Washington University in St. Louis Washington University Open Scholarship

Engineering and Applied Science Theses & Dissertations

McKelvey School of Engineering

Spring 5-15-2018

Medial Axis Approximation and Regularization

Yajie Yan Washington University in St. Louis

Follow this and additional works at: https://openscholarship.wustl.edu/eng_etds Part of the <u>Computer Sciences Commons</u>

Recommended Citation

Yan, Yajie, "Medial Axis Approximation and Regularization" (2018). *Engineering and Applied Science Theses & Dissertations*. 336. https://openscholarship.wustl.edu/eng_etds/336

This Dissertation is brought to you for free and open access by the McKelvey School of Engineering at Washington University Open Scholarship. It has been accepted for inclusion in Engineering and Applied Science Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

Washington University in St. Louis School of Engineering and Applied Science Department of Computer Science and Engineering

> Dissertation Examination Committee: Tao Ju, Chair Jeremy Buhler Ayan Chakrabarti David Letscher Alvita Ottley

Medial Axis Approximation and Regularization

by

Yajie Yan

A dissertation presented to The Graduate School of Washington University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

> May 2018 Saint Louis, Missouri

Contents

Li	List of Figures				
A	cknov	${ m wledgments}$	X		
\mathbf{A}	bstra	\mathbf{x}	ii		
1	Intr	roduction	1		
	1.1	Shape Representation	2		
	1.2	Preliminaries	4		
	1.3	Medial Axis	7		
		1.3.1 Definition \ldots	7		
		1.3.2 Properties	0		
		1.3.3 Applications $\ldots \ldots \ldots$	1		
		1.3.4 Challenges $\ldots \ldots \ldots$	13		
	1.4	Contributions and Overview	.4		
2	App	proximation of Medial Axis	.6		
	2.1	Introduction	6		
	2.2	Related Works	8		
	2.3	Preliminaries	21		
	2.4	Approximating Medial Axis of Voxel Shapes	23		
	2.5	Approximating Medial Axis of Smooth Shapes	33		
	2.6	Algorithm	38		
	2.7	Results	ŧ0		
		2.7.1 Voxel shapes $\ldots \ldots 4$	1		
		2.7.2 Smooth shapes $\ldots \ldots 4$	15		
		2.7.3 Meshes	18		
	2.8	Summary	19		
3	Pru	ning of Medial Axis	51		
	3.1	Background and Motivation	51		
	3.2	Related Works	5 4		
	3.3	Erosion Thickness in 3D: Definition	56		
		3.3.1 Review: ET in 2D	6		
		3.3.2 Defining ET in 3D	57		

	3.4	Erosion Thickness in 3D: Computation	53
		3.4.1 Graph construction	53
		3.4.2 Graph-restricted ET and error bound	54
		3.4.3 Algorithm: burning the medial axis	55
		3.4.4 Algorithm: computing skeleton	57
	3.5	Results	'1
	3.6	Summary	'2
4	Con	clusion and Future Works	5
	4.1	Future works	7
		4.1.1 Exact burning of medial axis in 3D	'8
		4.1.2 Hierarchy of burn times	'9
		4.1.3 Recovery of Tree Topology	;1
Ap	pen	dix A Proof for Voxel Core	5
-	A.1	Proximity of voxel core to medial axis	35
	A.2	Proximity of voxelization to smooth shape	0
	A.3	Homotopy equivalence of voxelization to smooth shape	12
Ap	pen	dix B Proof for Erosion Thickness	7
-	Б.1	Properties of ET in 3D)7
		B.1.1 Finiteness)7
		B.1.2 Continuity	18
		B.1.3 Local minima	0
	B.2	Error bound of graph-restricted ET)1
\mathbf{Re}	fere	$nces \ldots \ldots$	6
Vi	ta.		4

List of Figures

1.1	An example for mesh and voxel representation of a ball in 2D and 3D. The mesh represents the boundary of ball with points and edges in 2D (a), and contains additionally triangle faces in 2D (b). Instead, the your represent	
	tation (c, d) uses cubes as basic elements. Elements of largest dimension	
	belonging to the ball are colored in red (edges in 2D and faces in 3D) and	
	lower dimensional elements in black (points in 2D and edges in 3D)	3
1.2	Illustration of some basic concepts in 2D. Every simplex is a face of itself and	0
	a proper face for all simplices below it (left). A mixed simplicial complex in	
	2D may not recover its original topology after taking the interior and then the	
	closure again (middle). However, a pure simplicial complex of full dimensional	
	simplices is equal to the closure of the interior of itself (right)	5
1.3	Two equivalent definitions for medial axis. (a) Blobby shape and its medial	
	axis. A medial point (orange) with its non-unique closest boundary points	
	(blue) define the medial ball at that point. (b) Grassfire analogy: the same	
	medial axis is obtained as the quench site of grassfire that starts from the	
	boundary of the shape. Maximal ball radii are visualized with lowest value	-
1 4	colored in darker blue.	8
1.4	A generic medial axis in 3D has 5 stable types of points with local topology	
	equivalent to x_1 - x_5 . A point whose heighborhood is equivalent to x_6 only appears during the burning of the model axis (Figure 2.4), which we will discuss	
	in Chapter 3. There we shall also see the "exposure" of a neighborhood is	
	done by removing a set of sectors (red) so that remaining part (gray) becomes	
	a boundary component	9
1.5	The medial axis of a 2D (a) [82] and 3D (b) shape. Note how topological	0
	features like holes and handles are captured by the medial axes	10
1.6	Using medial axis for shape simplification [85]	11
1.7	Using medial axis for shape smoothing [50].	11
1.8	Medial axis provides alternative shape representations which are beneficial for	
	animation (a) and hand tracking (b). Image courtesy of [86, 87]	12
1.9	Illustration of a 2D example where the medial axis is sensitive to a small bump	
	on the boundary. A noisy branch is extended to capture it	13

	is colored in red, and the dual Voronoi diagram $VD(P)$ in blue (b). The voxel	
2.2	core is the subset of $VD(P)$ dual to the elements in $DT(P)$ that intersect O. Voxels can be connected via common face, edge, or vertex in 26-connectivity.	24
	In 6-connectivity, which is what we use to study voxel shapes, voxels are only connected via a common face. Topologically, our voxel shape is equivalent to its interior without the boundary. Each case is illustrated using two voxels	
	(blue) and the equivalent underlying smooth shapes (light red).	25
2.3	Subdivision of a 2D voxel shape results in the voxel core (blue) that approaches	20
2.4	The voxel core (grav) of the 3D voxel shape gets closer to the true medial axis	30
2.1	(green) after voxel subdivision	31
2.5	Voxelization converts a smooth shape O bounded by B into a voxel shape O_h	
0.0	bounded by B_h . O_h consists of voxels of size h whose centers are in O	34
2.6	Steps for approximating the medial axis of a hand shape represented as trian- gle mesh (a). We first voyalize the shape (b). We then extract the voyal core	
	(c) with each point colored by the radius of the smallest sphere enclosing its	
	nearest boundary vertices (blue and red for small and large radii), and prune	
	it to its λ -subset (d)	39
2.7	The performance of our method (and the Tetgen component), HJ [74] and JST [37]. 5 shapes are voxelized at 10 resolutions (128^3-1280^3) . The memory (top)	
	and time (bottom) consumption is plotted against the number of boundary	
	voxel vertices (horizontal axis) for each shape at each resolution. HJ and JST	
	both terminate early due to lack of memory $(+)$ and crashing($^{\circ}$), while our method finishes processing all inputs	/1
2.8	For a high resolution CT scan of corn roots (top), the medial axis is computed	71
	using our method (middle), which is then fed to Erosion Thickness (to be	
	presented in Chapter 3) for a clean curve skeleton (bottom). Each point of	
	the skeleton is colored by the local thickness (radius of medial ball) of the	
	shape (blue:thin, red:thick). Two regions are zoomed in to show how a good	10
2.0	Different) metal area (ashing complicated structures	43
2.9	Different λ voxel cores (counting) are extracted for Empsoid shape voxelized at 4 resolutions (rows) to show how λ affects geometric convergence of medial	
	axis. The true medial axis at top-left is highlighted in red. Each point of full	
	voxel cores is colored with radius of minimal enclosing sphere of the nearest	
	boundary voxel vertices.	44

2.10	Comparison with sampling-based methods, SAT [50], PC [2], and DZ [23]. Next to each medial axis we show: the time and memory. Topology informa- tion is reflected in a pair of Euler characteristics and number of components, where incorrect numbers are marked red. Two pairs are shown before and	
2.11	after duplicate faces are removed for methods other than ours	46
2.12 2.13	similar level of geometry at 512° with much lower cost	47 48 49
3.1	The medial axis of a bumpy 3D shape contains numerous spurious sheets. Erosion Thickness (ET) highlights parts of the medial axis that represent significant shape features. We use ET to create a family of clean, topology-	50
3.2	Local vs global measures. Top: Medial axes of the same 2D shape colored (blue for low and red for high) by the object angle measure, the circumradius measure, and Erosion Thickness. Bottom: Subsets of the medial axes where the respective measure is higher than some threshold. Local measures, such as object angle and circumradius, fail to differentiate major shape features	52
3.3	(e.g., the horse legs) from boundary noise (e.g., hair)	53
3.4	Burning on the medial axis of a 3D shape (shown in the top-left) at five time points and at completion	50
3.5	A 2D example where for x the burn time, i.e. the length of the shortest path can only be realized in the limit as the path cannot include the red part of the cylinder base. Otherwise the path needs to fork onto the cylinder, creating a	09
3.6	much longer exposing tree, thus failing to capture burn time	62
3.7	and edges between them	04 70
	troned by parameters σ_2 and σ_1	10

3.8	Comparison with local significance measures. Each measure is visualized by heat color (higher values use hotter color) and transparency (higher values are more opaque). Next to it is the portion of the medial axis whose significance values are higher than a threshold. Note that ET is sensitive to thin, plate-like	
3.9	features (see inserts) while being robust to perturbation	72 73
3.10	Skeletons generated by our method are robust to boundary noise due to stability of ET	74
4.1	Burning processes and products in 3D unified in the same framework. From left to right, each burning takes a shape and a function defined on it (0 everywhere in the initial Hand shape), and produces the quench site - a thin medial structure, and the burn time on it. Two consecutive burn times define a significance measure (ET) that highlights features expanded along all but one dimension of the embedding space. Hence, planer features stand out on MA embedded in 3D space, while tubular features on MC embedded in 2D MA	81
4.2	A visualization of a curve skeleton inside the roots 3D image, which contains many topological errors (a). Small cycle (b) and merging (c) of branches can be identified with the help of skeleton and the associated shape measures	83
A.1 A.2	Notations used for Lemma A.1.1. \dots Notations used in the proof of Theorem 2.4.4. Left: <i>e</i> is a voxel face. Right: <i>e</i> is a voxel edge shared by only two voxel faces. Viewing direction is parallel	86
A.3	to the e	87
A.4	contains B_h	91
A.5	They share the face having two inside voxel centers. Image courtesy of [77] (a): A dual cube whose 8 corners are voxel centers. (b): The voxel occupancy of the dual cube (8) in Figure A.4. Red/White dots: in/outside voxel centers. (c): The boundary of our topology preserving shape in the cube (and pairing- cube below) is formed by pressing the center of the voxel boundary toward the face shared by the cube-pair, resulting in a manifold surface whose boundary curve is colored by black	94 95
		55

B.1	G^+ adds jump-arcs (green) to G on each triangle edge, forming the complete	
	graph within a triangle	102
B.2	Each chord c (dashed) is a segment within a triangle of a path on M . It can	
	be "snapped" to an arc a (solid) in one of the three ways	103
B.3	A path γ of an exposing tree meandering in a triangle (dashed) can be straight-	
	ened into one chord (solid) that's simpler and shorter	104

Acknowledgments

It feels surreal that I am concluding my PhD study. During this challenging journey, I am always grateful to the people who supported me.

First and foremost, I would like to dedicate my most sincere gratitude to my advisor, Dr. Tao Ju. He took me in the lab at a very special moment, and provided me all the resources I need to pursue my study. He guided me into an interesting and valuable research area, showed me the beauty of theoretical reasoning, and taught me the importance of passion and tenacity. To me, Dr. Tao Ju is not only an academic advisor, but also a mentor in life. In addition to my advisor, I would like to thank other committee members: Dr. Jeremy Buhler, Dr. David Letscher, Dr. Alvitta Ottley and Dr. Ayan Chakrabarti for their thoughtful comments that helped me improve my work.

I would also like to thank my collaborators at St Louis University: Dr. Kyle Sykes, Dr. Erin Chamber, and Dr. David Letscher. Their expertise in topology opened my eyes for many interesting math problems, and their focus on mathematical rigor ensured the quality of our study.

At the same time, the companionship of many lab-mates has made my PhD study much less stressful. I want to thank Hang Dou, Zhiyang Huang, and Ming Zou for being the best comrades and thought challenger. I admire Chen Liu, Hang Yan, and Huayi Zeng for their hard work and intriguing brainstorming.

Personally, I owe everything to my parents whose unconditional love propels me forward fearlessly.

Last but not the least, I would like to give my special thanks to my wife Li. Because of her, every life changing moment is not as rough, and I've become a better person.

Yajie Yan

Washington University in Saint Louis May 2018 Dedicated to my parents.

ABSTRACT OF THE DISSERTATION

Medial Axis Approximation and Regularization

by

Yajie Yan

Doctor of Philosophy in Computer Science Washington University in St. Louis, May 2018 Research Advisor: Professor Tao Ju

Medial axis is a classical shape descriptor. Among many good properties, medial axis is thin, centered in the shape, and topology preserving. Therefore, it is constantly sought after by researchers and practitioners in their respective domains. However, two barriers remain that hinder wide adoption of medial axis.

First, exact computation of medial axis is very difficult. Hence, in practice medial axis is approximated discretely. Though abundant approximation methods exist, they are either limited in scalability, insufficient in theoretical soundness, or susceptible to numerical issues. Second, medial axis is easily disturbed by small noises on its defining shape. A majority of current works define a significance measure to prune noises on medial axis. Among them, local measures are widely available due to their efficiency, but can be either too aggressive or conservative. While global measures outperform local ones in differentiating noises from features, they are rarely well-defined or efficient to compute. In this dissertation, we attempt to address these issues with sound, robust and efficient solutions. In Chapter 2, we propose a novel medial axis approximation called voxel core. We show voxel core is topologically and geometrically convergent to the true medial axis. We then describe a straightforward implementation as a result of our simple definition. In a variety of experiments, our method is shown to be efficient and robust in delivering topological promises on a wide range of shapes. In Chapter 3, we present Erosion Thickness (ET) to regularize instability. ET is the first global measure in 3D that is well-defined and efficient to compute. To demonstrate its usefulness, we utilize ET to generate a family of shape revealing and topology preserving skeletons. Finally, we point out future directions, and potential applications of our works in real world problems.

Chapter 1

Introduction

Shapes are the building blocks of the world. Understanding of shapes is beneficial to many research fields [52, 56, 84]. For example, robots rely on shape recognition techniques to navigate in their environment without bumping into obstacles. Plant scientists study the shape of roots to better engineer the plants. Pharmaceutical researchers apply insights gained from analyzing the shape of proteins for drug design. In all these scenario, automatic shape analysis and understanding play a crucial part. However, computational tools that match the performance of human vision are still at large. Moreover, such tools are often required to outperform human in terms of quantitative accuracy, making the solution even more challenging.

In light of this, shape analysis has become an active research area. The de-facto approach to carry out shape analysis is to find proper shape descriptors which facilitate high-level reasoning, e.g., shape matching and retrieval [80, 15, 91, 92]. A shape descriptor is an abstraction of the shape, which is defined by its geometry and topology. Geometry describes what the shape looks like, while topology reflects more structural information, including the number of components and holes. Existing shape descriptors often capture one aspect of the shape, whereas medial axis encodes both the geometry and topology. We believe focusing on both aspects of a shape is beneficial to shape understanding and its downstream applications. Hence, medial axis is the topic of the dissertation.

To carry out any computation on shapes, we need to represent them in digital forms. Different representations lead to different methods of performing the same analysis. In the remaining of the chapter, we first discuss a few common shape representations. We then introduce the medial axis and highlight its relation to different shape representation. Lastly we give an overview of the dissertation.

1.1 Shape Representation

There are two common shape representations: boundary and volume. Depending on the specific application, one can be more useful than the other. For displaying an opaque object, representing the boundary usually suffices. In other applications, e.g. medicine and biology where the interior of an object is often studied, it is important to represent the entire volume of the shape.

Boundary representation only encodes the interface separating the inside and outside of the shape. A popular realization is triangular mesh, which samples the shape boundary with points and connects them with triangle faces (see Figure 1.1). As a simplicial complex, the elements in a triangle mesh either are separated, or intersect at a common vertex or edge. This light-weight and simple representation is widely used in computer graphics for shape exchange, rendering, and geometry processing. One drawback about triangular meshes is they can only represent piece-wise linear shapes exactly. Furthermore, in areas like manufacturing, a human being often needs to digitally build and manipulate mechanical parts



Figure 1.1: An example for mesh and voxel representation of a ball in 2D and 3D. The mesh represents the boundary of ball with points and edges in 2D (a), and contains additionally triangle faces in 3D (b). Instead, the voxel representation (c, d) uses cubes as basic elements. Elements of largest dimension belonging to the ball are colored in red (edges in 2D and faces in 3D), and lower dimensional elements in black (points in 2D and edges in 3D).

with free-form surfaces and curves, which is hard to perform on a triangular mesh. For these purposes, B-Rep (boundary representation) is adopted. In contrast to a triangle mesh where each face is a linear polygon, a B-rep is more versatile in the sense it contains a collection of parametric patches whose spatial relationship is encoded using a graph. A parametric patch is a "deformed" square shape with each point on 2D domain assigned a position in 3D. Because of its mathematical definition, parametric patches can be relatively easy to edit while preserving curved surface details.

Volume representation treats the shape as partitioning the space into interior and exterior. Two typical representations are voxel shape and tetrahedral mesh. Voxel shape partitions the entire space using cubes of same size, known as voxels. The center of each cube is marked with a flag indicating in- or out-side of the shape. The shape has its interior represented as the union of inside voxels, bounded by faces, edges, and vertices (see Figure 1.1). On the other hand, a tetrahedral mesh represents a shape using a set of tetrahedra. Similar to a triangle mesh, a tetrahedral mesh is a simplicial complex in 3D. Each tetrahedron represents a part of the volume inside the shape, with vertices, edges, and triangle faces that are on the boundary or interior of the shape. Like any simplicial complex, elements in a tetrahedral mesh are not allowed to overlap, hence two tetrahedra can only be disjoint, or share a common vertex, edge, or triangle face. With the interior modeled, volumetric representation facilitates many application such as volume-preserving shape deformation, physical simulation, and finite element analysis, all of which are difficult to perform with boundary-only representations.

While there is no "best" shape representation that all techniques should target, it makes sense to design a method based on the most commonly used shape representation. In the following, we choose to focus on triangle meshes and voxel shapes for two reasons. First these shapes are general enough to which other types of shapes can easily convert. Second, our research is greatly inspired by and heavily based on a large body of medial axis methods designed for triangle meshes and voxel shapes.

