12-8-2016

Gaussian Kernel Based Medial Zone Computation

Weiling Li
ling.379810026@gmail.com

Gaussian Kernel Based Medial Zone Computation

Weiling LI

B.S., Xi'an Jiaotong University, 2014

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Connecticut

2016

APPROVAL PAGE

Master of Science Thesis

Gaussian Kernel Based Medial Zone Computation

Presented by

Weiling LI, B.S.

Major Advisor _____
Horea T.Ilieş

Associate Advisor _____
Jiong Tang

Associate Advisor _____
Xu Chen

University of Connecticut

2016

ACKNOWLEDGMENTS

I would first like to thank my major advisor Prof. Horea T.Ilieş of the Mechanical Engineering department at the University of Connecticut. Prof.Ilieş had been guiding me throughout the entire time of my master's degree program. With his advises, I was able to understand the procedures of doing academic research and conduct my own research as presented in this thesis. During the process of both doing research and preparing thesis, prof.Ilieş was always by my side and providing generous help whenever I went into troubles.

I would also like to thank prof.Tang and prof.Chen who had given me valuable comments and advises on my thesis. As my academic committee members, They had been patient and supportive to my work and provided me important thoughts and questions. By attempting to answer the questions, I was able to build a more complete picture of my work and have a better understanding of my choices made throughout my research.

Nevertheless, I am also very grateful to Dr. Frol Periverzov(Congratulations!), my dear fellow labmate for generously sharing his ideas and knowledges not only in my field of research, but also about programming issues I encountered using C++ or LaTeX.

It is impossible not to mention my dear friends who had been supportive all the time. Last but not least, I would like to thank my parents: Dad and Mom for supporting me both spiritually and financially throughout my life of studying abroad. Without their support, none of these great things would ever happened to me. Also, my dear Shiqi, who had always believed in me and kept me motivated even in my most depress period.

I would like to express my deepest gratitude To all who had helped me in my master's life. It is your generous help that allowed me to had this wonderful journey of my life.

Contents

List of Figures

Abstract

The medial zone of a 3-dimensional semi-analytic domain Ω, denoted as $\mathcal{MZ}(\Omega)$, can be thought of as a "thick" version of the shape's skeleton with important applications in design, motion planning and geometric reasoning. The utility of the medial zones stems from the fact that they theoretically are homeomorphic to and have the same dimension as the original domain, so in this sense they capture the topology of the domain. However, the original approach to compute them involved a discrete Laplace estimation commonly used in edge detection algorithms, which was used to extract the points belonging to the medial zones. Despite being fast, this computational approach produced unwanted topological artifacts and did not satisfy the convergence properties of the medial zones as formulated, which limited their effectiveness.

In this work a new approach was proposed to compute points of the medial zones based on Gaussian Kernels located on the medial axis($\mathcal{MA}$) points whose widths are mapped to an approximate but differentiable function constructed over the domain. This computing paradigm produced a family of homeomorphic medial zones that have the same topology as the domain and converge to either the $\mathcal{MA}$ of the domain or to the domain itself. The resulting method was general in that it can be applied to domains of arbitrary complexity. The effectiveness of this approach was demonstrated by practical examples.

Key words: medial zone, medial axis, Gaussian kernel, homeomorphism, $\mathcal{MZ}$, $\mathcal{MA}$

Chapter 1

Introduction

Medial zone($\mathcal{MZ}$) as a "thickened" skeleton have shown its capability of supporting geometric reasoning, shape analysis and areas related to path or motion planning[15]. It is a concept extended from the well studied medial axis($\mathcal{MA}$)[4, 5, 33] which serves as an intermediary shape that carries 'the best of both worlds' namely the whole domain and its $\mathcal{MA}$. Conceptually, the $\mathcal{MA}$ which holds the crucial topological properties of the representing domain, has lower dimensions than it. Although the $\mathcal{MA}$ along with a radius function provides a compact representation of the geometry of the domain, the $\mathcal{MA}$ suffers from instabilities to small perturbations in the boundary of the domain as shown in figure 1.1. As a powerful alternative method, the concept of medial zone($\mathcal{MZ}$) was proposed[15] which defined a family of intermediate shapes that evolve between the $\mathcal{MA}$ and its representing domain. These shapes are homeomorphic to the original domain and can be used as an initial solution for many applications[15]. One example is shown in figure 1.2, where the $\mathcal{MA}$ is not able to provide the shortest route because of the fact that it has to stick to the equal-distance interior points of the given shape. Meanwhile, computing the shortest path within a $\mathcal{MZ}$ resulted in a shorter and dynamically feasible path which is free of collision. In this particular case, the $\mathcal{MZ}$ provided the shortest path connecting the two highlighted points and it was 13.34% shorter than the solution provided by $\mathcal{MA}$.

