graphical Models and Image Processing* 58, 6 (1996), 503–509.
- [128] WRIGHT, M., CIPOLLA, R., AND GIBLIN, P. Skeletonization using an extended Euclidean distance transform. *Image and vision computing* 13, 5 (1995), 367–375.
- [129] YOSHIKAWA, S., BELYAEV, A., YOKOTA, H., AND SEIDEL, H. Fast and faithful geometric algorithm for detecting crest lines on meshes. In *Proceedings of 15th Pacific Conference on Computer Graphics and Applications* (2007), pp. 231–237.
- [130] ZHANG, Y., AND WANG, P. Analytical comparison of thinning algorithms. *International Journal of Pattern Recognition and Artificial Intelligence* 7, 5 (1993), 1227–1246.