1.2 Preliminaries

We define basic concepts using point set language to ensure later we can have discussion with minimal ambiguity. See Figure 1.2 for a visual presentation of some of the concepts.

The open ball is a building block in topology.

Definition 1.2.1. Commonly a ball in space centered at point $p \in \mathbb{R}^n$ with radius r is defined as $\mathbb{B}_r(p) = \{x \in \mathbb{R}^n \mid d(x, p) \leq r\}$, where d(x, y) is the Euclidean distance between two points. We also have a sphere $\mathbb{S}_r(p) = \{x \in \mathbb{R}^n \mid d(x, p) = r\}$. The **open ball** is the remainder after taking away the sphere from the ball $\mathbb{B}_r(p) \setminus \mathbb{S}_r(p)$.



Figure 1.2: Illustration of some basic concepts in 2D. Every simplex is a face of itself and a proper face for all simplices below it (left). A mixed simplicial complex in 2D may not recover its original topology after taking the interior and then the closure again (middle). However, a pure simplicial complex of full dimensional simplices is equal to the closure of the interior of itself (right).

The unit ball is a specialized open ball centered at the origin $\mathbf{0} \in \mathbb{R}^n$ with unit radius: $\mathbb{B} = \{x \in \mathbb{R}^n \mid ||x|| = 1\}$. Another example is the **half ball**, which is defined as the unit open ball restricted to the half-space of the last dimension $\mathbb{H} = \{x \in \mathbb{R}^n \mid ||x|| = 1 \text{ and } x_d \leq 0\}$. In the following we only consider balls with non-negative radii.

We can then re-define the already familiar concepts like *open* and *closed* set using open ball.

Definition 1.2.2. Given a set $A \subseteq \mathbb{R}^n$, it is **open** if for any point $p \in A$ there exists an open ball with positive radius that's fully contained in A. A is **closed** if its complement $\mathbb{R}^n \setminus A$ is open. A is **bounded** if two points in it cannot be infinitely far away, that is $p, q \in A$, and $d(p,q) \leq l$, where $l \geq 0$. We call a set **compact** if it's both open and bounded.

With these constructs, a shape can be mathematically modeled as a compact subset in space \mathbb{R}^n . We further formalize the ideas of boundary and interior of a shape.

Definition 1.2.3. The **boundary** of a set $A \in \mathbb{R}^n$, denoted as ∂A is the set of points for which any open ball, however small, always intersects with both A and its complement. The **closure** of A, defined as $\overline{A} = A \cup \partial A$, adds the boundary part to itself. Conversely, the removal of ∂A from A, i.e. $A \setminus \partial A$ produces the **interior** of A.

A concrete example of shape modeled mathematically is a simplicial complex (Figure 1.2).

Definition 1.2.4. A k-simplex δ is the convex hull of k + 1 vertices P. In particular, k-simplex is a vertex, edge, triangle, and tetrahedron for k = 0, 1, 2, 3. A k-simplex is said to have dimension k. A face of δ is the convex hull of a non-empty subset of P, thus a simplex ranging from any vertex to δ itself. A **simplicial complex** is a point set composed of finite number of simplices. It contains as element every face of every simplex. The intersection of two simplices in the complex is either a common face (vertex, edge, triangle, etc.) or empty. Simplicial complex offers an abstraction to discrete shape representation. For example, a triangular mesh is a simplicial complex. It is also the vehicle that enables analysis of theoretical properties for many practical tools operating on meshes. Hence, while being a mathematical object, simplicial complex is closely relevant to practical applications. In Figure 1.2, we show two example simplicial complexes in 2D. One contains purely triangles, while another contains two and lower dimensional simplices. Note that those simplices of lower dimension will be lost after opening since they have no interior of same dimension as those full dimensional elements.

1.3 Medial Axis

1.3.1 Definition

Medial axis is a classical shape descriptor. Since the introduction by Blum in 1961 [12], medial axis has received wide attention due to its faithful representation of a shape's geometry and topology at the same time. Many possible applications using medial axis are explored in computer graphics, computer vision, and computational biology.

Most commonly, the medial axis of a shape can be defined as the set of points inside the shape with non-unique closest points on the boundary. Mathematically, consider a bounded and open set O of space \mathbb{R}^n . Any point $x \in O$ has a set points on boundary ∂O equally closest to x:

$$\Gamma(x) = \{ y \in \partial O \mid d(x, y) = d(x, \partial O) \}$$

where d is the Euclidean distance. The medial axis M is then

$$M = \{ x \in O \mid |\Gamma(x)| \ge 2 \}$$

As an example, a blobby shape with its medial axis is shown in Figure 1.3(a). Notice a maximal inscribed ball centers at each medial point x and touches the boundary at points $\Gamma(x)$, known as medial ball. The loci of medial ball centers coincide with the medial axis, and their radii are often referred as the radius function. Medial axis and the radius function give rise to the notion of Medial Axis Transform (MAT).



Figure 1.3: Two equivalent definitions for medial axis. (a) Blobby shape and its medial axis. A medial point (orange) with its non-unique closest boundary points (blue) define the medial ball at that point. (b) Grassfire analogy: the same medial axis is obtained as the quench site of grassfire that starts from the boundary of the shape. Maximal ball radii are visualized with lowest value colored in darker blue.

It is worth mentioning one more intuitive definition of medial axis, i.e. the "grassfire analogy" proposed by Blum. Imagine the shape represents a grass prairie. As illustrated in Figure1.3(b), a fire starts on the entire boundary of the prairie simultaneously. The fire propagates along inward normals at a constant speed. Different fire fronts quench when they meet. The quench site forms the medial axis. The arrival time at the quench site gives the radius function. Though other definitions of medial axis exist, they are equivalent in the



Figure 1.4: A generic medial axis in 3D has 5 stable types of points with local topology equivalent to x_1 - x_5 . A point whose neighborhood is equivalent to x_6 only appears during the burning of the medial axis (Figure 3.4), which we will discuss in Chapter 3. There, we shall also see the "exposure" of a neighborhood is done by removing a set of sectors (red) so that remaining part (gray) becomes a boundary component.

sense that the unique MAT is obtained. We refer the readers to the recent survey [82] for more definitions.

While simply defined, the medial axis often exhibits a rich structure consisting of curved lines (in 2D) and sheets (in 3D) that join in a rather complex manner. Studying the structure of a medial axis is crucial to understanding its unique properties and several applications. In Figure 1.4 we take a close look at the types of points on an example medial axis M in 3D. Notice it consists of 3 sheets glued along singular curves, or *seams*. Multiple seams meet at a *junction point*. We assume the medial axis is a generic structure, which means the types of local neighborhood are stable under infinitesimal perturbation to all medial points. According to [30], there are five types of points depicted by x_1, \ldots, x_5 on M in the figure. Each point contains one or more sectors in its local neighborhood. More precisely, x_1 is a manifold point since it has a neighborhood equivalent to a 2D open ball; x_2 is a boundary point since it's locally equivalent to a 2D half ball; while $x_3 \cdot x_5$ are all singular points whose neighborhoods are made up of multiple sectors. x_6 is not a stable type of points in generic situation, but it will arise during the *burning* of medial axis as we shall see in Chapter 2. In some of these neighborhoods, we can identify a set of sectors (colored in red) so that removing them will "open up" the neighborhood, making the remaining part equivalent to a half-disk (colored in gray). Note type 2 and 6 neighborhoods are "open" already. Later we shall see this operation is a crucial component when we introduce burning of a medial axis in 3D in Chapter 3.

1.3.2 Properties



Figure 1.5: The medial axis of a 2D (a) [82] and 3D (b) shape. Note how topological features like holes and handles are captured by the medial axes.

The elegant definition of medial axis entails a set of important properties. Here we highlight four. See Figure 1.5 for two examples.

Thin. Medial axis is guaranteed to be at least one dimensional lower than its defining shape. In 2D, it is a 1D planar graph consisting of curved lines joining at junction points; In 3D, it is a surface structure containing 2d "sheets" glued along non-manifold curves. **Topology preserving**. Medial axis faithfully encodes topology of the input shape, i.e. the number of components and genus. Specifically, for a 2D shape genus counts the number of holes which are preserved by loops of lines in the medial axis. For a 3D shape, genus counts the number of handles, each of which corresponds to a loop of sheets in the medial axis.

Centered. Medial axis naturally sits in the center of the shape because every medial point is "sandwiched" by two or more boundary points. Hence the name "medial".

Reconstructibility. Medial axis transform can reconstruct the original shape using union of the medial balls. Further, the level of details of reconstruction can be controlled by selecting only a subset of medial balls.

1.3.3 Applications



Figure 1.6: Using medial axis for shape simplification [85].



Figure 1.7: Using medial axis for shape smoothing [50].

Through its desirable properties, medial axis has found its way into many high-level applications. Tam and Heidrich [85] propose a simplification method that first removes branches



Figure 1.8: Medial axis provides alternative shape representations which are beneficial for animation (a) and hand tracking (b). Image courtesy of [86, 87].

from medial axis and then reconstructs a simplified shape (Figure 1.6.). Similarly, Miklos et al. [50] achieves shapes smoothing by simplifying away small scale features on the medial axis (Figure 1.7.). To efficiently render excessively detailed laser scanned data, Peters and Ledoux [59] utilize the medial axis to down-sample point clouds without severe loss of visual quality. Inversely, to better reconstruct shapes from incomplete scans, Tagliasacchi et al. [83] up-sample the data to complement the missing region from an estimated medial-axislike structure. In computational biology, researchers use the skeletal structure derived from medial axis to locate alpha-helices and beta-sheets of proteins [10]. Recently, medial axis has inspired a line of works that investigate effective shape approximation in more interactive settings, e.g. computer animation and hand tracking [86, 87, 88] (Figure 1.8.). Other applications take advantage of the shape depicting property of medial axis to improve feature extraction, shape metrology, segmentation and part decomposition [65, 72, 64, 66, 42].

1.3.4 Challenges

Admittedly, it is not without cost to achieve these properties. On one hand, exact computation of medial axis is very difficult. Due to concave regions on the shape, the medial axis often contains hyperbolic lines and surfaces in 2D and 3D. Especially in 3D, tracking these geometries and their intersections exactly requires finding roots of high-degree polynomials [1]. The only exact algorithm [21] is difficult to implement, and slows down quickly even when shapes are only slightly complicated. Therefore, most efforts have been focused on approximating medial axis. However, this is not a trivial task either. For example, one class of existing methods attempts to straightforwardly simulate the grassfire analogy on a voxel shape by peeling off voxels layer by layer until the subset of voxels are left which can be used to approximate the medial axis. However, these methods generally scale poorly and ignore theoretically bounding the difference between their results and the true medial axis. The other class of methods resorts to sampling the shape on the boundary and proves the Voronoi diagram or some derivative structure of the samples approximates the medial axis under a set of theoretical guarantees. However, numerical robustness is affected by the high complexity of the algorithms.



Figure 1.9: Illustration of a 2D example where the medial axis is sensitive to a small bump on the boundary. A noisy branch is extended to capture it.

On the other hand, medial axis is notoriously sensitive to even tiny bumps on shape's boundary. As illustrated in Figure 1.9, even a tiny bump on the boundary is captured by some branch of the medial axis. This is due to the very definition of medial axis: every point on the noisy branch has two or more nearest points within the bump, thus being part of the medial axis. As a clean medial axis is vital to many applications, a major research direction concentrates on medial axis pruning. These two fundamental challenges significantly hinder wide adoption of medial axis.

1.4 Contributions and Overview

In this dissertation, we aim to tackle the two challenges with novel solutions.

Approximation using voxel core To address the computation challenge, we present *voxel* core, a new medial axis approximation solution. Existing methods either lack theoretically supported error analysis between approximation result and true medial axis, incur large memory footprint thus scale poorly, or fall short in numerical robustness and ease of reproducibility. Our voxel core, while simply defined, features theoretically sound properties as being thin, topology preserving, geometrically close to the true medial axis of a voxel and smooth shape. In practice, voxel core is simple to implement, computationally efficient, and delivers theoretical promises robustly for both voxel shapes and smooth surfaces.

Pruning using Erosion Thickness To address the hyper-sensitivity of medial axis in 3D, we present a novel significance measure called *Erosion Thickness*. The majority of the existing works define a significance measure over the medial axis to prune away the subset considered as less important guided by the measure. Among them, most are locally defined.

While being simple and fast to compute, local measures often mistake noises for features and vice versa. More desirable measures are based on global shape information. These measures can better distinguish features from noisy branches, but are rare and expensive to compute. Our Erosion Thickness is the first mathematically defined global measure that's noise-discriminative and efficient to compute. To demonstrate its usefulness, we utilize ET to compute a family of skeletons with surfaces and curves to represent planar and tubular features on the shape. Because of the good properties of ET, our skeletons are noise-resilient, shape revealing, and topology preserving.

In the following, we first describe medial axis approximation using voxel core in Chapter 2. We then present Erosion Thickness significance measure in Chapter 2. Finally, we conclude the dissertation with some discussion and future works.

Chapter 2

Approximation of Medial Axis

2.1 Introduction

Despite having a simple definition, in 3D the medial axis is a complex network of curved surfaces for a moderately complex shape. Exact computation methods exist, e.g. [21]. They guarantee to reproduce medial axis with properties as discussed in Section 1.3.2, but are only feasible for simple shapes due to high computational cost. In practice, most existing methods resort to approximation to be able to scale to complex real-world shapes. Unlike exact methods, medial axes generated by approximation methods don't always carry all theoretical soundness. Thus, reliably computing sound medial axes approximation is a fundamental challenge.

Current research in medial axis approximation generally follows two approaches: voxel-based and sampling-based. **Voxel-based** methods operate on voxel shapes, which are native representation for shapes in e.g. medical imaging and physical simulation (sec. 1.1), or can be obtained from boundary-only shapes via the voxelization process. The methods work by peeling voxels layer by layer starting from the boundary of the shape until the remaining subset can approximate the medial axis. The peeling process can easily preserve the topological equivalence of the voxel shape, and often leads to a simple implementation with strong numerical robustness. However, they need to store the entire voxel volume in memory, which incurs an expensive time and space cost. Moreover, the theory quantifying the geometric similarity between the computed and true medial axis is lacking. On the other hand, **sampling-based** methods sample the given shape with a set of sufficiently dense points set on or near the boundary. They then compute the Voronoi diagram, and take a subset of it to approximate the medial axis of the original shape. Compared to voxel-based methods, these methods can scale to much larger input sizes since only boundary samples are needed. The theory support behind this line of works is also abundant, showing both topological and geometrical convergence of the approximate results to the true medial axes when certain conditions are met. However, most methods involve non-trivial geometric procedures (e.g. multiple passes of Voronoi diagram computation) which are not numerically robust. Due to this reason, results of sampling-based methods are often ridden with topological errors.

Contributions In this chapter, we address this challenge with a new method for approximating the medial axes of both voxel and boundary-only shapes that is simple, efficient, and numerically robust. More importantly the method features all theoretical guarantees for voxel and smooth shapes as it is supported by two novel observations on voxel shapes. First, we show that the medial axis of a voxel shape can be approximated with both topological and geometrical guarantee using the *voxel core*, i.e. the interior subset of Voronoi diagram of voxel corners on the boundary. Second, we show that the voxel core can serve as a theoretically sound approximation to the medial axis of any smooth shape with converging behavior via sufficiently fine voxelization of the shape. These theoretical insights lead to a simple approximation method applicable to both voxel shapes (natively) and boundary-only shapes (via voxelization). Compared to existing voxelbased methods, our method not only scales better to large input volumes but is also equipped with convergent guarantees. Compared with current sampling-based methods, our method is not only simpler to implement but can capture the topology of the input more robustly.

For the remaining of the chapter, we plan to review the works that are most relevant to our method. Then we will formalize various notions including voxel shapeand the voxel core. Theoretical results are stated and proofs are given for topological guarantees, while other proofs for geometric guarantees are presented in the appendix. After the theory part, the algorithm is self-explanatory and presented in sec. 2.6. We conclude the chapter with experimental results and comparisons.

2.2 Related Works

We briefly review two closely related bodies of work, i.e. voxel-based methods and samplingbased methods. Readers can check the recent surveys [82, 75] for more medial axis generation approaches.

Voxel-based methods These methods operate on voxel shapes and approximate the medial axis using a subset of the voxels featuring properties similar to the medial axis, such as being thin, centered, and preserving the topology of the shape [76, 69]. Focusing on only voxels lends these methods great simplicity. In the core of such a method is a thinning process, where one simply strips away layers of voxels starting from the boundary that meet certain criteria and keep those voxels essential for maintaining the right topology [70, 11].

The order of removal of voxels is usually guided by non-Euclidean distance metric, such as Manhattan distance [58, 90], chamfer distance [63], and < 3, 4, 5 > distance [6]. However these distances are dependent on the voxel grid orientation. Thus the medial axis results are sensitive to transformation like rotation of the shape. More accurate and transformationinvariant results can be obtained using the Euclidean distance field [5, 29], or other fields derived from more global shape information [37, 74, 67, 68]. Nevertheless, these enriched fields require more information for each voxel, hence incurring drastically more time and space cost, both of which is a cubic function of the voxel resolution. As imaging technologies mature, voxel shapes of resolution 1024^3 are not uncommon in these days. Therefore the lack of scalability puts a strong limit on the applicability of these methods.

Instead of having voxels of same size, a voxel shape can be organized into a hierarchical structure which serves space hashing functionality while minimizing its storage footprint. Utilizing this feature, other methods search for medial axis features initially in every voxel [28, 45], and only refine the search when necessary [27, 78, 79]. Still, the computational cost can be relatively high.

In practice, most of the voxel-based methods can and have been applied to non-voxel shapes, e.g. triangle meshes, via voxelization. However, to our knowledge, there is no medial axis approximation work based on this technique that formally discusses the topological and geometric quality of such approximation.

Sampling-based methods As the name suggests, this class of methods consider samples on or around the boundary of the shape. They usually output a subset of the Voronoi diagram of the sample points or some derivative structure as the medial axis. Without worrying about the interior of the shape, these methods are natively more efficient and scalable than voxel-based methods. More effort is therefore devoted to which part of Voronoi diagram is to keep and assuring the subset they consider can serve as a good medial axis approximation, meeting both topological and geometric properties. In 2D, one can simply take the Voronoi diagram that's inside the shape when the sampling is dense enough [14]. This is not true in 3D because not every point of the Voronoi diagram inside the shape is close to the true medial axis due to the existence of "sliver" triangles a Voronoi element can be dual to [2]. A sampling-based method usually needs to carefully ensure no part that's distant from the medial axis is in the result, or no part that's close is missed out.

Some works keep a subset of the Voronoi diagram filtered by some angle-based criteria [8, 23]. The output is shown to approach to the true medial axis geometrically as sampling distance goes to zero, but may not maintain the same topology as the shape. Giesen et al. [32] discover that the unstable manifold of the Voronoi diagram features the same topology as the smooth shape and can also achieve the geometric convergence with the help of the angle-filtration from [23]. Unfortunately this structure is extremely expensive to compute [16].