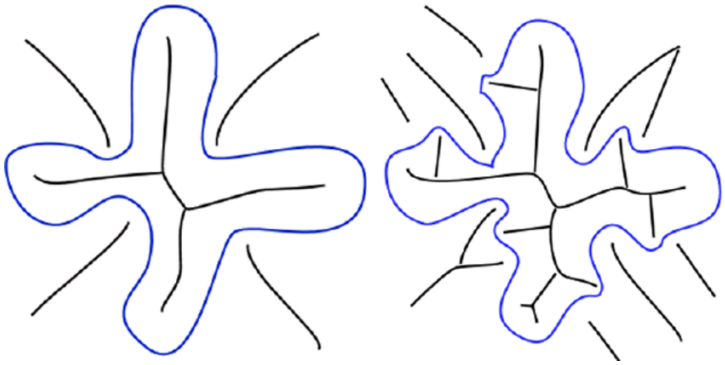
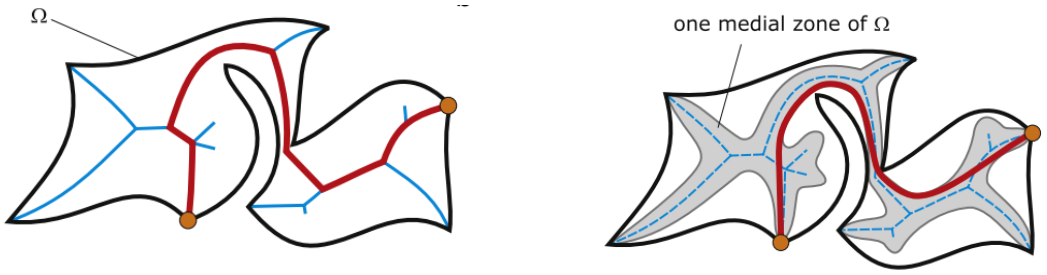


Figure 1.1: [27]Medial axis is unstable under small boundary perturbations.



(a) Solution given by the exact $\mathcal{MA}$.

(b) Solution given by one $\mathcal{MZ}$ of Ω

Figure 1.2: [15] Path planning solution provided by $\mathcal{MA}$ and $\mathcal{MZ}$

The concept of $\mathcal{MA}$ can be intuitively understood by examining a grassfire model as follows. Considering a closed curve as shown in figure 1.3;

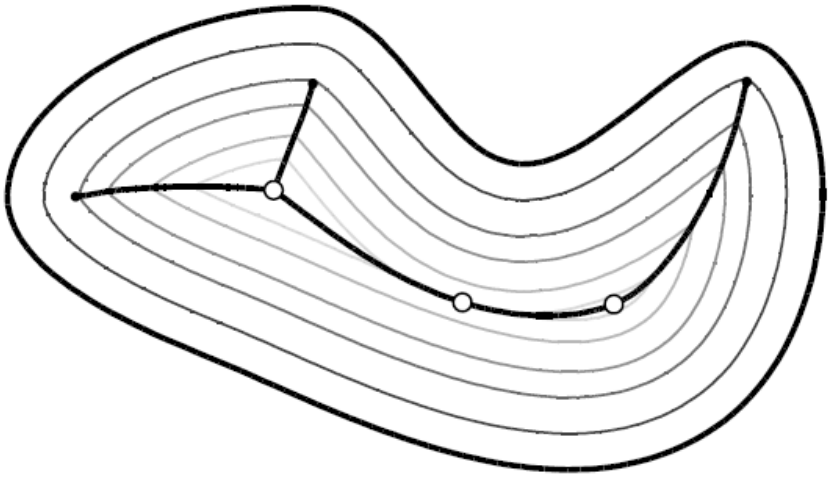


Figure 1.3: [3] A closed curve and its $\mathcal{MA}$

1. A fire was started at the same exact moment at every point on the closed curve;
2. The spreading speed of the fire is a uniform constant at every point and every direction;

Then the $\mathcal{MA}$ consist of all the points which two or more fire fronts meet with each other and the distance function associated with each $\mathcal{MA}$ point can be computed by the speed of the fire and the time consumed to lit the point. Another description of such concept, which is more mathematical, states as follows: $\mathcal{MA}$ is a set of points which has at least two closet points on the curve. Despite the different descriptions, these two intuitive interpretations of the set of $\mathcal{MA}$ describe the exact same definition of $\mathcal{MA}$.