Amenta et al. [2] propose to consider "poles" of Voronoi diagram and use the dual of the power diagram of them as the medial axis. Known as power shape, the structure stays close to the true medial axis and preserves the same homotopy of the shape, except consisting of full dimensional solids, e.g. tetrahedra in 3D, thus not thin. Later, thinness is achieved by replacing power shape with the medial axis of a union of ball representation of the shape which can be computed exactly [4, 85]. However this delicate structure is difficult to be computed robustly as it involves multiple passes of Voronoi diagram computation and geometric intersections. The widely used implementation by Miklos et al. [50] routinely produces results with duplicate faces and closed "pockets" as we will see later. Generally speaking, our method is equally theoretically sound, and efficient as samplingbased methods, while being simpler to implement and numerically robust similar to voxelbased methods.

2.3 Preliminaries

Here we introduce some constructs tightly related to the theoretical study in this chapter.

We first briefly review two classical computation geometry objects, namely the Voronoi diagram and Delaunay triangulation, for which examples are illustrated in Figure 2.1. Similar to sampling-based methods, our method is also heavily based on these structures.

Definition 2.3.1. Given a point set $P \subset \mathbb{R}^n$, a.k.a. sites, the **Voronoi diagram** VD(P) partitions the space into Voronoi cells, each associated with a site. The **Voronoi cell** of $p \in P$ contains all points in \mathbb{R}^n to which p is closer or equally distant than other sites. A Voronoi cell in 3D has d dimensional elements on its boundary where d = 2, 1, 0, namely **Voronoi faces, edges, vertices**, while a Voronoi cell in 2D only has Voronoi edges and vertices on the boundary. The **Voronoi diagram** of P, VD(P), is made up of the boundary of all Voronoi cells. Each d-dimensional element of VD(P) is dual to a (n - d)-dimensional element in the corresponding **Delaunay triangulation** of the set P, denoted by DT(P). Denote an element in DT(P) by $e \in DT(P)$, and its dual $\tilde{e} \in VD(P)$, the **Delaunay element** is defined as the convex hull of points in P whose dual Voronoi cells share \tilde{e} on their boundary.

First, observe that the interior of a Voronoi cell is strictly closer to only one input site, while its boundary is closer to two or more sites. Therefore VD(P) can be equivalently defined as the set of points in \mathbb{R}^n that are equally closest to multiple sites. Second, let's assume all points are in generic positions, meaning at most three points can lie on a circle and four points on a sphere. Then, a Voronoi face, edge, and vertex can be equidistant to two, three, and four sites respectively. As a result, the elements in the dual structure, DT(P), are all simplices. To be precise, a *d*-dimensional element $e \in DT(P)$ for $d = 3, \ldots, 0$ is a tetrahedron, triangle, edge, and vertex, respectively. In many cases the generic assumption is not satisfied, e.g. when P is the voxel corners on the boundary of a voxel shape. In this case the Voronoi diagram VD(P) will have edges and vertices that are equidistant to arbitrary number of sites. As a result, their duals will become general polygons and polyhedron. However, both situations are nicely covered by our definition of Delaunay element.

We would like to highlight one property about Delaunay triangulation, i.e. the empty ball property, which often stands out to be very useful in many Voronoi diagram and Delaunay triangulation related reasoning.

Theorem 2.3.1. *Empty ball property.* Given the point set P and its Delaunay triangulation DT(P), e is an element in DT(P) if and only if the vertices of e lie on an n-dimensional empty ball whose interior contains no other points from P.

One can show that for a Delaunay element, any empty ball of it is centered on the corresponding Voronoi dual element.

We then formally introduce Hausdorff distance, which is a common way of quantifying difference between two point sets. We are interested in measuring distance between two medial axes.

Definition 2.3.2. Given two non-empty sets $A, B \subset \mathbb{R}^n$, and the distance function d(x, X)from any point $x \in mathbbR^n$ to a set $X \subset \mathbb{R}^n$, we can define a one direction distance from
A to B as

$$d_{A \to B} = \sup_{x \in A} d(x, B)$$

similarly we have for the direction B to A

$$d_{B \to A} = \sup_{x \in B} d(x, A)$$

the larger of the two defines Hausdorff distance

$$d_H(A, B) = \max\{d_{A \to B}, d_{B \to A}\}$$

2.4 Approximating Medial Axis of Voxel Shapes

We are now ready to conduct a formal study of medial axis, particularly we argue to use our voxel core as an approximation for the medial axis of a voxel shape, which is justified by rigorously analyzing its topological and geometric properties.

We consider voxel shapes in 3D. As a reminder, voxels are cubes of same sizes that tightly pack and partition the space \mathbb{R}^3 . The boundary of a voxel is made up of voxel faces, edges, and vertices. Neighboring voxels may share an entire vertex, edge, face, or nothing. A voxel shape O is the interior of the union of a set of finite voxels. Voxels belonging to the shape are often called foreground or object, while those outside can be referred to as background. Our voxel shape definition implies 6-connectivity for object voxels, meaning two voxels need to share a face to close the space between them. See Figure 2.2 for intuitive examples. The book by Klette and Rosenfeld [39] is a comprehensive material for voxel topology.



Figure 2.1: A 2D illustration of voxel core. Given a simple voxel shape O bounded by boundary B whose vertices are P (a), the Delaunay triangulation DT(P) of P is colored in red, and the dual Voronoi diagram VD(P) in blue (b). The voxel core is the subset of VD(P) dual to the elements in DT(P) that intersect O.



Figure 2.2: Voxels can be connected via common face, edge, or vertex in 26-connectivity. In 6-connectivity, which is what we use to study voxel shapes, voxels are only connected via a common face. Topologically, our voxel shape is equivalent to its interior without the boundary. Each case is illustrated using two voxels (blue) and the equivalent underlying smooth shapes (light red).

Voxel shapes features a regularity which the proofs in the following exploit:

Lemma 2.4.1. Given a point $p \in \mathbb{R}^n$ lying on a voxel element e (e.g. edge or face), the closest voxel vertices to p are vertices of e.

Proof. This should be intuitive in 2D and 3D since each voxel is a cubic shape. To see that any point p on a voxel element e, we grow a sphere centered at p. The sphere will touch some vertex of e before reaching to vertices of other voxel elements.

Led by this feature and the empty ball property, we present a property unique to voxel shapes. Denote together the boundary voxel vertices, edges, and faces for a voxel shape O by B, and particularly the boundary vertices by P. It is true that:

Theorem 2.4.1. The boundary set B is a subset of the Delaunay triangulation DT(P) of the boundary vertices P.

Proof. According to the empty ball property (Theorem 2.3.1), it suffices to show there exists an empty ball for each element $e \in B$. We can construct a ball with center at the centroid of e passing through all vertices of e, because its vertices are equally far from the centroid. The ball is empty because no other points in P are closer to the center of the ball than vertices of e by Lemma 2.4.1.

We stress that this property is not enjoyed by a general 3D polyhedron, i.e. not every edge or face of a polyhedron is Delaunay.

A useful consequence of the property is that B divides DT(P) into two subsets. One subset is the closure of voxel shape \overline{O} , the other is the complement of O. Note that this is not a clean division, since the two subsets overlaps at B. The rationale behind is by definition a Delaunay element itself is closed with its boundary. Using the first subset, we define voxel core as its dual in the Voronoi diagram VD(P):

Definition 2.4.1. The voxel core, C, of a voxel shape O with boundary vertices P is the subset of VD(P) whose dual Delaunay elements intersect O.

The need for intersection also reflects that DT is closed with boundary while our voxel shape O is open. Please refer to Figure 2.1 to get a visual sense of voxel core.

While simply defined, our voxel core possesses all the properties one would desire for a medial axis approximation. In the following we show that voxel core is thin, homotopic equivalent to the voxel shape, and completely enclosed within the voxel shape. In addition, the voxel core is at most a quarter of voxel away from the true medial axis.

Thinness: This is a direct result from duality between voxel core and Delaunay triangulation. Though part of DT(P), all voxel vertices lie on the boundary ∂O of the shape. Therefore, they do not intersect with O. By definition, their duals, i.e. voxel cells of full dimension, are not part of the voxel core. To put it formally,

Theorem 2.4.2. C has no 3-dimensional cells.

As mentioned before, due to being non-generic, the boundary vertices P of a voxel shape are in non-generic positions. As a result, the voxel core, as a subset of VD(P), will contain singular d-dimensional Voronoi elements for d = 1, 0. To see how singular Voronoi edge can arise, consider a voxel-thick tube, i.e. a horizontal sequence of voxels one next to another. Its voxel core will be a horizontal line passing through all the voxel centers. To see how singular vertex is possible, simply consider an isolated voxel. Its voxel core is a single vertex. **Homotopy equivalence:** The topology of the voxel core is equivalent to that of its defining voxel shape as revealed by the duality. Formally speaking:

Theorem 2.4.3. C is homotopy equivalent to O.

Proof. Our proof follows the one used for Theorem 7 in [3]. We are going to utilize an intermediate structure called *nerve* to show that C captures the homotopy type of O. A nerve is a product for a family of sets, whose element is a collection of sets. The nerve is an abstract simplicial complex that captures the intersection relationship between different sets, with a k-simplex for every k sets having non-empty intersection. A result about nerve is The Nerve Theorem which states that the nerve of a family of convex sets is homotopy equivalent to their union.

Our idea then is to build a family of convex sets that exactly cover O, and show that C is topologically equivalent to the nerve of these sets. Let \tilde{C} be the Delaunay elements dual to C. Since union of \tilde{C} is equal to \bar{O} , we construct a set of modified Delaunay elements \tilde{C}' from \tilde{C} where each element is free of points in ∂O , such that $\cup \tilde{C}' = O$. Since each element in \tilde{C}' is a convex set, they form a family of convex sets, whose nerve is homotopy equivalent to their union by the Nerve Theorem. We then show C is essentially equivalent to the nerve, which requires a 1-to-1 correspondence between an elements $e \in C$ and some k-simplex (k > 0) s in the nerve, and within each correspondence e is homotopy equivalent to s. First we note the correspondence is easy to build. For every k Delaunay cells sharing non-empty boundary, there exists a dual k-sided Voronoi cell $(k \ge 3)$, an edge (k = 2), or a vertex (k = 1). The k Delaunay cells also contribute to exactly one k-simplex in the nerve. Hence the 1-to-1 correspondence exists. Second, if $k \le 3$, e will be trivially topologically equivalent to s. Otherwise, s is a higher dimensional simplex that's not as thin as e. This is not an issue because e is contained in s as a sub-complex. Since we can collapse s down to e without changing its topology, e is homotopy equivalent to s.

We have shown that C is homotopy equivalent to the nerve of \widetilde{C}' . By transitivity of equivalence, C is homotopy equivalent to O.

Proximity: Not only voxel core C preserves the correct topology of the shape O, it is also within close proximity to the true medial axis M. We show this by bounding the distance from any point on C to the closest point on M. Let h, the length of a voxel, be the voxel size. We have:

Theorem 2.4.4. For any point $x \in C$, $d(x, M) \leq \frac{1}{4}h$

Here we present a sketch of the proof. Please refer to the appendix A for details. Denote $\Gamma_P(x)$ to be the points in boundary vertices P that are closest to x, and $\Gamma_B(x)$ the set of points in $B = \partial O$ that are closest to x. We inspect how different they are. If $\Gamma_P(x)$ happen to be a subset of $\Gamma_B(x)$, then x is a point on M by the definition of medial axis - we are done. Otherwise, we look for a medial point which is not far from x. Since we have $\Gamma_B(x)$ which contains some boundary element e closest to x, and at least a boundary vertex in $\Gamma_P(x)$, they define a medial point y. Since the boundary element e can be a voxel face or edge, the proof then proceeds with each case and shows, in either case, the segment between x and y is no larger than the bound and contains a point on M.

As can be expected, the geometric bound implies that we can make the difference between C and M arbitrarily small by using sufficiently small voxels to represent a given voxel shape O. For example, we can subdivide each voxel of size h in O into $k \times k \times k$ voxels of size h/k. Then the maximal distance from C to M will be reduced to (1/4k)h. The subdivision and the resulting voxel cores are depicted in Figure 2.3 and 2.4 for a 2D and 3D voxel shape. Observe that, as more subdivision is applied, the voxel cores not only approach the medial



Figure 2.3: Subdivision of a 2D voxel shape results in the voxel core (blue) that approaches true medial axis (green).

axes closer, but also obtain a fuller coverage of the medial axes. This phenomenon leads us to conjecture that the distance from M to C is also bounded, i.e. C converges to M as $h \rightarrow 0$.

Enclosure: We conclude our discussion about voxel core by showing, just like the medial axis, the voxel core C strictly lies inside the voxel shape O. Also, we can show no Voronoi elements can be inside O other than C. The two facts lead to a simple in-core test: given a voxel shape with boundary vertices P, a Voronoi element of VD(P) is on the voxel core if and only if the vertices of that element are in the shape. To prove the property, we first show C is disjoint from B - the boundary of O. Concretely, the only elements of VD(P) that intersect B, are those that are dual to the elements in B, denoted by \tilde{B} .

Lemma 2.4.2. A Voronoi element $e \in VD(P)$ intersects B if and only if $e \in \widetilde{B}$.

Proof.

If $e \in \widetilde{B}$, then e intersects with B. Since VD(P) does not contain full dimensional



Figure 2.4: The voxel core (gray) of the 3D voxel shape gets closer to the true medial axis (green) after voxel subdivision.

element, i.e. Voronoi cells, e can only be dual to a boundary edge or face. In either case, there exists an empty ball centered on the centroid of the boundary element passing through all its vertices (see Proof of Theorem 2.4.1). Also, in light of our discussion about the empty ball property, the center of the ball must lie on the Voronoi element, here e. Hence e must intersect with the boundary element.

If e intersects with B, then $e \in \tilde{B}$. By Lemma 2.4.1, the intersection between e and some boundary element (from B) can only be close to the vertices of that element. By definition of Voronoi element, e is created due to those closest vertices, which give rise to the dual of e. Clearly, that dual is on the boundary element. Hence $e \in \tilde{B}$.

We are now ready to use the just proved Lemma to show enclosure:

Theorem 2.4.5. $C \subset O$.

Proof. Recall C is only made up of the dual of Delaunay elements that intersect O. Since boundary B doesn't intersect O, \tilde{B} is not in C. Also, by Lemma 2.4.2, only \tilde{B} intersects with B. The two facts imply C is disjoint with B, and is either completely inside or outside of O. If we show any piece of C is in O, entire C will be in O. This can be done by picking a boundary face f, which is shared by two Delaunay cells, one being outside of O, the other intersects with O. We denote the latter one t. The dual of f is a Voronoi edge \tilde{f} with two vertices, each dual to one of the just mentioned Delaunay cells. By definition, the vertex of \tilde{f} dual to t is on C. Also, that vertex is inside O, since \tilde{f} intersects B only once (by Lemma 2.4.2 and a Voronoi element at most intersects its dual once). Hence, we've shown C is inside O. As mentioned earlier, we have a simple in-core test based on the assumption that C is equal to the inside subset Voronoi diagram VD(P), i.e. any Voronoi element of VD(P) is inside O if and only if it is in C. Note we just proved sufficiency. Now we show necessity, i.e. the inside Voronoi elements must be in C. The only worry one might have is some inside elements might be dual to the non-boundary Delaunay elements which do not intersect O. We denote those Delaunay elements by E. However this is not possible. Reusing the above proof, we can show the dual of E are disjoint from B, and lie completely outside of O following the same analysis about one Boundary face.

2.5 Approximating Medial Axis of Smooth Shapes

Aside from voxel shapes, another class of shapes is smooth shapes which are C^2 continuous everywhere. They are the basis on which many sampling-based medial axis methods derive their theoretical guarantees. Therefore, in this section, we also consider applying voxel core to approximate the medial axes of smooth shapes. We will quantify the approximation ability both topologically and geometrically. The analysis is broken down into two parts. First, we convert a smooth shape to a voxel shape via the voxelization process, and give the conditions on voxel sizes under which the voxel shape can well approximate the original shape. Then the voxel core of the voxel shape is used to approximate the medial axis of the smooth shape, for which we show topological correctness and geometrical convergence with a set of conditions.

Voxelizing smooth shape: We make explicit a few notations as illustrated in Figure 2.5. Let O be the given smooth shape bounded by a C^2 continuous manifold surface B. The reach r of O [20] is defined to be shortest distance between any point on B to the medial



Figure 2.5: Voxelization converts a smooth shape O bounded by B into a voxel shape O_h bounded by B_h . O_h consists of voxels of size h whose centers are in O.

axis of O or the complement $\mathbb{R}^n \setminus \overline{O}$, also known as the local feature size [2, 23]. We use Gauss digitization [43] to voxelize O at voxel size h. The resulting voxel shape O_h consists of voxels whose centers are inside O. The boundary of O_h is denoted by B_h , and vertices of B_h are referred to as P_h .

First, we can bound the distance between the voxel shape and the original smooth shape. We show two bounds, one between the voxel shape boundary vertices and the smooth boundary, the other between the open sets representing voxel and smooth shape.

Theorem 2.5.1. For any $h < \frac{2\sqrt{3}}{3}r$, (1) $d_H(P_h, B) < \frac{\sqrt{2} + \sqrt{3}}{2}h$ (2) $d_H(O_h, O) < \frac{\sqrt{3}}{2}h$

Please refer to the appendix for the proof. In other words, if the voxel can fit in a sphere of radius equal to the smooth shape's reach, the voxel shape produced and the smooth shape will be close to each other. Next, we show voxelization also captures the topology of the smooth shape, whenever the voxels are sufficiently small.

Theorem 2.5.2. For any $h < \frac{\sqrt{3}}{3}r$, O_h is homotopy equivalent to O.

The proof is given in the appendix, which is based on a result by Stelldinger et al. [77] that establishes the requirements of a topology preserving reconstruction from a voxel shape. Together the two theorems guarantee a topologically correct and geometrically close voxel shape can be obtained for a smooth shape via voxelization with sufficiently small voxels.

Approximating MA of smooth shape: Knowing voxelization can be done with theoretical guarantees, and based on the previously obtained results about voxel core, we move on to show the voxel core well approximates the true medial axis of the smooth shape. Topology wise, we will show the voxel core captures the topology of the smooth shape. Geometry wise, we will show a stable subset of the voxel core converges to a stable subset of the true medial axis.

Let C_h be the voxel core of the voxel shape O_h produced by the voxelization of a smooth shape O at voxel sizes h. By Theorem 2.5.2, when h is small enough, O_h is homotopy equivalent to O. Since C_h is homotopy equivalent to O_h by Theorem 2.4.3, it follows that:

Theorem 2.5.3. C_h is homotopy equivalent to O if $h < \frac{\sqrt{3}}{3}r$.

The geometric convergence of voxel core relies on a concept called λ -medial axis [19]: while the medial axis is notoriously sensitive to perturbation on the boundary of its defining shape, its λ -medial axis is stable. Chazal and Lieutier define λ -medial axis as the medial points whose nearest boundary points can only fit in a sphere of radius greater than λ . They show the λ -medial axis of the perturbed shape is within close proximity to that of the original shape as long as λ is sufficiently big.

We are particularly interested in a theoretical result of λ -medial axis. Chazal and Lieutier define a ϵ -noisy sample of a surface B as a point set P whose Hausdorff distance with the surface is less than ϵ . Then they are able to show as ϵ goes to zero, the λ subset of the Voronoi diagram of P, denoted as $VD_{\lambda}(P)$ converges geometrically to the λ -medial axis, M_{λ} of the shape O bounded by B. It is hopeful that we can make use of this result because the boundary vertices of our voxel shape, P becomes a ϵ -sample of the surface of O as long as voxel size h is small enough. Indeed, we can show the λ subset of C_h , $C_{h,\lambda}$ converges to the λ subset of the true medial axis of O, M_{λ} .

Theorem 2.5.4. For any $\lambda > 0$ such that the mapping $M(\lambda) = M_{\lambda}$ is continuous at λ , and any sequence $\{h_n\}$ such that $\lim_{n\to 0} h_n \to 0$,

$$\lim_{n \to 0} d_H(C_{h_n,\lambda}, M_\lambda) = 0$$

Proof. It's clear that when h_n eventually gets below $\frac{2\sqrt{3}}{3}r$, P_{h_n} , the boundary vertices of O_{h_n} , will have a bounded distance from O (Theorem 2.5.1), thus becoming an ϵ -sample of O. To use the result by Chazal and Lieutier, we want to show for any given λ , there exists a sufficiently small h_n in the sequence $\{h_n\}$ such that $C_{h_n,\lambda}$ is exactly the part of λ subset $VD_{\lambda}(P_{h_n})$ inside O. Since $C_{h_n,\lambda}$ is also the subset of $VD_{\lambda}(P_{h_n})$ in O_{h_n} , we turn to show for $VD_{\lambda}(P_{h_n})$ its part inside O is equivalent to its part inside O_{h_n} , when $h_n < \lambda/(\sqrt{3} + \sqrt{2}/2)$ for a given λ .

We proceed by contradiction. Suppose there exists a point $x \in VD_{\lambda}(P_{h_n})$ inside O but outside O_{h_n} . Then x's nearest vertices in P is enclosed in a sphere of radius greater than λ . By Theorem 2.5.1 (2) there is a point $y \in B_{h_n}$ such that $d(x, y) < h_n \sqrt{3}/2$. Also, among P the closest vertex $z \in P_{h_n}$ belongs to the boundary element containing y and forms distance $d(y, z) < h_n \sqrt{2}/2$ (diagonal of a voxel face). Therefore all the closest boundary vertices for x will fit in a sphere of radius smaller than $h_n(\sqrt{3} + \sqrt{2})/2$, meaning $h_n > 2\lambda/(\sqrt{3} + \sqrt{2})$. Hence the contradiction.

Similarly, suppose the other direction is not true, i.e. there exists a point $x \in VD_{\lambda}(P_{h_n})$ inside O_{h_n} but not in O. Again due to Theorem 2.5.1 (2) there is a point y on B whose distance to x is $d(x,y) < h_n\sqrt{3}/2$, and for y we can find a point $z \in P$ with $d(y,z) < h_n(\sqrt{3} + \sqrt{2})/2$. It follows that the smallest sphere enclosing the nearest boundary vertices for x has a radius smaller than $h_n(\sqrt{3} + \sqrt{2}/2)$, leading to $h_n > \lambda/(\sqrt{3} + \sqrt{2}/2)$. Hence the contradiction.

In summary, given λ and $h_n < \min(r2\sqrt{3}/3, \lambda/(\sqrt{3} + \sqrt{2}/2)), C_{h_n,\lambda}$ equals the part of $VD_{\lambda}(P_{h_n})$ inside O. By the convergence result, $C_{h_n,\lambda}$ converges to M_{λ} as h_n goes to 0 \Box

The theorem states that λ specifies how large the subset of the true medial axis our voxel core can converge to. By making λ smaller, the voxel core will to converge to a larger subset of the true medial axis. When λ reaches 0, the voxel core will be equal to the true medial axis. On the other hand, the voxel core converges to a smaller subset of the true medial axis if λ is large. In theory, we will always have a sequence of voxel size that goes to 0. Therefore reducing λ is always desirable. Whereas as we will see in the results section, λ serves as a balance in practice between approximation accuracy and computational cost. Due to the unique and prevailing stair-case artifacts introduced by voxelization, we need to set λ to be sufficiently large to prune away the corresponding noisy branches on the voxel core. In the meantime, λ cannot be too large otherwise the remainder voxel core will not be able to capture important features that are thinner than λ . However if we pick a small λ in hope to approximate a better coverage of the medial axis, a great number of voxels of even smaller sizes are required in light of the conditions we have on voxel size which is upper bounded by a fraction of λ .

2.6 Algorithm

Our simple definition and the highly-practical in-core test enables a 3-step algorithm.

Step 1: Voxelization This step implements the voxelization process described in sec 2.5. If the input is not a voxel shape, e.g. a triangular mesh, it will be converted to a voxel shape made up of voxels with centers inside the shape. Otherwise, the input shape is directly passed to the next step.

Step 2: Extraction of voxel core This step exactly computes the voxel core as defined in Definition 2.4.1 of the voxel shape obtained directly or indirectly from the input. First the voxel vertices on the boundary of the voxel shape are obtained. Second, the Voronoi diagram of this point set is computed. Third, using the simple in-core test implied by Property 2.4.5, the voxel core is easily extracted as the subset of Voronoi diagram inside the voxel shape.

Step 3: Pruning Finally, we output a subset of the voxel core which mimics the λ subset as much as possible. Specifically, we explicitly ensure topology using a topology preserving contraction operation to prune away as many as possible the elements whose nearest boundary voxel vertices can be enclosed in a sphere of radius less than λ .



Figure 2.6: Steps for approximating the medial axis of a hand shape represented as triangle mesh (a). We first voxelize the shape (b). We then extract the voxel core (c) with each point colored by the radius of the smallest sphere enclosing its nearest boundary vertices (blue and red for small and large radii), and prune it to its λ -subset (d)

The steps are illustrated in Figure 2.6. The most complex procedures of our algorithm are voxelization and Voronoi diagram computation. Both of them are fairly standard computational modules for which existing packages are available. Due to this reason, our algorithm is simpler to implement than existing sampling-based medial axis approximation methods. Particularly, the Voronoi diagram can be computed more robustly in our context since the input points are representable in integer coordinates.

Implementation details For a non-voxel shape represented as polygonal mesh, we use Polymender [38] as the voxelization tool. It is very efficient in both time and memory, tolerant to mesh defects (e.g. holes, self-intersections), and outputs a compact octree an optimized voxel shape representation that can reach high effective voxel resolution (e.g. 4098^3). In step 2, we obtain the Voronoi diagram as the dual of the Delaunay triangulation of the boundary voxel vertices using Tetgen [73], a highly efficient and robust tetrahedralization tool. As mentioned earlier, we represent all boundary voxel vertices using integer coordinates to maximize the robustness of Tetgen. In step 3, we perform the contraction process of [47] from the boundary of the remainder of the voxel core iteratively. We keep removing, without changing the topology, those faces and edges 1) that intersect the boundary of the remaining voxel core, and 2) whose closest boundary vertices can fit in a sphere of radius smaller than specified λ , until no such elements can be identified.

Complexity The complexity of voxelization using Polymender depends on the depth d and the number of boundary vertices |P| of the output octree, and the number of polygonal faces in the input mesh m. The process takes O(d(m+|P|)) time and O(|P|) memory. Though the time and space of Voronoi diagram could take $O(|P|^2)$ time in the worst case, the complexity is much less, i.e. O(|P|) in time and memory for well-distributed points on a surface [7]. The time and memory of the rest of Step 2 and Step 3 are proportional to O(|P|).

2.7 Results

We evaluate our method with different kinds of input shapes, including voxel shapes, smooth shapes, and meshes. We first compare with state-of-the-art voxel-based methods for scalability evaluation. We then compare our robustness with sampling-based methods. Lastly we showcase quality medial axes results for meshes. For all experiments λ is specified assuming the shape fits in a unit cube. We use a workstation with 3.47GHz CPU and 24GB memory to collect all the timing and memory statistics.



Figure 2.7: The performance of our method (and the Tetgen component), HJ [74] and JST [37]. 5 shapes are voxelized at 10 resolutions (128^3-1280^3) . The memory (top) and time (bottom) consumption is plotted against the number of boundary voxel vertices (horizontal axis) for each shape at each resolution. HJ and JST both terminate early due to lack of memory (+) and crashing(*), while our method finishes processing all inputs.

2.7.1 Voxel shapes

We first evaluate the scalability of our method using five smooth shapes (Ellipsoid, Mug, Elk, Hand, and Fertility) represented as very high resolution triangular meshes, all featuring enough variation in topology and geometry. Each shape is voxelized at 10 resolutions, from 128^3 to 1280^3 with an increment of 128. Then the resulting voxel shapes are fed to our algorithm (Figure 2.7). Assuming each shape is scaled to fit in a unit box, we use $\lambda = 0.015$ for Elk shape to capture its thin ears, and $\lambda = 0.025$ for other shapes. As can be seen both time and memory usage of our method scales linearly with the number of input boundary vertices |P|. All inputs can be processed by our method using under 10GB memory, and most time is spent on computing the Voronoi diagram (using Tetgen). We compare with two state-of-the-art voxel based methods. One is JST, the recent work of Jalba et al. [37], another is the Hamilton-jacobi skeleton [74], shortened as HJ. Both methods use Euclidean distance transform to realize the grassfire analogy as close as possible, which is crucial to ensure centerdness and thinness for approximating medial axis as voxels. Observe that the time and space complexity of HJ and JST exhibit super-linear growth since they need to process all uniform voxels within the shape. As a result, they can only process voxel shapes of resolution up to 640^3 before running out of memory.

We also demonstrate our performance on a challenging real-world biomedical dataset in Figure 2.8. The input is a CT scan of 4-week old corn roots at the resolution of $1560 \times 789 \times 1041$. The roots manifest complex branching structures and geometric properties which can be better visualized and studied with a good approximate medial axis. As depicted by the figure, our method is capable of processing this input under one minute using only 1.5GB memory. The resulting voxel core makes it possible to compute a topologically correct curve skeleton using the medial axis regularization method to be presented in Chapter 3. The skeleton better reveals the roots structure using curves that are much cleaner than the surface of medial axis. The skeleton is also equipped with a measure that reflects the local thick of the roots. Such capability enabled by our method is crucial to biologists to conduct shape analysis on the root system architecture (RSA). Note the other two methods (HJ and JST) are not able to process this voxel shape because it's impossible to store the entire volume with voxels of same size in memory as required by the methods.



Figure 2.8: For a high resolution CT scan of corn roots (top), the medial axis is computed using our method (middle), which is then fed to Erosion Thickness (to be presented in Chapter 3) for a clean curve skeleton (bottom). Each point of the skeleton is colored by the local thickness (radius of medial ball) of the shape (blue:thin, red:thick). Two regions are zoomed in to show how a good medial axis helps with revealing complicated structures.



Figure 2.9: Different λ voxel cores (columns) are extracted for Ellipsoid shape voxelized at 4 resolutions (rows) to show how λ affects geometric convergence of medial axis. The true medial axis at top-left is highlighted in red. Each point of full voxel cores is colored with radius of minimal enclosing sphere of the nearest boundary voxel vertices.

2.7.2 Smooth shapes

Next we present results for smooth shapes. First we discuss practical impacts of choices of voxel size h and λ using the smooth shape of Ellipsoid. In figure 2.9, the shape is voxelized at resolution 128³, 256³, 768³, and 1280³. For each resolution three λ -subsets are inspected, with $\lambda = 0.015, 0.025, 0.065$. We observe a pattern conforming with Theorem 2.5.4, i.e. λ controls the trade-off between accuracy and efficiency. When λ is small, voxel core converges to a larger subset of the true medial axis, achieving more accurate approximation. However, for such a simple shape, noises due to stair-case artifacts of voxelization won't disappear until voxel size h reduces to sufficiently small at resolution 1280³. On the contrary, big λ worsens approximation quality, e.g. the rim of the medial axis is missing, but allows convergence to begin as early as resolution 128³.

Second, we show how the topological and geometric quality of our results compare with three sampling-based methods: the method by Dey and Zhao (DZ) [23], the power crust method (PC) [2], and the union-of-ball method (SAT) [50]. We use the 4 out of 5 smooth shapes as used in the scalability test. For our method, we voxelize each shape into voxel shapes at 3 different resolutions, and set λ in the same shape dependent way as the scalability test. For other methods, we use their default settings, and set the scale factor s = 1.0 for SAT to turn off its regularization function. The intermediate point samples of the smooth shapes generated by SAT are used as input to PC and DZ.

From Figure 2.10 we see our results achieve similar visual quality than the other methods starting from resolution 512³, at a higher time and space cost since we need more samples. Nevertheless, a key advantage of our method is topological robustness. Notice that DZ tends to generate many small islands, and PC produces numerous closed "pockets", as reflected in



Figure 2.10: Comparison with sampling-based methods, SAT [50], PC [2], and DZ [23]. Next to each medial axis we show: the time and memory. Topology information is reflected in a pair of Euler characteristics and number of components, where incorrect numbers are marked red. Two pairs are shown before and after duplicate faces are removed for methods other than ours.



Figure 2.11: Comparing with SAT on a smooth shape with thin features. SAT requires extremely high sampling rate and computational cost, but still produces many holes. Our method is able to capture the topology at resolution 256^3 and similar level of geometry at 512^3 with much lower cost.

the incorrect Euler characteristics and number of connected components. Although SAT is backed-up by theoretical guarantees to generate topologically correct and thin medial axes, in practice we have observed a large number of duplicate faces and occasional closed pockets (see Elk and Fertility). While it's easy to remove duplicate faces in a post-process, pockets cannot be trivially eliminated without affecting topology. In contrast with these methods, we are able to deliver the theoretical promise of topology preservation robustly in the results.

We've also noticed the efficacy in capturing the right topology with our sampling strategy of using voxel vertices on boundary when delicate features are present in the input shape. We feed a mathematically defined shape with thin connections between four blobs to SAT and our method, and compare the results in Figure 2.11. SAT needs a lot of samples (sampling distance 0.002) to preserve the thin features in the medial axis with expensive time (30min) and memory (10GB) cost, but at the same time produces many holes. However our method



Figure 2.12: Approximating the medial axes of mesh shapes (at resolution 1024^3).

captures the correct topology starting from resolution as low as 256^3 and the voxel core achieves a comparable visual quality at the resolution of 512^3 ($\lambda = 0.03$).

2.7.3 Meshes

Finally, we present results for input represented by triangular mesh. Our theoretical properties are not guaranteed to hold in practice because they would require infinitely small voxel size for polygonal meshes which often have reach of size 0. Even so, we have observed our method performs well on a set of triangulated meshes. Figure 2.12 shows results for a subset of the meshes we test on. The voxel core of each shape captures the right topology and most important geometric features. All the results are computed at resolution of 1024^3 with $\lambda = 0.025$ at the cost of less than 3 minutes and 5GB memory, while voxel shapes of such scale are not possible to process for voxel-based methods like HJ and JST.



Figure 2.13: Approximating the medial axis a Neptune triangle mesh at two fine resolutions. For shape as complex as Neptune in Figure 2.13, a small λ is necessary to preserve fine features like the trident, which requires even finer voxels for topology preservation. Because of the prominent efficiency, our method can still afford the computation with reasonable cost even at resolution 2048³, which produces smoother results than at resolution 1024.

2.8 Summary

In this chapter, we introduce a new medial axis approximation, called voxel core. It features many desirable properties just like the true medial axis. For voxel and smooth shapes, their voxel cores well approximate the true medial axes with both topological and geometrical guarantees. The solid theory leads to a medial axis approximation method that's easy to implement, numerically robust, time and space efficient, and works well for voxel and smooth shapes, and even for regular triangle meshes commonly seen in the graphics community.

Chapter 3

Pruning of Medial Axis

3.1 Background and Motivation

As a result of last chapter, we are able to approximate medial axis robustly and efficiently. However, we learn during the introduction that medial axis is extremely sensitive to noise on the shape's boundary. For example in Figure 3.1, small scale noises lead to many spurious branches that obscure the major structure of the medial axis of a Dolphin shape. In order to tame such sensitivity, a majority of research works focus on pruning branches that arise from boundary noise under the guidance by some form of significance measure. A significance measure rates the importance of each medial axis point using the shape feature around it. Two types of significance measure exist, i.e. local and global.

Local measures are defined based on the close-by geometry around each medial point. Therefore they don't always work and situations exist where they can fail to remove noise or preserve important subset. Nevertheless, local measures are popular for their intuitive and simple definition, and can be computed with small overhead. On the other hand, global measures consider more shape information to rate a medial point. Hence they outperform



Figure 3.1: The medial axis of a bumpy 3D shape contains numerous spurious sheets. Erosion Thickness (ET) highlights parts of the medial axis that represent significant shape features. We use ET to create a family of clean, topology-preserving skeletons made up of both 2D sheets and 1D curves.

local measures in terms of isolating noise from main features. However it is difficult to define a global measure, and the computation is usually more expensive than local measures.

It is even more challenging to formulate global significance measures on medial axis in 3D than in 2D due to the increasingly complex structure. The only well-defined global measure that we are aware of is the Medial Geodesic Function (MGF) [22], which measures the geodesic distance between the two closest points on the shape boundary for a medial axis point. However, MGF is expensive to compute due to many geodesic distance evaluation on the surface. So far, well-defined ET in 3D is unknown.

Contributions In this chapter, we define, analyze and compute Erosion Thickness on the medial axis of a 3D shape, which we believe is the first well-defined and efficient-to-compute



Figure 3.2: Local vs global measures. Top: Medial axes of the same 2D shape colored (blue for low and red for high) by the object angle measure, the circumradius measure, and Erosion Thickness. Bottom: Subsets of the medial axes where the respective measure is higher than some threshold. Local measures, such as object angle and circumradius, fail to differentiate major shape features (e.g., the horse legs) from boundary noise (e.g., hair).

global significance measure in 3D.

Erosion Thickness in 3D definition. Liu et al. [48] define ET in 2D as the difference between the arrival time of a fire front propagating over the medial axis and the radii of maximal balls. Inspired by the same idea, we introduce a burning process over the medial axis in 3D. Like its counterpart in 2D, our ET in 3D properly highlights major features of a medial axis (Figure 3.1).

Efficient approximation algorithm We develop an efficient algorithm that approximates ET in 3D on a piece-wise linear medial axis with bounded approximation error.

Skeletonization algorithm Guided by Erosion Thickness, we compute a family of skeletons, each as a subset of medial axis combining both 2-dimensional sheets with high ET values and 1-dimensional curves that follow the ridges of ET. Two intuitive parameters control the pruning of the sheets and curves independently. The skeletons preserve the topology of the 3D shape and are stable under boundary perturbations (Figure 3.1).

3.2 Related Works

In this section, we first briefly review the research in the domain of medial axis simplification. Then we focus on the works most relevant to ours.