The biggest contribution of $\mathcal{MA}$ is the fact that for any given geometric domain, is homotopically equivalent to the domain and has lower dimension, means that the parts of the geometry and its $\mathcal{MA}$ are connected in a same way[3]. This property had found applications in shape recognition as well as

path planning where $\mathcal{MA}$ draws great interests in providing solutions for queries involving geometrical or abstract domain[34, 20]. Figure 1.2a shows an example of its applications in providing a collision free route connecting two highlighted points. Besides, as the point of $\mathcal{MA}$ carries the local thickness of a shape(and sometimes the direction of the distance function), it has the ability to serve as a representation for shapes or objects in computational geometry. Because this representation carries the local thickness of the shape, it can be used in applications such as meshing, geometric design and even tool-path creation in computer aided manufacturing[25, 32].

Despite having such properties, $\mathcal{MA}$ along was can not be properly applied in some of the applications[15]. A fairly good example had already been shown in figure 1.2, Although the path provided by $\mathcal{MA}$(figure 1.2a) is guaranteed to be collision free, the route is neither differentiable nor the shortest. On the other hand, a 'thickened' version of $\mathcal{MA}$ as shown in figure 1.2b, provided a route that was not only shorter than figure 1.2a but also differentiable which means a continuous velocity and acceleration was allowed.

This 'thickened' version of $\mathcal{MA}$ is then called the $\mathcal{MZ}$. The first clear definition of $\mathcal{MZ}$ was proposed by Eftekharian and Ilieş[15] who defined a family of intermediate shapes that evolve between the $\mathcal{MA}$ and its representing domain. As inherited from $\mathcal{MA}$, $\mathcal{MZ}$ is homeomorphic to the original domain and can be served as an initial solution for many applications such as shape optimization and motion planning.

The concept of both $\mathcal{MA}$ and $\mathcal{MZ}$ are fairly intuitive, however, the computational approach obtaining both sets have always been problematic and difficult[1]. Previously Eftekharian and Ilieş[15] had proposed an efficient computation paradigm for computing $\mathcal{MZ}$ by capturing points around $\mathcal{MA}$ from a differentiable approximate distance function over the domain. However, such paradigm introduced topological artifacts as shown in Figure 1.5. In this study, a new computational procedure was proposed which such that the undesired topological artifacts introduced by previous method were avoided.

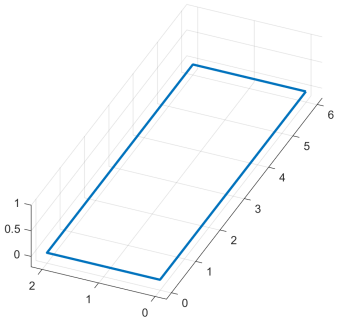
1.1 Motivation

Conceptually, as shown from figure 1.4a to 1.4d, the region of a $\mathcal{MZ}$ was created by two steps: (1) construct a differentiable approximate distance function[2] over the domain. (2) select the points around the crest line from the maximum local curvature(the crest line of the approximate distance function) to a desire lower bound. Under this paradigm, the homotopy equivalence was preserved by the differentiable approximate distance function. With a proper mechanism which can capture the desired $\mathcal{MZ}$ points, a family of differentiable homeomorphic shapes can be achieved.