Simplification There are three general approaches to de-noising medial axes. The first approach *smoothes* away small scale bumps on the shape boundary. The smoothed shape gives rise to a less complicated medial axis [24, 62, 31]. An undesirable side effect is that smoothing could introduce changes to close shape features, resulting in different topology than the original shape [50]. The second approach relies on *template* deformation to fit prescribed medial axis into a group of similar shapes [33, 61, 60]. Though useful in biological shape analysis, this approach is limited to shapes with pre-determined structures. Our method adopts the third approach, which *prunes* away parts of the medial axis suggested as noise by a significance measure. During the pruning process, a contraction scheme is often employed to maintain the right topology.

Local Measures In local measures, the significance is defined by the local configuration of a medial axis point and its nearest neighbors on the shapes boundary. For example, object angle measures the angle spanned by the vectors from a medial axis point to its two nearest neighbors on the boundary [9, 2, 23, 28, 79]. Measures related to object angle include the propagation velocity of [13], the outward flux of [74], and the stability ratio of [46]. Another measure considers the circumradius of the nearest neighbors on the boundary [18, 17]. The subset of the medial axis where the circumradius is above a constant λ (known as the λ -medial axis) is provably stable under certain class of boundary perturbations. The main drawback of local measures is that they cannot capture the size of features (Figure 3.2). Being scale-ignorant, object angle is high both within prominent features and inside small boundary bumps (e.g., horse hair). Object angle is also low in the transition area between shape parts (e.g., between horse leg and body). On the other hand, circumradius only captures thick parts of the shape and easily misses long but thin features (e.g., horse leg).

Global Measures In global measures, a larger neighborhood is considered to rate a medial point. Several global measures exist in 2D, among which Erosion Thickness (ET) is a popular one, which evaluates how much the shape shrinks after the medial axis branch is shortened [35, 14, 53, 8, 5, 71, 48]. Therefore, ET can differentiate boundary noise from long but thin features (Figure 3.2 top-right). ET also has several desirable properties. In particular, ET has no local minima on the medial axis. As a result, thresholding ET never disconnects the medial axis (Figure 2 bottom-right). Other global measures in 2D that capture shape loss include the area of erosion [71, 8] and the Potential Residue [54, 55]. However, global measures in 3D are rare. As far as we know, Medial Geodesic Function (MGF) [22] is the only well-defined global measure in 3D which is extended from the Potential Residue measure in 2D [54]. Unfortunately, as we shall see later, MGF is sensitive to geodesic perturbation on the shape. Though ET is expected to be more robust, we are not aware of an extension in 3D.

3.3 Erosion Thickness in 3D: Definition

In this section, we first give a brief review of ET in 2D. We then introduce our measure by extending ET from 2D to 3D.

3.3.1 Review: ET in 2D

Denote M the medial axis of a 2D shape S, with R(x) being the maximal ball radii at each point $x \in M$. The importance of a medial point is indicated by how much the shape has shrunk after eroding the medial axis from its end points. The remaining shape is reconstructed by the union of the maximal balls centered on the remaining part of medial axis. In this picture, erosion of M starts from the end y and stops



at medial point x. The shape loss, i.e. erosion thickness w.r.t. x is measured by the length of the red curve: $R(y) + d_M(x, y) - R(x)$, where d_M is the distance on M. Note that ET is small if y lies inside a small boundary bump, where the lost portion would be thin (Figure 3.3 top).

Evaluating ET has been based on several heuristics [71], as multiple ends could be considered as y in d_M , until Liu et al. gave a precise definition in [48] by extending Blum's burning



Figure 3.3: The burn time (BT, left), maximal ball radii (R, middle), and Erosion thickness (ET, right) in a 2D ellipse (top) and a 3D ellipsoid (bottom) with small bumps.

analogy (Figure 1.3, right). Recall as Blum's grassfire quenches, the medial axis forms. Now imagine, as soon as an end point of M forms, we allow another fire starts from that point. Again, the fire moves at unit speed along M. It dies out at a junction point unless there is only one remaining branch, in which case the fire continues along that branch. Finally multiple fronts meet and quench each other. This burning process gives rise to a burn time function over the medial axis, BT(x), which computes $R(y) + d_M(x, y)$ for $x \in M$. ET is then:

$$ET(x) = BT(x) - R(x).$$

Note in Figure 3.3 (top) the subtraction correctly suppresses the importance of the branch corresponding to the small bump. In fact, ET in 2D reflects tubularity of the shape.

3.3.2 Defining ET in 3D

Similar to 2D, Erosion Thickness in 3D corresponds to how much the 3D shape has shrunk after its medial axis is eroded. Inspired by Liu et al. [48], we define ET in 3D as the difference between the burn time of a fire over the medial axis (of the 3D shape) and the maximal ball radii (as depicted in Figure 3.3, bottom). While ET in 2D reflects the tubularity of shape, ET in 3D indicates the plate-likeness. High ET values appear at the center of wide and thin plates, where the difference between the lateral distance (i.e., burn time) and the vertical distance (i.e., maximal ball radius) to the shape boundary is large. In defining ET with such behavior, the main challenge is the formulation of burning on the complex structure of medial axis in 3D.

Motivation Intuitively, we want to extend the burning rules to medial axis in 3D as the following:

1. Fire starts from the boundary of medial axis at the time equal to the maximal ball radius. Recall in the 2D case the burn time reflects the distance from a medial point x to the shape's boundary, which includes not only the fire's travel distance on the medial axis, but also the distance from an end to the shape. With the start time equal to the radius function we are able to model this behavior.

2. Burning stops at a seam line or point if multiple sheets remain there. This is a generalization of the die-out rule at a junction point in 2D, which is critical to prevent burning a hole from the middle of a sheet.

3. At last fire fronts quench each other when they meet.

We next illustrate the new burning process in Figure 3.4. Here, shape S consists of a "fin" on top of a thicker and wider "board". Correspondingly, the medial axis M has a fin sheet and a board sheet joining along a seam. Fire starts from boundary at an earlier time on the fin sheet than on the board sheet due to rule 1. Fire on fin sheet dies out as it reaches the seam line on the main board due to the die-out rule (back view of time T1). Interestingly, in this particular geometric configuration, fire front on the board sheet catches up with that


Figure 3.4: Burning on the medial axis of a 3D shape (shown in the top-left) at five time points and at completion.

on the fin at point p (back view of T2). Notice p has the type of neighborhood equivalent to x_6 as depicted in Figure 1.4. The front is immediately divided into two segments A and B at p. The portion A merges with one front on the fin sheet, against which portion B dies-out (back view of T3). Later on, two fronts merge into one (at point q at time T4) and finish burning the rest of the board.



Our definition essentially realizes the above burning process. To better motivate it, consider a one-sheet medial axis, i.e. the one in the left figure without the top fin sheet. In this case burning can be fully described by only rule 1 and 3. In fact, for a medial point x, the fire trajectory corresponds to the shortest path γ to boundary ∂M , and burn time is simply defined by its length, i.e. geodesic distance. Now add back the fin sheet which is shorter and narrower than the board. Suppose x is burned away by the fire front on board sheet along a path whose intersection with the seam line is point z. According to rule 2, the fin sheet portion at z must have been burned by another fire along some path γ' before γ arrives at z. Otherwise, γ would die out at z and not be able to reach x.

This motivation shows path tree is a good model for encoding the burning rules. For a point $x \in M$, its path tree is rooted at x with leaves on ∂M . More importantly, the tree must have an interior vertex if it passes through a seam line on M, and must spawn one branch onto each sheet. This implicitly captures the burn path on other sheets that die-out at the seam, encoding rule 2. Note a path tree degenerates to a single path if it lies on a single sheet. To define burn time, we aim to find the "shortest" path tree at x. We first define root-to-leaf length, which is the distance between root to a leaf plus from leaf to the shape boundary, encoding rule 1. We then define the length of a path tree, which is simply the maximal root-to-leaf length among all leaves. Finally, among all valid path trees at x, we call the one with the minimum length the shortest path tree, whose length captures the burn time, BT(x). This way we naturally generalize geodesic distance when M contains only one sheet.

Definition of burn time We now proceed to the formal definitions of the idea of path tree and burn time. Recall in the discussion regarding the topology of the medial axis 1.3.1, the local neighborhood N(x) of a point $x \in M$ consists of one or more sectors, each being equivalent to a full or half disk. For instances, a manifold point has a single disk-shaped sector, while a singular point has multiple half-disk sectors.

Definition 3.3.1. We call a point **exposed** by an **exposure set** of sectors $\{s_1, \ldots, s_k\}$ if the removal of them results in no full disk in N(x).

We then formally name a path tree exposing tree

Definition 3.3.2. A tree on M is an **exposing tree** if every interior vertex is exposed as a boundary point by the sectors containing its child edges.

This definition implies the boundary points, ∂M have empty exposure set, which naturally captures the fact that fire starts from the boundary spontaneously. Length of the tree is defined as length of the longest path, called the primary path.

Definition 3.3.3. The **length** of an exposing tree Γ at $x \in M$ is the maximum length among all rooted paths from x to a leaf vertex $y \in \partial \Gamma$

$$L(\Gamma) = \max_{y \in \partial \Gamma} (d_{\Gamma}(x, y) + R(y))$$

where d_{Γ} is the path distance on Γ , R is the radius function over M. The path realizing the length from x to some leaf y is called the **primary path** of Γ .

Finally, the burn time of a point is given as length of the shortest exposing trees at the point. **Definition 3.3.4.** Let \mathcal{T}_x be the set of all exposing trees at $x \in M$. The **burn time**, $BT: M \to \mathbb{R} \cup \{\infty\}$ is

$$BT(x) = \inf_{\Gamma \in \mathcal{T}_x} L(\Gamma)$$

The exposing tree that realizes the burn time is called a **burn tree**.

Note we use *infimum* instead of minimum because at some points the limit is not achievable. See Figure 3.5 for an example. There, the shortest length of the path is achieved when the path goes through the red part of the singular curve. That will force another path to spawn onto the cylinder to make it an exposing tree. However, the length of the new exposing tree will be significantly longer than the original path. Hence the burn time cannot be realized exactly, and can only be captured by the infimum.



Figure 3.5: A 2D example where for x the burn time, i.e. the length of the shortest path can only be realized in the limit as the path cannot include the red part of the cylinder base. Otherwise the path needs to fork onto the cylinder, creating a much longer exposing tree, thus failing to capture burn time.

With BT in place, ET in 3D is defined as the difference between BT and the maximal ball radii, R:

$$ET(x) = BT(x) - R(x).$$

Similar to the 2D counterpart, our definition of ET in 3D possesses a set of nice properties that can be explained intuitively from the burning perspective. We give the proofs in the Appendix B.

Finiteness ET is finite for almost any $v \in M$, as long as a fire can reach it from ∂M . One exception is that points on a closed surface cannot be reached by any fire due to lack of boundary, leading to infinite burn time. Therefore ET on closed surfaces is infinite.

Continuity ET is continuous away from seam lines as fire on M propagates in a continuous way. Even along seam lines, ET varies continuously on the disk of a point burned by the last fire front.

Local minima Interior to M, ET is free of any local minima at points away from ∂M . Intuitively, this is because fire only starts from boundary and propagates inward, giving points more interior higher burn times. Also, in an infinitesimal neighbor of any point that can be burned (away from closed surface), there exists at least some direction along which the radius function changes no faster than the burn time. Together they imply ET at the point decreases along that direction outward, showing the point is not a local minimum.

3.4 Erosion Thickness in **3D**: Computation

We have designed an efficient algorithm to approximate ET on a piece-wise linear medial axis M represented as a triangular mesh equipped with the radius function R defined at each vertex. Our output is approximated ET values defined at vertices. To do this discretely, we construct a graph $G = \{N, A\}$ to approximate M. We then compute ET on this graph. The core of the algorithm is computing burn time, for which we modify the classical Dijkstra's shortest path algorithm to adapt to the non-manifoldness of M.

3.4.1 Graph construction

Since M induces a graph structure naturally, one could simply use it as G where nodes N contains all triangle vertices and arcs A is triangle edges. Instead, we take a graphsubdivision approach inspired by Lanthier's method [44]. In addition to the original vertices of M, we allow each edge to be refined by inserting extra nodes on the edge. User controls sampling rate by specifying the maximal distance between two nodes on the same edge is lower than a value ω . Doing so effectively increases accuracy of the computed ET. More importantly, we can bound the error of approximating ET to be proportional to the distance between graph nodes, which is in turn bounded by ω .



Figure 3.6: Illustration of graph used for burning and dualization. (a) A refined graph on a triangle (Solid dots and hallow dots: nodes on vertices and edges. Red and blue: edge-arcs and face-arcs). (b) Between adjacent nodes on an edge along a singular curve (orange), there is one edge-arc on every face sharing the edge (grouped by same color). (c) The primary arcs (blue and red) induce a subdivision on the triangle, whose dual (green) is a polyline network with vertices at centroids of non-primary arcs (squares) and empty regions (dots) and edges between them.

Figure 3.6 (left) shows the refined graph for a triangle face. Within each triangle face, a node can be connected to an adjacent node on the same edge via an edge-arc, or to a node on the opposite edge via a face-arc. In the end, a pair of adjacent nodes on a non-manifold edge will be connected by multiple arcs, each on a triangle sharing that edge (Figure 3.6, right). This is the key that sets our refined graph apart from a traditional one, which is crucial to model multiple paths that are infinitely close to the seam while on different sheets. Such case does not arise in manifold region.

3.4.2 Graph-restricted ET and error bound

Before we start computing ET on G, we need to re-inspect the definition of ET. Originally ET is defined in a continuous setting which carries over to the input piece-wise linear M.

Now we only work on the graph G which is a strict subset of M. Fortunately, we only need to modify the definition of the burn time (Definition 3.3.4) by replacing infimum with minimum.

In particular, we are able to show the modified ET_G restricted on graph G is no smaller than the true ET, and is upper bounded by a quantity related to the true ET.

Theorem 3.4.1. Let |M| be the number of triangles in the input triangle mesh M, g be the maximal gradient magnitude of R on any triangle edge on ∂M , and ω be the maximal distance between adjacent nodes of G on a triangle edge. For any node v in G, its ET restricted on G, ET_G is bounded:

$$ET(v) \le ET_G(v) \le ET(v) + (2|M| + g)\omega$$

The proof is given in section B.2 of Appendix B.

3.4.3 Algorithm: burning the medial axis

Recall in the burning analogy, fire starts from boundary of M and always advance along shortest path. Hence, to compute burn time BT, we simulate the burning on G by computing shortest path tree. We resort to Dijkstra's shortest path algorithm. Most notably, we generalize Dijkstra's so that it propagates distance on a non-manifold according to our burning rules.

Algorithm 1 illustrates how we conduct the burning process. Notably, the algorithm computes the graph restricted ET values exactly at each vertex. We use a priority queue Q to store unburned nodes, ordered by their burn time non-decreasingly. Initially all nodes are unburned with burn time set to infinite. Only boundary nodes carry finite burn times equal to their radii. This aligns with burning rule 1. During each iteration of the burning loop, the node with smallest *estimated* burn time in Q is popped. Though it is most likely to pass on a fire to the neighbors, the node must be burned completely without any remaining sheets. This is realized by burning the primary sector and other sectors thus made open, and then check what's left.

— On one hand, if the node is indeed burned, fire will be propagated forward along every arc to the neighbor on the other end who will probably perform updates on two levels. On the sector level, the neighbor receives update via a specific sector s, whose *primaryArc* will record where the fire comes from if it gives a better estimated burn time. On the node level, the neighbor's primarySector and the best estimated burn time will be updated if sector s achieves the best among all remaining sectors.

— On the other hand, the node is charred. In this case, we need to update the burn time of this node to be the minimum burn time of its remaining sectors and make that sector the new primary sector. We then re-insert the node back to Q. This way of handling partially burned nodes implements the die-out rule elegantly. The loop terminates when there are no more nodes to burn. Note, the algorithm also stops when the most front node's estimated burn time is infinite. This means all the remaining nodes in Q belong to a *closed surface* of M. These nodes don't have any open sectors, and thus will be left unburned. Lastly ET is obtained by subtracting R from the computed BT.

A few interesting situations are worth noting here. A manifold node can be burned completely all at once. In contrast, a non-manifold node will be touched by fire, i.e. charred, multiple times due to the multiple sectors it has. The last time it is reached by a fire is when the node is left with a manifold disk sector and thus is burned. Each sector of a node monitors a branch of fire from some boundary point on ∂M . As long as the sector is alive, the estimated burn time recorded there is only going down due to receiving better update from neighboring, burned nodes. Thus the overall estimated burn time of the node gets smaller. This promotes the node to move to the front of Q, eventually gets the chance to be inspected at some iteration. However, if the node is found to be just charred and inserted back to Q, it can only re-emerge at a later time, thus its estimated burn time will be higher, or at least the same as last time.

Similar to Dijkstra's algorithm, the runtime complexity of the algorithm is determined by number of times of operations on Q and the complexity of each. Such operations include getMin, pushWithKey, and updateKey. They are called at most O(|A|) times, where |A|is number of arcs. Using a priority queue, each operation's cost is at most $O(\log|N|)$, where |N| is number of nodes. Therefore, our algorithm has the same complexity as Dijkstra's, i.e. $O(|A| * \log(|N|))$.

3.4.4 Algorithm: computing skeleton

As an application, our ET can be utilized to create a family of clean skeletons from the given medial axis M. A skeleton is loosely considered as a subset of the medial axis that correspond to prominent features of a surface S. It is temping to directly prune M using ET to obtain a skeleton. Unfortunately the result will not preserve the topology anymore. Though being free of local minima, ET in 3D contains saddle points where M will disconnect at any threshold above the function values. In addition, since ET prioritizes subset of M corresponding to planar parts of the S, thresholding can easily lose tubular features. To address this issue, we will compute medial curves MC to remedy topology and capture

Algorithm 1 Burn(G): Burning medial axis in 3D. G: graph, with boundary nodes ∂G .