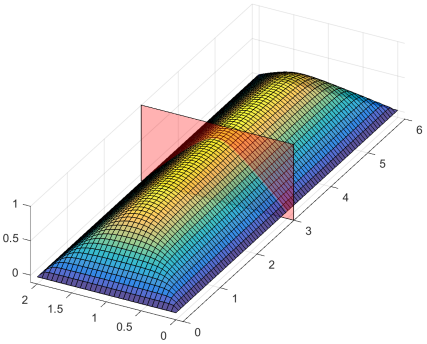
However, in the actual implementation, the mechanism, which is used to identify the $\mathcal{MZ}$ points,

[1] *Stability and Computation of Medial Axis*[3] section 6 & 7 introduced the issues. Giblin et al[18] explored the difficulties of constructing a smooth boundary from $\mathcal{MA}$

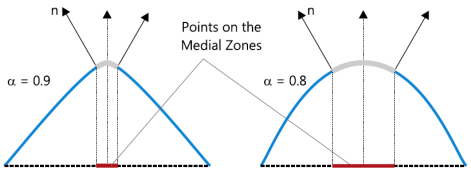
[2] by 'approximate' means the function is constructed based on the distance function but it is modified in a way so that it is differentiable everywhere inside the domain



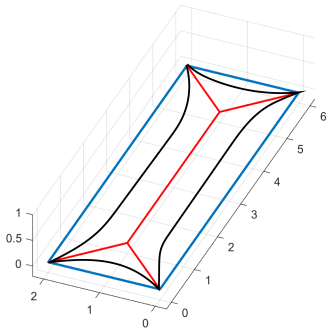
(a) 2D rectangular domain



(b) Differentiable function over the domain



(c) Select desire curvature range[15]



(d) Medial zone by a desire curvature range

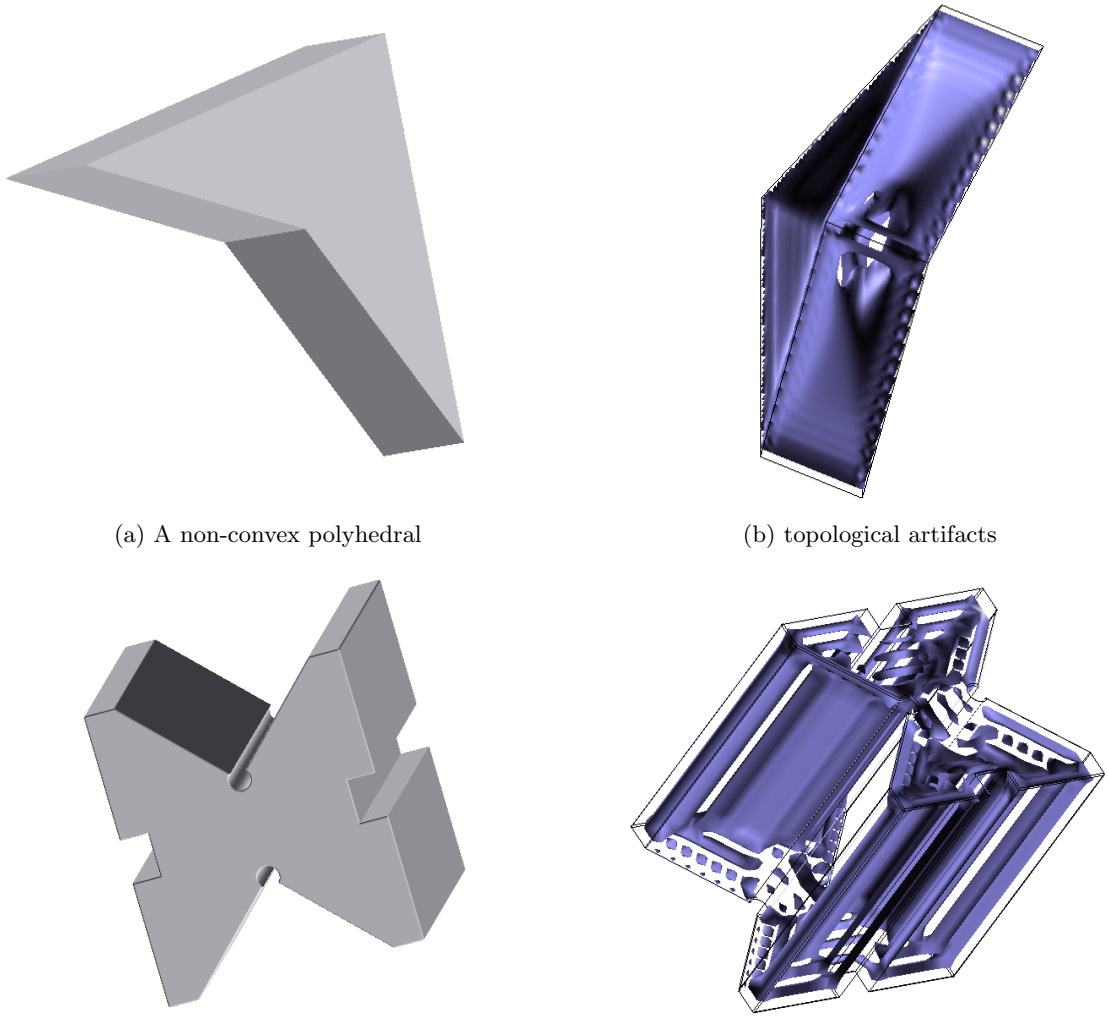
Figure 1.4: Medial zone computed by previous method

failed to accomplish the goal given by the theoretical formulation. As a result, undesired topological artifacts as introduced in previous section were created. Previously, the procedure of computing $\mathcal{MZ}$ consist of the following:

1. Compute an approximate differentiable distance function over the domain.
2. Compute the Laplacian(see chapter 2 for details) value of the approximate differentiable distance function.
3. Extract points whose Laplacian value is below a specified threshold, and these extracted points would be considered as points of the $\mathcal{MZ}$.

The original intention of using the Laplacian value as a label to identify $\mathcal{MZ}$ points came from the fact which Laplacian value is closely related to local curvature and it is much cheaper to compute. Although it is computationally efficient, the resulting $\mathcal{MZ}$ computed by previous implementation have shown that the Laplacian edge detecting algorithm chosen in previous work was not able to capture the points near the ridges and ravines correctly. The Laplacian edge detection algorithm applied to previous work used an approximation of the Laplacian(which is a widely used edge detection algorithm) to compute an approximation of the Laplacian value. It was not clear that whether the simple point selecting mechanism or the Laplacian approximation that was chosen, caused the topological artifacts

occurred in the computed $\mathcal{MZ}$. As shown in the results (see figure 1.5), the $\mathcal{MZ}$ computed by previous method can not guaranty homeomorphism which violated the original intention of formulating $\mathcal{MZ}$.



(a) A non-convex polyhedral

(b) topological artifacts

(c) Complex shape

(d) topological artifacts

Figure 1.5: Example of topological artifacts computed by previous computation approach

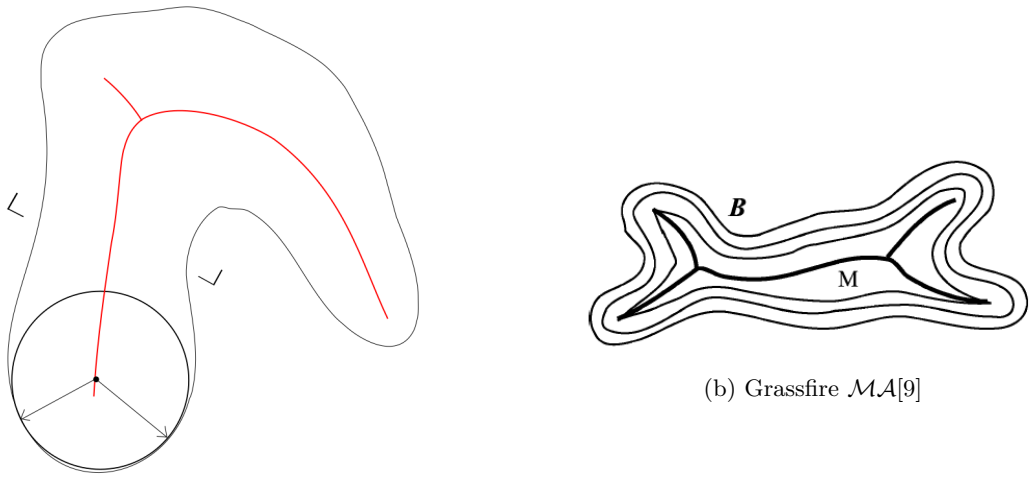
1.2 Background and related work

1.2.1 Computational approach of the $\mathcal{MA}$

Although the definition of $\mathcal{MA}$ is relatively straight forward, computing the $\mathcal{MA}$ point and the associated distance function is very challenging. Generally, the definition of $\mathcal{MA}$ can be interpreted in different ways that would lead to very different computational approaches.