1:	for $v \in G$ do	\triangleright Initialize Q with all nodes
2:	for all $s \in v.sectors$ do	
3:	$s.time \leftarrow \infty$	
4:	$s.burned \leftarrow False, s.primeArc \leftarrow \text{null}$	
5:	$\mathbf{if} \ v \in \partial G \ \mathbf{then}$	
6:	$v.time \leftarrow v.R$	
7:	else	
8:	$v.time \leftarrow \infty$	
9:	$v.burned \leftarrow False, v.primeSector \leftarrow network for the sector + network of the se$	ull
10:	while Q is not empty do	⊳ Main loop
11:	$v \leftarrow Q.getMin()$	
12:	if $v.primeSector \neq null$ then	⊳ Burn prime
13:	$v.primeSector.burned \leftarrow True$	
14:	for all exposed $s \in v.sectors$ do	\triangleright and sectors thus exposed
15:	$s.burned \leftarrow True$	
16:	$s.time \leftarrow v.time, \ s.primeArc \leftarrow null$	
17:	if all $v.sectors$ are burned then	$\triangleright v$ is burned - propagate fire
18:	$v.burned \leftarrow True$	
19:	for $s \in v.sectors$ and $a \in s.arcs$ do	\triangleright to neighbors via arcs of each sector
20:	$u \leftarrow \text{other end of } a$	
21:	$t \leftarrow$ sector of u that contains a	
22:	if not $u.burned$ and not $t.burned$ the	nen
23:	$h \leftarrow a.len + v.time$	
24:	if $h < t.time$ then	
25:	$t.time \leftarrow h, t.primeArc \leftarrow a$	
26:	if $h < u.time$ then	
27:	$u.time \leftarrow h, u.primeSector$	$or \leftarrow t$
28:	Q.updateKey(u, u.time)	
29:	else \triangleright v is	charred - update burn time & re-insert it
30:	$v.time \leftarrow$ the minimal time of unburned	d sectors of u
31:	$v.primeSector \leftarrow$ the sector with that t	time
32:	Q.pushWithKey(v,v.time)	

tubular parts. Again, our idea is inspired by the grassfire analogy. Recall the quench site of grassfire of a 2D shape is its 1D medial axis. Since we just perform a burning on the 2D medial axis M, we seek the quench site as our medial curves MC using dualization, which should be thin and preserve the topology of M. Following that we will also compute a significance measure similar to ET on M. The skeletons will then be the combination of M and MC parameterized by the amount of pruning guided by ET defined on each.

Our goal is to approximate the quench site of burning on M. When algorithm 1 terminates, we have a primaryArc at every sector s of a vertex v, that records v.s is burned by the fire from neighbor u. We can recover the burn tree of v if we collect all primaryArcs as we trace from v.s back to the boundary for every sector s of v. Doing this for every vertex gives us all the burn trees. They are a subset of the arcs of graph G, denoted as A', each arc being a "burned". Intuitively, if we let those arcs of fire continue to propagate, they should quench somewhere in the "middle" of the remaining part of $M \setminus A'$. This idea becomes concrete if we inspect a triangle face of M (Figure 3.6, right). If we only leave the edge- and face-arcs that are primary within a triangle, we can see they induce a subdivision of the triangle consisting of a set of empty regions. The regions don't overlap, and each is bounded by edge-arcs and primary edge- or face-arcs. According to our intuition, the quench site will be in each region, medial to the primaryArcs boundary since they represent strip of fires that eventually burn the region.

Computationally, we obtain a dual structure for each region by connecting the dual point on each non-primary arc to the dual point of the region. The collection of dual structure in all triangles will naturally form a network of polyline which serves as our medial curves MC. It can be shown that MC preserves the topology of the medial axis. Similar to a medial axis, MC can be noisy since by construction it is "medial" to its defining set M. To regularize MC, we follow the approach of [48] to compute a burn time on it, BT_{MC} . We then compute a similar ET measure on MC, ET_{MC} , which highlights parts of MC that correspond to tubular features on the surface S. Finally, we produce a family of skeletons controlled by two parameters θ_2, θ_1 . As seen in Figure 3.7, a skeleton combines sheets from M and curves from MC guided by the two ET measures using a topology preserving thinning process [47].



Figure 3.7: Skeletons combine prominent subset of medial axis and medial curves controlled by parameters θ_2 and θ_1 .

3.5 Results

In this section we compare ET computed using our algorithm with both local and global measures. We use C++ for all computation and OpenGL for visualization. An existing medial axis generation method, Scale Axis Transform (SAT) [50] is used to generate the triangulated medial axes as our input. We perform all experiments on a quad-core PC with 3.2GHz Xeon CPU and 12 GB RAM.

Comparison of significance measures We first compare ET with two local measures, i.e. object angle and circumradius. Here we show the comparison using the medial axes of the noisy Dolphin and Elephant shape in Figure 3.8. Our ET measure excels local measures in differentiating noise from features. While highlighting thin features (the tail of dolphin), object angle is also high near boundary noise (the large number of islands in the thresholded medial axis). While circumradius is much less sensitive to boundary noise, most of the thin features are lost as well. In contrast, ET highlights thin plate features while nicely suppresses boundary noise.

Compared with MGF, we found ET to be more stable under boundary perturbations that significantly alter geodesic distances. As shown in Figure 3.9, after ripples are added to one side of a box, the distribution and range of MGF values notably change while ET stays roughly the same. Moreover, ET is much more efficient to evaluate. For this example, MGF is computed in 4 days, while ET takes only 18 seconds. The similar stability against boundary noises can also be observed in the row corresponding to MGF in Figure 3.8.

Skeletons Due to ETs resistance to noise, the skeletons we computed under its guidance are also very robust to boundary perturbation. Shown in Figure 3.10, we found the skeletons barely changes after noises are added to the surface of the Dolphin and Elephant shape.



Figure 3.8: Comparison with local significance measures. Each measure is visualized by heat color (higher values use hotter color) and transparency (higher values are more opaque). Next to it is the portion of the medial axis whose significance values are higher than a threshold. Note that ET is sensitive to thin, plate-like features (see inserts) while being robust to perturbation.

3.6 Summary

We have presented a novel global significance measure over the medial axis of a 3D shape by extending Erosion Thickness previously defined in 2D. Compared to local measures, Erosion Thickness is able to better differentiate parts of the medial axis correspond to noises from



Figure 3.9: Comparison with MGF, a global significance measure. After expensive computation the result of MGF is sensitive to geodesic perturbations, while ET is efficient and robust.

those that correspond to major features on the shape. In contrast to other heuristics based "global" measures, our measure is mathematically well-defined. We are therefore able to analyze its key properties. We also provide an efficient algorithms for computing the measure and generating clean skeletons based on the measure.



Figure 3.10: Skeletons generated by our method are robust to boundary noise due to stability of ET.

Chapter 4

Conclusion and Future Works

In this dissertation, we have presented novel solutions that regularize and approximate medial axis. Because of many good properties, a considerable amount of effort has been made to utilize medial axis in domains such as visualization, medicine, and manufacture. However, there are fundamental barriers that prevent wide application. We refer to them as the *instability* and *approximation* challenges.

We first consider the approximation challenge with a provably good, efficient and robust medial axis proxy called *voxel core*. Existing approximation method generally falls into two categories: voxel-based and sampling-based. Voxel-based methods work on voxel shapes, and output a subset of the voxels to approximate the medial axis. They are simple to implement under the thinning framework which treats the shape as a dense volume of voxels and repeatedly peels off as many boundary voxels in the order guided by some field. The process guarantees that the result will have the same topology as the input. Nevertheless, these methods are dwarfed by their limited scalability since all voxels need to be densely stored in memory, limiting the resolution of the voxel shapes they can process. Especially, the issue is exacerbated in recent methods that use more advanced guidance fields to better capture the medial-like subset, as a larger footprint is incurred by each voxel. Alternative methods employ sparse voxel representation to optimize space usage, but are more difficult to implement. Moreover, few methods study the geometric closeness between their results and the true medial axis. On the other hand, sampling-based methods are more scalable as they only distribute samples on or around a shape's boundary. Particular efforts are focused on providing certain theoretical guarantees conditioned on sample density. As more property is guaranteed, ranging from proximity, homotopy equivalence, to topological thinness, these methods generally become more involved. While some methods can be as simple as picking a subset of Voronoi diagram by some local criteria, more difficult methods require multiple computation of Voronoi diagram and non-trivial geometric intersection which is inherently unstable. Consequently, sampling-based methods trade numerical robustness and ease of implementation for soundness. While also taking samples on boundary, our method is inspired by an insight in a voxel shape: no more samples than its boundary vertices are needed whose Voronoi diagram inside the shape, dubbed as voxel core, well approximates the medial axis. Theoretically, this allows us to show voxel core is topology preserving and geometrically close to the medial axis of voxel shapes. Additionally with some condition on voxel sizes, similar theoretical guarantees exist for input with smooth shapes. Furthermore, our simple algorithm compares favorably with existing methods. When processing voxel shapes, a key advantage in scalability gives us the capability of handling effective resolution of 2048^3 with reasonable cost, for which voxel-based methods quickly exhaust memory. For smooth mesh inputs, our algorithm is significantly more robust, free of topological errors which sampling-based methods routinely produce.

Then in Chapter 3 we approach the instability issue. Given a medial axis, existing research takes three approaches to simplify it: boundary smoothing, template fitting, and pruning. As the former two approaches either might alter the topology of the shape, or must need to fix the topology of the medial axis, most methods take the third approach. They first define a significance measure on the medial axis, and then perform a topology preserving contraction process to prune away as many as possible the non-important branches. While local measures are the most available due to being efficient and well-defined, they are inferior than global measures in terms of properly differentiating noises from main features on the medial axis. The most popular local measures are either too aggressive, discarding thin features as noises, or too conservative, preserving tips of noisy branches as important features. Global measures outperform local ones in this aspect since they evaluate the importance of a medial point in a much more global context of the shape. However they are extremely rare and costly. Our method also follows the pruning approach. The key difference is we successfully define Erosion Thickness (ET) in 3D, which is the first well-defined global measure in 3D. We rigorously prove it is finite, continuous, and free of local minima. The key to the definition is the burning of medial axis in 3D inspired by the grassfire analogy (Figure 1.3). We develop an efficient algorithm to approximately compute ET for a medial axis given as a triangular mesh, for which we also bound the approximation error. In addition, we utilize the residue of the burning of the medial axis to compute a family of expressive skeletons whose surface sheets capture planar regions and curves capture tubular parts on the shape.

4.1 Future works

While the two works we have presented compare favorably with state-of-the-art methods, there is certainly room for computational and theoretical improvements. We also see potential application of our tools in solving real-world problems.

4.1.1 Exact burning of medial axis in 3D

Underlying our Erosion Thickness is a discrete simulation of burning of the medial axis mesh M. Currently we approximate M with a graph G which finely samples edges of Mwith extra points to reduce errors introduced by the approximation. It is hopeful that one can totally eliminate the error by conducting an exact burn. Since burn time generalizes geodesic distance, a possible direction is to extend the exact geodesic distance algorithm [51, 81] which works on a piece-wise linear manifold (with or without boundary).

The geodesic distance measures the length of the geodesic path between source and target on a surface which becomes a straight line once the surface is flattened. Conceptually, in the piece-wise setting, it is equivalent to identify a strip of faces which contains the geodesic path. Then the geodesic distance can be exactly computed by flattening these faces and measuring the straight distance between the source and target. To do this computationally, Surazsky et al. [81] propagate "windows" using a priority queue in a Dijkstra-like fashion starting from the edges directly opposite to the source point. A window can clip against existing windows on an edge if it provides a better distance than the overlapped subset. At all times, each window represents a set of points that are either directly or indirectly visible to the source via a sequence of previous windows. The algorithm terminates when finishing processing all windows in the queue. Since the target is covered by some window, the geodesic distance is readily available, and the geodesic path can be recovered through a series of windows.

Apparently, some algorithmic challenges remain to generalize this algorithm due to the complex structure of medial axis.

- First, we are interested in propagating distance from multiple, edge-based sources rather than single, point-based source as burning starts from the entire boundary of M. At first glance, this change seems manageable, since we could instead convert boundary edge to a point-source determined by the intersection of the medial balls of the edge's ends. However, as we shall see later, allowing multiple such sources triggers a technical difficulty unique to our setting.
- Second, similar to how we deal with non-manifold nodes, windows should be propagated in a sheet-aware manner. This in turn entails a few challenging situations. At a certain time a non-manifold edge might already contain a few windows. As a new window is propagated to this edge from a boundary edge *a*, it needs to know which windows are on the same sheet and only subdivide them. Suppose after this update, a subset of this edge is considered exposed and a window is propagated from it, carrying further the information about source *a*. Here comes the technical issue hinted before. It's possible that part of this subset is exposed prematurely. In fact, it has a shorter distance to a different boundary edge *b*, and the exposure should be caused by a window from *b*. However, the window from *b* comes after the exposure and fails to update the subset correctly. Hence, we need to have some roll-back mechanism to undo the propagation back to a certain time point. Note this is less of an issue for manifold mesh, since such errors are temporary. Without the non-manifold blockage (recall the die-out burning rule), correct updates can always be made when all windows in the queue are processed.

4.1.2 Hierarchy of burn times

Grassfire analogy inspires Liu et al. [48] to design burning of medial axes of shapes in 2D, which we generalize to 3D. In retrospect, we realize the notion of burning can be further exploited. In fact, burning could potentially formulate a general framework that defines medial structures in lower dimensions. To hint the direction we are pursuing, let's summarize what we have so far in our already familiar 3D setting as depicted in Figure 4.1. Consider a shape $O \subseteq \mathbb{R}^3$. The grassfire essentially defines a Euclidean distance function DF everywhere in O, whose critical points capture thin medial axis, and the subset restricted to M is referred to as the radius function R. To help regularize M, we next define the burn time BT_M on M, from which R is subtracted to define the measure ET_M . We also approximate the quench site of BT_M to obtain medial curves MC that are even thinner than M, and denote the subset of BT restricted to MC by BT_{MC} . Willing to regularize MC, we obtain the burn time BT_MC on MC as defined in [48], and subtract BT_M to define the measure ET_C . Although we don't compute it, we conjecture the quench site of MC is of even lower dimension, e.g. medial points.

The above can be summarized in a unified burning which we envision as follows. Given a n-dimensional open set $O^n \subseteq \mathbb{R}^n$ equipped with a function F^n over O^n , burning of the set starting from its boundary ∂O gives a burn time function BT^n that's defined over O. The residue of the burning, or burn axis is a (n-2)-dimensional structure M^n that's thinner than O. The Erosion Thickness ET^n can be defined by subtracting F^n from BT^n which highlights planar features, i.e. parts of O that are more compressed along 1-of-n dimension up to rigid transformation. Recursively applying to O this burning process $d = 1, \ldots, n$ times gives a measure ET^{n-d+1} whose residue is a (n-d)-dimensional medial form. An overarching advantage of such a framework is it provides a principled way to define medial forms in any dimension, including medial axis, medial curves, and medial points in 3D as special cases. We will be able to rigorously analyze and verify the properties of medial forms and their derivative descriptors. The major challenge we face is to show such a framework exists. For instance, we need to show in 3D, medial axis, curves and points can be defined in



Figure 4.1: Burning processes and products in 3D unified in the same framework. From left to right, each burning takes a shape and a function defined on it (0 everywhere in the initial Hand shape), and produces the quench site - a thin medial structure, and the burn time on it. Two consecutive burn times define a significance measure (ET) that highlights features expanded along all but one dimension of the embedding space. Hence, planer features stand out on MA embedded in 3D space, while tubular features on MC embedded in 2D MA.

the unified way suggested by the framework, which is rather challenging since straightforward extensions of medial axis to lower dimension do not necessarily define medial curves or points that are thinner and have the correct topology.

4.1.3 Recovery of Tree Topology

Finally, we argue the two methods in this dissertation are of valuable practical implications for solving a real-world problem. In plant science, it is well known that the root system architecture (RSA) of plants serve many functions key to the plants' well-growth. Generally, these functions are analyzed by extracting shape traits of RSAs, which in turn help identify the underlying useful genes. Previously, high quality roots data were rare and only primitive shape traits were used [57, 25], which significantly limited the understanding of the RSA. As 3D root imaging techniques mature [89, 93], more roots represented as 3D voxel shapes of high resolution become available. The hope is with these data, more advanced shape traits that better discriminate genes can be extracted. However, the necessary computational tools for this task are still lacking. The challenges are two-folded.

- First, the imaging resolution is still not sufficient to capture all the details of a RSA correctly. As a consequence, topological errors prevail in the captured shape. These topological errors can severely influence the quality of the traits to be extracted.
- Second, extraction of meaningful shape traits requires understanding a wide spectrum of variation in both structure and geometry of roots. However such a shape analysis solution is still elusive.

As part of a larger collaboration, we are fortunate to have access to a set of high resolution 3D images of roots of corns at different growth stages. In Figure 4.2, the surface of one of the dataset is displayed. As we observe, there are numerous topological errors including cycles, touchings, and disconnected components. It's not hard to imagine how some structural dependent traits, e.g. branching factor, can be erroneous if these topological errors are left unattended. For this reason, as a future work, we plan to repair the topology of the roots based on skeletons. We think this is hopeful since the skeleton of roots offers a compact representation that encodes both the structure and geometry. Roughly, we aim to pursue an idea of the following. First we compute the voxel core of the roots' voxel shape, which



Figure 4.2: A visualization of a curve skeleton inside the roots 3D image, which contains many topological errors (a). Small cycle (b) and merging (c) of branches can be identified with the help of skeleton and the associated shape measures.

is pruned by ET to obtain the skeleton. Guided by measures like the local thickness and width made available by ET, we then identify the regions of the skeleton that are likely to contain topological errors (Figure 4.2). A set of repairs will be proposed local to each region independently. Finally, a global choice will be made as to what operation to perform for every region that combined together will lead to a skeleton that optimizes a set of goals. Knowing that the output must be of tree topology, a likely solution for the last step is to formulate an energy that optimizes for the geometry of the roots, subject to hard topological constraints. We further note that skeletons are also good candidate for shape traits extraction. Featuring a clear graph structure, the skeleton of roots yields a natural decomposition of the roots into segments. Those segments can be helpful in counting number of branches, and selectively merged to identify roots whirl structure. This work will lead to an automatic pipeline that takes 3D image of roots and outputs various shape traits. We believe such a pipeline could significantly reduce the time cost for biologists to perform analysis, and thus yield positive impact on the landscape of relevant research in plant engineering.

Appendix A

Proof for Voxel Core

A.1 Proximity of voxel core to medial axis

We will show the property of proximity of voxel core (Theorem 2.4.4). Before that, it is useful to the proof to have the following fact:

Lemma A.1.1. Consider three points $x, p, q \in \mathbb{R}^n$ s.t. d(x,p) < d(x,q). Given a unit direction \vec{n} which forms acute angle with \vec{pq} , i.e. $(q-p) \cdot \vec{n}$, a point y on the ray emanating from x is $y = x + s * \vec{n}$ where s > 0. When

$$s = \frac{d(x,q)^2 - d(x,p)^2}{2(q-p) \cdot \vec{n}}$$

y is equidistant to p and q, i.e. d(y,p) = d(y,q). See Figure A.1 for notations.

We can show the lemma is true by expanding $d(y, p)^2$ and substituting in the nominator of s to arrive at $d(y, q)^2$.

Next we proceed to the theorem which states the distance between a given voxel core point and its closest medial point is bounded. We need the following notations illustrated in Figure



Figure A.1: Notations used for Lemma A.1.1.

A.2. Let *B* and *P* be the entire boundary and boundary vertices of voxel shape *O*. Recall $P \subset B$. For any point on the voxel core $x \in C$, denote $\Gamma_P(x) \subseteq P$ the set of closest points among *P*, and $\Gamma_B(x) \subseteq B$ the set of closest points on *B*. We can find a point $p \in \Gamma_B(x)$ contained in a boundary element $e \subseteq B$. Name the vertex of *e* closest to *p* by *v*. *v* is also the closest to *p* among all *P*. There also exists a boundary vertex among the closest boundary vertices set $\Gamma_P(x)$ that's not any vertex of *e*. Note $d(x, v) \ge d(x, q) \ge d(x, p)$. We re-state Theorem 2.4.4 here:

Theorem A.1.1. (Theorem 2.4.4) For any point $x \in C$, $d(x, M) \leq \frac{1}{4}h$

Proof. The idea is to move x, which is already close to one boundary point p, along some unit direction until it reaches a point y such that y is equidistant to p, and q - another point on boundary B. Lemma A.1.1 will be used to bound the distance between x and y. Note



Figure A.2: Notations used in the proof of Theorem 2.4.4. Left: e is a voxel face. Right: e is a voxel edge shared by only two voxel faces. Viewing direction is parallel to the e.

that since x is guaranteed inside O, we can then reason that y or some point between x, y must be on the medial axis.

Based on the type of element e, we show the theorem for two cases as shown in Figure A.2.

case 1: e is a boundary face. In this case, we choose \vec{n} to be the unit direction of vector x-p, and we will find y along \vec{n} . To make use of Lemma A.1.1, we need to show $(q-p) \cdot \vec{n} > 0$, that is both q and x are on same side of the supporting plane of e. Starting from the right angle $\angle vpx$ and $d(x,v) \ge d(x,q)$, we have:

$$d(v,p)^{2} = d(x,v)^{2} - d(x,p)^{2} \ge d(x,q)^{2} - d(x,p)^{2}$$

Since e contains p, v, a vertex of e, is the closest boundary vertex to p, we have:

$$d(q, p) > d(v, p)$$

It follows from two above relations

$$d(q,p)^{2} + d(x,p)^{2} > d(x,q)^{2}$$

which proves $(q, p) \cdot (x, p) > 0$.

Now applying Lemma A.1.1, we obtain along the ray from x with direction n a point y equidistant to p, q and forms distance to x:

$$d(x,y) = \frac{d(x,q)^2 - d(x,p)^2}{2(q-p) \cdot \vec{n}}$$

This distance can be bounded. First we should note the nominator is shown to be no larger than $d(v, p)^2$ earlier, which in turn is no bigger than $\frac{h^2}{2}$, the squared half diagonal of the square face e. Denominator can be lower-bounded by noting q should be at least h away from the supporting plane of e, hence is no smaller than 2h. Distance between x, y is shown to be bounded:

$$d(x,y) \le \frac{h}{4}$$

Lastly, we assert the segment x, y must contain a medial axis point. Otherwise, y is a medial point. This is because $x \in C$ is inside shape O, and x, y doesn't hit M, so all points on x, y are in O. Since y is a point inside O that has two closest points on B, fitting the medial axis definition, $y \in M$. This leads to the bound of distance to M:

$$d(x,M) \le d(x,y) \le \frac{h}{4}$$

case 2: e is a boundary edge. In this case, e can only be shared by only two voxel faces, denoted as f_1, f_2 , whose normals are $\vec{n_1}, \vec{n_2}$ pointing toward x. This time we need to show q, x are on the same side of the supporting plane of at least one face. Again, let \vec{n} be the unit direction of x - p, which can be written as combination of $\vec{n_1}, \vec{n_2}$:

$$\vec{n} = a_1 \vec{n_1} + a_2 \vec{n_2}$$

Note a_1, a_2 are all positive, since otherwise is covered by case 1. Following the same analysis of case 1, we can show $(q - p) \cdot (x - p) > 0$, which implies $(q - p) \cdot \vec{n}$. Replacing \vec{n} with its linear combination form, we have:

$$a_1(q-p) \cdot \vec{n_1} + a_2(x-p) \cdot \vec{n_2} > 0$$

For this to be true, at least one term is positive. Since a_1, a_2 are positive, without loss of generality, we have $(q - p) \cdot \vec{n_1} > 0$. Hence q, x are on one side of the plane supporting face f_1 . Applying Lemma A.1.1, we have a point y equidistant to q, p along the ray $x + s * \vec{n_1}$, such that:

$$d(x,y) = \frac{d(x,q)^2 - d(x-p)^2}{2(q-p) \cdot \vec{n_1}}$$

Similarly, we can bound it. The nominator can be again upper-bounded by $d(v, p)^2 = \frac{h^2}{4}$, i.e. the squared half length of voxel edge e. The denominator is lower-bounded by 2h. Together we have the bound

$$d(x,y) \le \frac{h}{8}$$

Following the similar argument in end of case 1, we can show from x to y the segment either contains a medial axis point before y, or y is a medial axis point. Hence the bound of the distance to M:

$$d(x,M) \le d(x,y) \le \frac{h}{8}$$

A.2 Proximity of voxelization to smooth shape

We show Theorem 2.5.1 that states the smooth shape O and its voxelization, i.e. a voxel shape O_h are close to each other when the voxel size h is sufficiently smaller than r, the reach of O. Let the boundary of O be B, boundary vertices of O_h be P_h , we re-state the theorem.

Theorem A.2.1. (Theorem 2.5.1)For any $h \le \frac{2\sqrt{3}}{3}r$, (1) $d_H(P_h, B) < \frac{\sqrt{2} + \sqrt{3}}{2}h$ (2) $d_H(O_h, O) < \frac{\sqrt{3}}{2}h$

Proof. Lachaud and Thibert [43] show that the Hausdorff distance between the smooth boundary and its voxelized boundary in dimension d is bounded above by $\sqrt{d}/2$, when the voxel size is smaller than $2r/\sqrt{d}$. In our case, the two surfaces are B_h and B at dimension d = 3. We then have the following specialized theorem, upon which we base our proof:

(Theorem L.T.) For any $h \leq \frac{2\sqrt{3}}{3}r$

$$d_H(B_h, B) \le \frac{\sqrt{3}}{2}h$$

proof of (1) We first show the one direction of the first Hausdorff distance. That is, given $x \in P_h$, there exists a point $y \in B$ such that $d(x,y) < \frac{\sqrt{2}+\sqrt{3}}{2}h$. This is immediately



Figure A.3: Notations used for the proof of Theorem 2.5.1. Left: bounding $d_H(P_h, B)$ for direction $B \to P_h$. Middle: bounding $d_H(O_h, O)$ for direction $O_h \to O$. Right: the one directional distance $B \to B_h$ implies a band (orange) along B that contains B_h .

true, since P_h is a subset of B_h and due to Theorem L.T., we can find a point $y \in B$ s.t. $d(x,y) \leq \frac{\sqrt{3}}{2}h < \frac{\sqrt{2}+\sqrt{3}}{2}h.$

Then we show the bound for the direction $d(x, P_h)$ for a point $x \in B$. First there exists a point $y \in B_h$ and $d(x, y) \leq h\sqrt{3}/2$ (Theorem L.T.). Since y is on the boundary of voxel shape, it must be contained by a boundary element e (face or edge). By Lemma 2.4.1, the closest boundary vertex for y in P_h , p, will be one of the vertex of e, which is no larger than $\sqrt{2}/2h$, half the length of the diagonal of a voxel face. By triangle inequality, $d(x,p) \leq d(x,y) + d(y,p) \leq h(\sqrt{2} + \sqrt{3})/2.$

proof of (2) To bound the distance for the direction $O_h \to O$, we only need focus on a point x that's in O_h but not in O. There must be a voxel containing x whose center c is in O. Since x is outside of O, immediately there exists a point $y \in B$ between x and c. We have $d(x, O) \leq d(x, y) \leq h\sqrt{3}/2$, half diagonal of the voxel.

Next, we bound the distance for the direction $O \to O_h$ by considering a point $x \in O \setminus O_h$. The Hausdorff distance bound in Theorem L.T. means for each point on B there centers a ball with radius $h\sqrt{3}/2$ that intersects B_h . The union a such balls will form a band around B, each side having thickness of $h\sqrt{3}/2$ (Figure A.3, right). The band completely contains B_h in its interior. Since x is inside O but outside of O_h , it is in the space sandwiched between the two boundaries B and B_h , as illustrated in the example in the Figure. Hence, a ball of radius $h\sqrt{3}/2$ centered at x must intersect with B_h , proving the bound.

A.3 Homotopy equivalence of voxelization to smooth shape

We show the voxel shape O_h is homotopy equivalent to the smooth shape O under small enough voxel size. We will build the proof based on a notable result by Stelldinger et al. that shows a *topology preserving* reconstruction from the voxel shape is homeomorphic to the smooth shape. Roughly two sets $A, B \subseteq \mathbb{R}^n$ are homeomorphic to each other if there is a bijection mapping between them, which implies homotopy equivalence between the two. Therefore, our plan is to show we can obtain a surface satisfying the topology preserving criteria. Then homotopy equivalence will trivially follow.

Consider a group of $2 \times 2 \times 2$ adjacent voxels, whose centers form vertices of a cube. We call it a **dual cube** since it is dual to the vertex shared by these 8 voxels as depicted in Figure A.5. Now every vertex of a dual cube corresponds to the center of a voxel which is either in/out of O. Determined by the insideness of each vertex, a dual cube can be in 14 canonical configurations up to rotational, reflectional and complementary symmetry. Note

even with this number of possibilities, it is still difficult to guarantee under what condition the voxel shape O_h encodes the topology of O, as some configurations are non-manifold or ambiguous. Remarkably, Stelldinger et al. are able to show, only 8 of the configurations (see Figure A.4) are possible when voxels are small enough:

Theorem A.3.1. (Results from [77]) For any voxel size $h \leq \frac{\sqrt{3}}{3}r$, where r is the reach of smooth shape O, a dual cube can only be in 1 of the 8 configurations (Figure A.4). Particularly, configuration 8 always appear in pair, with the other dual cube in the complementary state, i.e. the non-shared vertex has flipped insideness.

Provided the voxel size condition is satisfied, Definition 15 of [77] shows that the shape reconstructed from the voxel shape O_h is **topology preserving** if its boundary in each dual cube is topologically equivalent to a disk as depicted in Figure A.4. Theorem 16 of the paper further states the reconstructed shape is also homeomorphic to the original smooth shape O. Next, we will build O' from O_h whose boundary $\partial O'$ is a topology preserving reconstruction to prove homotopy equivalence.

We build O' by reconstructing its boundary locally for each dual cube. Since dual cube 1 is completely in or out of O, it induces no surface. For dual cube 2-7, we simply take the voxel boundary surface within the cube. It's easy to verify the result has a disk topology. However this strategy doesn't work directly for dual cube (8) since it will introduce a non-manifold edge. Hence, in cube 8, after we obtain the voxel boundary, we push the center point onto the bottom face and deform the nearby boundary surface continuously. This also applies to the other cube of the pair. Due to complementarity, the local reconstruction in the cube-pair is equivalent to a disk (Figure A.4). Since all the operation is continuous, O' is homotopic equivalent to O_h . Also, O' is topology preserving, to which the original shape O is homotopy



Figure A.4: Top: 8 possible configurations for a dual cube when voxel size is sufficiently small (Theorem A.3.1). Bottom: A topology preserving reconstruction in each configuration is equivalent to a disk. Dual cube (8) always appears with a cube of the complementary states. That is, one in the pair has 2 inside and 6 outside voxel centers, while the other 2 outside and 6 inside voxel centers. They share the face having two inside voxel centers. Image courtesy of [77].


Figure A.5: (a): A dual cube whose 8 corners are voxel centers. (b): The voxel occupancy of the dual cube (8) in Figure A.4. Red/White dots: in/outside voxel centers. (c): The boundary of our topology preserving shape in the cube (and pairing-cube below) is formed by pressing the center of the voxel boundary toward the face shared by the cube-pair, resulting in a manifold surface whose boundary curve is colored by black.

equivalent (from Definition 15 and Theorem 16 of [77]). Hence, O is homotopy equivalent to O_h .

Appendix B

Proof for Erosion Thickness

We introduce the notations we will use in the following proof. For a shape S, the medial axis is \mathcal{M} equipped with radius function R, whose manifold, singular, and boundary points are $M_2, M_s, \partial M$. In addition, maximally closed subset of M is denoted by M_C . A point x in the closure of medial axis $M = \overline{\mathcal{M}}$ has at least one nearest point on ∂S whose distance to xis R(x).

B.1 Properties of ET in 3D

B.1.1 Finiteness

Theorem B.1.1. For a point in medial axis $x \in M$, $ET(x) = \infty$ if and only if $x \in M_C$, where M_C is the maximal closed subcomplex of M.

Proof. If $x \in M_C$, then intuitively any exposing tree for x (with root at x) will have infinite complexity because it has to keep spawning its paths looking for boundary points, which there are none on M_C . Hence the burn time BT(x) is infinite and also $ET(x) = \infty$.

Otherwise, $x \in M \setminus M_C$. We show that we can always find a finite exposing tree rooted at x. We first note $M \setminus M_C$ can be decomposed into manifold components $\mathcal{C} = \{C_1, \ldots, C_k\}$ by breaking it along the singular curves. We call a component C exposed by a set of components \mathcal{C}' if after removing every component in \mathcal{C}' from $M \setminus M_C$, C has some boundary. We can then order all components in \mathcal{C} such that components having some part from ∂M thus exposed by \emptyset come first, followed by components that can be exposed by those before it. Now we are ready to inductively build exposing tree for x. In base case, consider $x \in C \setminus M_S$ for all C that can be exposed by \emptyset , then there exists a finite exposing tree, which is a path, rooted at x with leaf on $\partial C \subset \partial M$. Then assume every point in $C_1 \cup \ldots \cup C_{i-1}$ (including all \emptyset -exposable components) has a finite exposing tree. Consider a point $x \in C_i$. If $x \in \partial C_i$, then an exposing set of sectors of x lie on one or more components in C_1, \ldots, C_{i-1} . For each such sector s, x can be connected by a path to an interior point y on s which in turn can connect to the already built exposing tree of y (by induction). Else, x is a interior point in C_i . Then we simply extend a path from x to a point y on the exposed boundary ∂C_i , and glue the exposing tree of y (constructed as previous step) with the path. Either way, a finite exposing tree of x is built.

B.1.2 Continuity

We first present a lemma that bounds the variation of radius function between two points x, y. Let $d_A(x, y)$ be the infimum of the length of paths between $x, y \in A$ restricted to a set A.

Lemma B.1.1. Given two distinct points $x, y \in M$, $R(x) \leq R(y) + d_M(x, y)$. The inequality becomes strict when $x, y \in \mathcal{M}$.

Proof. The sketch of the proof proceeds by contradiction. Intuitively, if radius function varies faster than the distance on the medial axis between two medial points, then the medial ball of y will be enclosed in the medial ball of x, thus y cannot be on the medial axis, leading to contradiction.

Theorem B.1.2. 1. ET is continuous and 2-lipschitz over $M_2 \setminus M_C$.

2. Consider a singular point $x \in M_s \setminus (\partial M \cup M_C)$. ET over $M_2 \cup x$ is

(a) Upper-semicontinuous at x.

(b) Continuous and 2-Lipschitz at x within some 2-dimensional disk $D \subseteq M$ containing x in its interior.

Proof. Let $A = M_2 \setminus M_C$, we prove the property by order.

1: We want to bound the derivative of function ET w.r.t. $x \in A$ over the domain A. This can be broken down to first introduce a change in the point x, then study how much EThas changed. Consider any two points $x, y \in A$. The change between them is the shortest distance on A, $d_A(x, y)$, which gives the change in the domain. The change of ET is:

$$|ET(x) - ET(y)| = |BT(x) - BT(y) - (R(x) - R(y))|$$
 definition of ET
$$< |BT(x) - BT(y)| + |R(x) - R(y)|$$
 absolute value property
$$< 2d_A(x, y)$$
 bound of variation of BT and lemma B.1.1

where the bound of BT can be shown to be $|BT(x) - BT(y)| \le d_A(x, y)$. Hence, ET is 2-Lipschitz, which implies continuity. 2(a): Similarly, for any $x \in M_s \setminus (\partial M \cup M_C)$ and $y \in A$ we can show the one side Lipschitz condition $ET(y) - 2d_A(x, y) \leq ET(x)$, meaning ET is upper-semicontinuous at x.

2(b): To show the continuity and Lipschitz condition within the disk D around x, we first find D. Intuitively D always exists as the last full disk after removing the minimal number of sectors around x that contain the root edges of the exposing tree that realizes BT(x). Then for $y \in D \cap M_2$, the analysis conducted for property 1 can be applied to show 2-Lipschitz and continuity on D.

B.1.3 Local minima

Theorem B.1.3. For any $x \in \mathcal{M} \setminus (\partial M \cup M_C)$ and any 2-dimensional disk $D \subseteq \mathcal{M}$ that contains x in its interior, there is some $y \in D$ such that ET(x) > ET(y).

Proof. Note there is always some point $y \in \partial D$ which is burned before x. Otherwise, no exposing tree exists for x which contradicts that $x \notin M_C$. Then we have $BT(x) - BT(y) \ge d_D(x, y)$, since $BT(y) + d_D(x, y)$ only captures the length of a branch going through y of the burn tree rooted at x (recall the definition of exposing tree). We then have:

$$ET(x) - ET(y) = BT(x) - BT(y) - (R(x) - R(y))$$
 definition of ET

$$\geq d_D(x, y) - (R(x) - R(y))$$
 see above analysis

$$\geq d_M(x, y) - (R(x) - R(y))$$
 $D \subseteq M$

$$\geq d_M(x, y) - |R(x) - R(y)|$$
 absolute value property

$$\geq 0$$
 lemma B.1.1

B.2 Error bound of graph-restricted ET

Theorem B.2.1. Let |M| be the number of triangles in the input triangle mesh M, g be the maximal gradient magnitude of R on any triangle edge on ∂M , and ω be the maximal distance between adjacent nodes of G on a triangle edge. For any node v in G, its ET restricted on G, ET_G is bounded:

$$ET(v) \le ET_G(v) \le ET(v) + (2|M| + g)\omega$$

We proceed with two parts to show the bound. We first show in Lemma B.2.1 that for some vertex v its exposing tree on M can be converted to an exposing tree on G, and the graph-restricted exposing tree is short in some sense. Then we show the complexity of this tree is also bounded in Lemma B.2.2. Together the two lemmas will lead to a proof of the theorem.



Figure B.1: G^+ adds jump-arcs (green) to G on each triangle edge, forming the complete graph within a triangle.

Lemma B.2.1. Given an exposing tree Γ on M rooted at v (located on triangle vertex or a triangle edge), there exists an exposing tree Γ' restricted on G whose length is bounded

$$L(\Gamma') \le L(\Gamma) + (2k+g)\omega$$

where k is the maximal number of edges of any root-to-leaf path in Γ .

Proof. The idea is the following. We first augment the graph G with some jump-arcs to obtain G'. Then for a node v in G, any exposing tree Γ for v in M, can be converted to an exposing tree Γ' restricted to G via an intermediate graph G^+ . The graph is constructed as depicted in Figure B.2.

With G^+ constructed, Γ is first converted to an exposing tree Γ^+ in G^+ , which is then converted to Γ' . The first conversion is done by "snapping" each chord of Γ lying on a triangle to an arc in G^+ in one of the three ways (see Figure B.2. Note approximation error will be introduced here, but the second conversion is lossless.



Figure B.2: Each chord c (dashed) is a segment within a triangle of a path on M. It can be "snapped" to an arc a (solid) in one of the three ways.