From a very broad point of view, the approaches can be divided into two interpretation mentioned at the very beginning: loci of bi-tangentially osculating maximum balls[18, 17] and the grassfire meeting fronts[36]. An intuitive example can be seen in figure 1.6

The grassfire meeting fronts interpretation can be stated as follows: consider a fire is set on every point on a shape's boundary at $t = t_0$ and the time when all the internal region of the shape is lit, is $t = t_t$. Also assume the spreading speed of the fire is constant in every direction and is uniformly applied to the region. Then within the time frame $t \in (t_0, t_t)$ the sequence of points where the fire fronts meet with each other would provide us the set of $\mathcal{MA}$ of the shape. This interpretation leads to a series of thinning algorithm related approaches which are very powerful in dealing with applications involving 2D image processing. Because the goal of this study was to establish an paradigm for constructing medial zones for 3D shapes, the discussion of thinning algorithms of 2D domain is not relevant. A comprehensive survey of such topics were well documented by Lam et al[21], Tamura[31] and Naccache[23].



(a) loci of maximum balls

(b) Grassfire $\mathcal{MA}$[9]

Figure 1.6: Examples of two different interpretation of $\mathcal{MA}$.

Besides fire front interpretation, $\mathcal{MA}$ can also be interpreted as the loci of maximum balls that have at least two equal distance to the boundary. This interpretation uses the concept of the distance from a given point to the boundary of the domain. One class of approaches is to compute the Voronoi diagram of the domain because it has been shown that the $\mathcal{MA}$ is a subset of the Voronoi edges. A second class of approaches that exploit this distance information is to compute the distance function of the domain and extract the $\mathcal{MA}$ as the projection of the ridges and ravines of this distance function onto the Euclidean space of the domain.

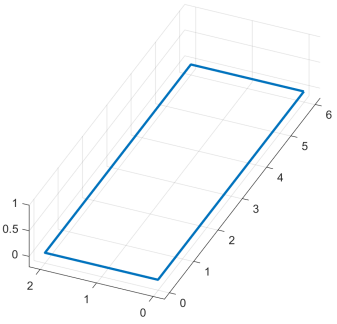
While exploring the algorithms of computing $\mathcal{MA}$ under the 'loci' interpretation, a well known problem arose and can be stated as the following: Although in principle, the $\mathcal{MA}$ can generally be computed exactly for semi-algebraic sets and the resulting $\mathcal{MA}$ would also in semi-algebraic form, we

are not aware of an implementation that can effectively compute the $\mathcal{MA}$ for any general semi-algebraic sets[3]. The term semi-algebraic set means that the domain is represented by the solutions of a finite family of algebraic equations or inequalities. Due to the difficulties of developing an robust and effective paradigm for computing an exact $\mathcal{MA}$, even the most advanced approaches dealing with an exact computation was limited to planar shapes, piecewise linear shapes or shapes which their boundaries are constructed by a finite number of balls. And as briefly introduced by Edelsbrunner[3], in the field of computing exact $\mathcal{MA}$, effective paradigm only appeared for rare and limited occasions, the difference between the paradigms were so large that non of them have the ability to be generalized.

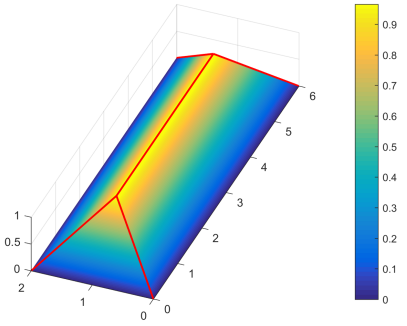
As discussed in previous section, a general procedure of such methods would consist of two major parts: (1) Constructing the distance function (2) Extracting the $\mathcal{MA}$ from the distance function. Consider an arbitrary domain in 2D, for any point inside the domain, the shortest distance to the boundary is given by the distance function over the domain. If we plot the distance function in the 3rd dimension as shown in figure 1.7.b, then the location of ridges and ravines would imply the location of the $\mathcal{MA}$[14][35]. Eftekharian et al used a basic R-function[3] system to combine the individual distance functions of each piece of the boundary while Xia and Tucker constructed the distance function via Eikonal equation. Although they took different routes of computing such function, the distance function is unique for a given domain which leads to very similar approaches when extracting the medial features. Noted that the $\mathcal{MA}$ is consist of the points which has a equal distance to multiple boundary points(at a local edge or crest point, the 'multiple boundary points' coincide), so the $\mathcal{MA}$ would naturally represent the local maximum or a singular point(such as a cusp) on the distance function. As stated in *Finite Volume Distance Field and its application to $\mathcal{MA}$ transforms*[35], Laplacian computation based feature extraction can serve as a fast and fairly robust tool to extract the ridges and ravines on the distance function. In figure 1.7.a to 1.7.c, such procedure was illustrated. The Ridges of the distance function over the rectangular domain was projected back to the original domain becoming $\mathcal{MA}$.