The rest of the work is to analyze the error introduced in the first conversion to achieve the bound. We can first bound for each chord $c \subseteq \Gamma$ the length of the arc $a \subseteq \Gamma^+$ it snaps to using the configurations in Figure B.2, giving:

$$|a| \le |c| + 2\omega$$

Since there are at most k chords along any root-leaf path γ in Γ , whose corresponding path in Γ^+ is γ^+ , we sum up the inequality for them, arriving at:

$$|\gamma^+| \le |\gamma| + 2k\omega$$

We also need to account for the difference between the radius function R at the leaf of γ and γ^+ , denoted as u and u^+ :

$$R(u^+) \le R(u) + g\omega$$

Simply adding the two above inequalities gives:

For any γ (correspondingly γ^+), $|\gamma^+| + R(u^+) \le |\gamma| + R(u) + (2k+g)\omega$



Figure B.3: A path γ of an exposing tree meandering in a triangle (dashed) can be straightened into one chord (solid) that's simpler and shorter.

Since any root-leaf path satisfies the above condition, it also apply to the exposing tree length. $\hfill \Box$

Lemma B.2.2. Given an exposing tree Γ on M rooted at v (on triangle vertex or edge), there exists another exposing tree Γ' such that

1. $L(\Gamma') \leq L(\Gamma)$

2. Each root-to-leaf path of Γ' has at most one chord on any single triangle (or one of its edge)

Proof. Intuitively, we can shorten the given tree Γ by straightening its intersection with a triangle in one of the two ways as depicted in Figure B.2. The resulting tree Γ' will be no longer than Γ , and only has at most one chord in any triangle.

Combining Lemma B.2.1, B.2.2, we have that for any exposing tree Γ on M for a node v on triangle vertex or edge, there exists a G-restricted exposing tree Γ' whose length is bounded:

$$L(\Gamma') \le L(\Gamma) + (2|M| + g)\omega$$

Since this holds for any exposing tree, it also holds for burn tree whose length realizes BT:

$$BT(v) \le BT_G(v) \le BT(v) + (2|M| + g)\omega$$

Note the left inequality holds because G is only a subset of M. Subtracting R(v) on all sides leads us to the condition in Theorem B.2.1:

$$ET(v) \le ET_G(v) \le ET(v) + (2|M| + g)\omega$$

References

- Oswin Aichholzer, Wolfgang Aigner, Franz Aurenhammer, and Bert Jüttler. Exact Medial Axis Computation for Triangulated Solids with Respect to Piecewise Linear Metrics, pages 1–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [2] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In Proceedings of the sixth ACM symposium on Solid modeling and applications, pages 249–266. ACM, 2001.
- [3] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2-3):127–153, jul 2001.
- [4] Nina Amenta and Ravi Krishna Kolluri. The medial axis of a union of balls. Computational Geometry, 20(1-2):25–37, 2001.
- [5] Carlo Arcelli and Gabriella Sanniti di Baja. Euclidean skeleton via centre-of-maximaldisc extraction. Image and Vision Computing, 11(3):163–173, 1993.
- [6] Carlo Arcelli, Gabriella Sanniti di Baja, and Luca Serino. Distance-driven skeletonization in voxel images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):709–720, 2011.
- [7] Dominique Attali and Jean-Daniel Boissonnat. A linear bound on the complexity of the delaunay triangulation of points on polyhedral surfaces. *Discrete & Computational Geometry*, 31(3):369–384, 2004.
- [8] Dominique Attali, Gabriella di Baja, and Edouard Thiel. Pruning discrete and semicontinuous skeletons. In *Image Analysis and Processing*, pages 488–493. Springer, 1995.
- [9] Dominique Attali and Annick Montanvert. Modeling noise for a better simplification of skeletons. In *Image Processing*, 1996. Proceedings., International Conference on, volume 3, pages 13–16. IEEE, 1996.
- [10] Matthew L Baker, Tao Ju, and Wah Chiu. Identification of secondary structure elements in intermediate-resolution density maps. *Structure*, 15(1):7–19, 2007.
- [11] Gilles Bertrand and Grégoire Malandain. A new characterization of three-dimensional simple points. Pattern Recognition Letters, 15(2):169–175, 1994.

- [12] Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- [13] Harry Blum. Biological shape and visual science (part i). Journal of theoretical Biology, 38(2):205–287, 1973.
- [14] Jonathan W Brandt and V Ralph Algazi. Continuous skeleton computation by voronoi diagram. CVGIP: Image understanding, 55(3):329–338, 1992.
- [15] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, pages 1704–1711. IEEE, 2010.
- [16] Frederic Cazals, Aditya Parameswaran, and Sylvain Pion. Robust construction of the three-dimensional flow complex. In *Proceedings of the twenty-fourth annual symposium* on Computational geometry, pages 182–191. ACM, 2008.
- [17] John Chaussard, Michel Couprie, and Hugues Talbot. A discrete λ -medial axis. In International Conference on Discrete Geometry for Computer Imagery, pages 421–433. Springer, 2009.
- [18] Frédéric Chazal and André Lieutier. Stability and homotopy of a subset of the medial axis. In Proceedings of the ninth ACM symposium on Solid modeling and applications, pages 243–248. Eurographics Association, 2004.
- [19] Frédéric Chazal and André Lieutier. The λ -medial axis. *Graphical Models*, 67(4):304–331, 2005.
- [20] Frédéric Chazal and André Lieutier. Smooth manifold reconstruction from noisy and non-uniform approximation with guarantees. *Computational Geometry*, 40(2):156–170, 2008.
- [21] Tim Culver, John Keyser, and Dinesh Manocha. Exact computation of the medial axis of a polyhedron. *Computer Aided Geometric Design*, 21(1):65–98, jan 2004.
- [22] Tamal K Dey and Jian Sun. Defining and computing curve-skeletons with medial geodesic function. In Symposium on geometry processing, volume 6, pages 143–152, 2006.
- [23] Tamal K Dey and Wulue Zhao. Approximate medial axis as a voronoi subcomplex. Computer-Aided Design, 36(2):195–202, 2004.
- [24] Andreas R Dill, Martin D Levine, and Peter B Noble. Multiple resolution skeletons. IEEE Transactions on Pattern Analysis and Machine Intelligence, (4):495–504, 1987.

- [25] david M. eissenstat. On the relationship between specific root length and the rate of root proliferation: a field study using citrus rootstocks. New Phytologist, 118(1):63–68, may 1991.
- [26] Rolando Estrada, Carlo Tomasi, Scott C Schmidler, and Sina Farsiu. Tree topology estimation. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1688– 1701, 2015.
- [27] Michal Etzion and Ari Rappoport. Computing voronoi skeletons of a 3-d polyhedron by space subdivision. *Computational Geometry*, 21(3):87–120, 2002.
- [28] Mark Foskey, Ming C Lin, and Dinesh Manocha. Efficient computation of a simplified medial axis. In Proceedings of the eighth ACM symposium on Solid modeling and applications, pages 96–107. ACM, 2003.
- [29] Yaorong Ge and J Michael Fitzpatrick. On the generation of skeletons from discrete euclidean distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelli*gence, 18(11):1055–1066, 1996.
- [30] Peter Giblin and Benjamin B Kimia. A formal classification of 3d medial axis points and their local geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):238–251, 2004.
- [31] Joachim Giesen, Balint Miklos, Mark Pauly, and Camille Wormser. The scale axis transform. In Proceedings of the twenty-fifth annual symposium on Computational geometry, pages 106–115. ACM, 2009.
- [32] Joachim Giesen, Edgar A Ramos, and Bardia Sadri. Medial axis approximation and unstable flow complex. In Proceedings of the twenty-second annual symposium on Computational geometry, pages 327–336. ACM, 2006.
- [33] Polina Golland, W Eric, and L Grimson. Fixed topology skeletons. In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, volume 1, pages 10– 17. IEEE, 2000.
- [34] Paul Heider, Alain Pierre-Pierre, Ruosi Li, and Cindy Grimm. Local shape descriptors, a survey and evaluation. In *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*, 3DOR '11, pages 49–56, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association.
- [35] Seng-Beng Ho and Charles R Dyer. Shape smoothing using medial axis properties. *IEEE transactions on pattern analysis and machine intelligence*, (4):512–520, 1986.
- [36] Philip M Hubbard. Approximating polyhedra with spheres for time-critical collision detection. ACM Transactions on Graphics (TOG), 15(3):179–210, 1996.

- [37] AC Jalba, A Sobiecki, and AC Telea. An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE transactions on pattern*, 2016.
- [38] Tao Ju. Robust repair of polygonal models. In ACM Transactions on Graphics (TOG), volume 23, pages 888–895. ACM, 2004.
- [39] Reinhard Klette and Azriel Rosenfeld. Digital geometry: Geometric methods for digital picture analysis. Elsevier, 2004.
- [40] Jan Knopp, Mukta Prasad, Geert Willems, Radu Timofte, and Luc Van Gool. Hough transform and 3d surf for robust three dimensional classification. In *European Confer*ence on Computer Vision, pages 589–602. Springer, 2010.
- [41] T.Y Kong and A Rosenfeld. Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing, 48(3):357–393, dec 1989.
- [42] Jacek Łukasz Kustra. Medial Descriptors for 3D Shape Segmentation, Reconstruction and Analysis. University of Groningen, 2015.
- [43] Jacques-Olivier Lachaud and Boris Thibert. Properties of gauss digitized shapes and digital surface integration. Journal of Mathematical Imaging and Vision, 54(2):162–180, 2016.
- [44] Mark Lanthier, Anil Maheshwari, and Jörg-Rüdiger Sack. Approximating weighted shortest paths on polyhedral surfaces. In *Proceedings of the thirteenth annual symposium* on Computational geometry, pages 274–283. ACM, 1997.
- [45] Yong-Gu Lee and Kunwoo Lee. Computing the medial surface of a 3-d boundary representation model. Advances in Engineering Software, 28(9):593–605, 1997.
- [46] Pan Li, Bin Wang, Feng Sun, Xiaohu Guo, Caiming Zhang, and Wenping Wang. Q-mat: Computing medial axis transform by quadratic error minimization. ACM Transactions on Graphics (TOG), 35(1):8, 2015.
- [47] L Liu, Erin W Chambers, David Letscher, and T Ju. A simple and robust thinning algorithm on cell complexes. In *Computer Graphics Forum*, volume 29, pages 2253–2260. Wiley Online Library, 2010.
- [48] Lu Liu, Erin W Chambers, David Letscher, and Tao Ju. Extended grassfire transform on medial axes of 2d shapes. *Computer-Aided Design*, 43(11):1496–1505, 2011.
- [49] Luis D. Lopez, Luis D. Lopez, Deepak Shantharaj, Lu Liu, Harsh Bais, and Jingyi Yu. Robust image-based 3D Modeling of Root Architecture. *Computer Graphics International*, 2011.

- [50] Balint Miklos, Joachim Giesen, and Mark Pauly. Discrete scale axis representations for 3d geometry. In ACM Transactions on Graphics (TOG), volume 29, page 101. ACM, 2010.
- [51] Joseph SB Mitchell, David M Mount, and Christos H Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.
- [52] João Camargo Neto, George E. Meyer, David D. Jones, and Ashok K. Samal. Plant species identification using Elliptic Fourier leaf shape analysis. *Computers and Electronics in Agriculture*, 50(2):121–134, 2006.
- [53] W Niblack, PB Gibbons, and D Capson. Generating connected skeletons for exact and approximate reconstruction. In Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on, pages 826–828. IEEE, 1992.
- [54] R Ogniewicz and M Ilg. Voronoi skeletons: Theory and applications. In Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on, pages 63–69. IEEE, 1992.
- [55] Robert L Ogniewicz and Olaf Kübler. Hierarchic voronoi skeletons. Pattern recognition, 28(3):343–359, 1995.
- [56] Krzysztof Okarma and Przemysław Mazurek. Application of Shape Analysis Techniques for the Classification of Vehicles, pages 218–225. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [57] I. Ostonen, Ü. Püttsepp, C Biel, O. Alberton, M. R. Bakker, K. Lõhmus, H. Majdi, D. Metcalfe, A. F. M. Olsthoorn, A. Pronk, E. Vanguelova, M. Weih, and I. Brunner. Specific root length as an indicator of environmental change. *Plant Biosystems - An International Journal Dealing with all Aspects of Plant Biology*, 141(3):426–442, nov 2007.
- [58] Kálmán Palágyi and Attila Kuba. A parallel 3d 12-subiteration thinning algorithm. Graphical Models and Image Processing, 61(4):199–221, 1999.
- [59] Ravi Peters and Hugo Ledoux. Robust approximation of the medial axis transform of lidar point clouds as a tool for visualisation. *Computers & Geosciences*, 90:123–133, 2016.
- [60] Stephen M Pizer, P Thomas Fletcher, Sarang Joshi, Andrew Thall, James Z Chen, Yonatan Fridman, Daniel S Fritsch, A Graham Gash, John M Glotzer, Michael R Jiroutek, et al. Deformable m-reps for 3d medical image segmentation. *International Journal of Computer Vision*, 55(2-3):85–106, 2003.

- [61] Stephen M Pizer, Guido Gerig, Sarang Joshi, and Stephen R Aylward. Multiscale medial shape-based analysis of image objects. *Proceedings of the IEEE*, 91(10):1670–1679, 2003.
- [62] Stephen M Pizer, William R Oliver, and Sandra H Bloomberg. Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):505–511, 1987.
- [63] Chris Pudney. Distance-ordered homotopic thinning: a skeletonization algorithm for 3d digital images. Computer Vision and Image Understanding, 72(3):404–413, 1998.
- [64] Dennie Reniers. Skeletonization and segmentation of binary voxel shapes. 2009.
- [65] Dennie Reniers, Andrei Jalba, and Alexandru Telea. Robust classification and analysis of anatomical surfaces using 3d skeletons. In *Proceedings of the First Eurographics conference on Visual Computing for Biomedicine*, pages 61–68. Eurographics Association, 2008.
- [66] Dennie Reniers and Alexandru Telea. Patch-type segmentation of voxel shapes using simplified surface skeletons. In *Computer Graphics Forum*, volume 27, pages 1837–1844. Wiley Online Library, 2008.
- [67] Dennie Reniers, Jarke van Wijk, and Alexandru Telea. Computing Multiscale Curve and Surface Skeletons of Genus 0 Shapes Using a Global Importance Measure. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):355–368, mar 2008.
- [68] Martin Rumpf and Alexandru Telea. A continuous skeletonization method based on level sets. In *Proceedings of the symposium on Data Visualisation 2002*, pages 151–ff. Eurographics Association, 2002.
- [69] Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016.
- [70] Punam K. Saha and Bidyut Baran Chaudhuri. Detection of 3-d simple points for topology preserving transformations with application to thinning. *IEEE transactions* on pattern analysis and machine intelligence, 16(10):1028–1032, 1994.
- [71] Doron Shaked and Alfred M Bruckstein. Pruning medial axes. Computer vision and image understanding, 69(2):156–169, 1998.
- [72] Lior Shapira, Ariel Shamir, and Daniel Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249, 2008.
- [73] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. ACM Trans. Math. Softw., 41:11:1–11:36, 2015.

- [74] Kaleem Siddiqi, Sylvain Bouix, Allen Tannenbaum, and Steven W Zucker. Hamiltonjacobi skeletons. International Journal of Computer Vision, 48(3):215–231, 2002.
- [75] Kaleem Siddiqi and Stephen Pizer. *Medial representations: mathematics, algorithms and applications*, volume 37. Springer Science & Business Media, 2008.
- [76] André Sobiecki, Andrei Jalba, and Alexandru Telea. Comparison of curve and surface skeletonization methods for voxel shapes. *Pattern Recognition Letters*, 47:147–156, 2014.
- [77] Peer Stelldinger, Longin Jan Latecki, and Marcelo Siqueira. Topological equivalence between a 3d object and the reconstruction of its digital image. *IEEE transactions on pattern analysis and machine intelligence*, 29(1), 2007.
- [78] Svetlana Stolpner and Kaleem Siddiqi. Revealing significant medial structure in polyhedral meshes. In 3D Data Processing, Visualization, and Transmission, Third International Symposium on, pages 365–372. IEEE, 2006.
- [79] Avneesh Sud, Mark Foskey, and Dinesh Manocha. Homotopy-preserving medial axis simplification. International Journal of Computational Geometry & Applications, 17(05):423–451, 2007.
- [80] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.
- [81] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. In ACM transactions on graphics (TOG), volume 24, pages 553–560. Acm, 2005.
- [82] A Tagliasacchi, T Delame, and M Spagnuolo. 3D Skeletons: A State-of-the-Art Report. Computer Graphics, 2016.
- [83] Andrea Tagliasacchi, Hao Zhang, and Daniel Cohen-Or. Curve skeleton extraction from incomplete point cloud. In ACM Transactions on Graphics (TOG), volume 28, page 71. ACM, 2009.
- [84] Ayellet Tal. 3D Shape Analysis for Archaeology, pages 50–63. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [85] Roger Tam and Wolfgang Heidrich. Shape simplification based on the medial axis transform. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 63. IEEE Computer Society, 2003.
- [86] Jean-Marc Thiery, Émilie Guy, Tamy Boubekeur, and Elmar Eisemann. Animated mesh approximation with sphere-meshes. ACM Transactions on Graphics (TOG), 35(3):30, 2016.

- [87] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. ACM Transactions on Graphics (TOG), 35(6):222, 2016.
- [88] Anastasia Tkach, Andrea Tagliasacchi, Edoardo Remelli, Mark Pauly, and Andrew Fitzgibbon. Online generative model personalization for hand tracking. ACM Transactions on Graphics (TOG), 36(6):243, 2017.
- [89] Christopher N Topp, Anjali S Iyer-Pascuzzi, Jill T Anderson, Cheng-Ruei Lee, Paul R Zurek, Olga Symonova, Ying Zheng, Alexander Bucksch, Yuriy Mileyko, Taras Galkovskyi, and Others. 3D phenotyping and quantitative trait locus mapping identify core regions of the rice genome controlling root architecture. *Proceedings of the National Academy of Sciences*, 110(18):E1695—-E1704, 2013.
- [90] YF Tsao and King Sun Fu. A parallel thinning algorithm for 3-d pictures. *Computer graphics and image processing*, 17(4):315–331, 1981.
- [91] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.
- [92] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.
- [93] Ying Zheng, Steve Gu, Herbert Edelsbrunner, Carlo Tomasi, and Philip Benfey. Detailed Reconstruction of 3D Plant Root Shape. In *ICCV*, 2011.

Vita

Yajie Yan

Degrees	B.S. Computer Science, Sichuan University. May 2009M.S. Computer Science, Beihang University. January 2012Ph.D. Computer Science, Washington University in St Louis. May 2018
Publications	 Yajie Yan, David Letscher, Tao Ju. Voxel Cores: Efficient, robust, and provably good approximation of 3D medial axes. SIGGRAPH 2018 (conditionally accepted). Yajie Yan, Kyle Sykes, Erin Chambers, David Letscher, Tao Ju. Erosion thickness on medial axes of 3D shapes. SIGGRAPH 2016. July 24-28, 2016, Anaheim, CA.

May 2018

MedialAxis, Yan, Ph.D. 2018