Constructing the distance function over the domain generally requires an algebraic or analytical expression of the boundary(i.e. implicit solid model representation. Such representations can describe the whole domain in piecewise continuous functions). However, such representation of the domain's boundary is not always available. Alternatively, some researchers developed fast $\mathcal{MA}$ computation of polyhedral domain(Polygon in 2D and polyhedron in 3D). For such domains, $\mathcal{MA}$ was computed as part of the Voronoi diagram or Delaunay triangulation. This idea comes from the fact that the $\mathcal{MA}$ is a special subset of the generalized Voronoi diagram that falls inside the domain. Under this general setup, Culver[7] put forward the computation of the exact $\mathcal{MA}$ for a polyhedron. The difficulties of computing an exact $\mathcal{MA}$ had driven the field into seeking an approximation algorithm of the exact $\mathcal{MA}$. The framework consist of a discretization of the space or the domain and the approximated set of $\mathcal{MA}$ is

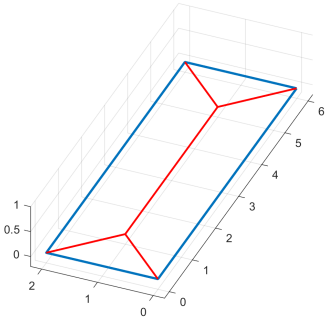
[3]R-function would be discussed in Chapter 2



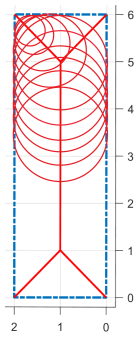
(a) A rectangular domain



(b) Distance function over the domain(Red line indicates the location of ridges and ravines)



(c) Medial axis of the domain



(d) Boundary recovery via maximum circles

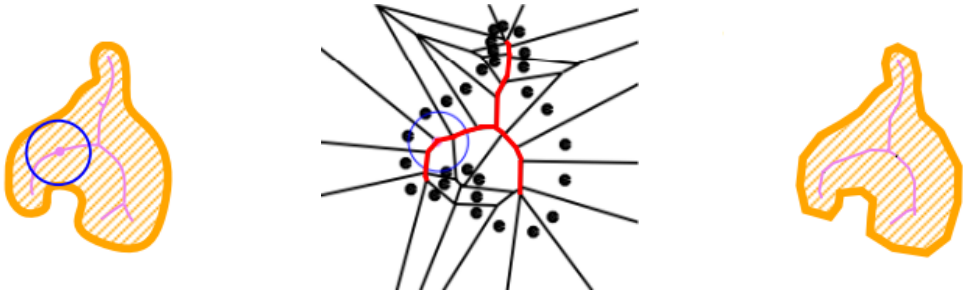
Figure 1.7: Procedure of computing $\mathcal{MA}$ and boundary recovery via distance function

obtained from the discretized space or shape. The methods of computing Voronoi diagram or Delaunay triangulation for $\mathcal{MA}$ was then extended to a discretized domain such as point clouds. Researchers such as Sheehy[29], Turkiyyah[33] conducted $\mathcal{MA}$ computations on an unorganized point set which is obtained by a sampling algorithm on the boundary surface. The $\mathcal{MA}$ computed by Sheehy's approach was approximated by the Delaunay triangulation on a set of sampled surface points. This set of sample points was dynamically adapted(basically means adapting the local sampling density) in order to provide a correct topological representation of the medial surface.

Although the community claim that it is very computational efficient to compute $\mathcal{MA}$ via a discretized domain representation with a well developed algorithm for generating Voronoi diagram, the resulting $\mathcal{MA}$ set would requires heavily trimming post-processing because this method would generate a significant amount of extra branches which are introduced by the discretization. An intuitive example can be found in Amenta's work[1] and it was also illustrated in figure 1.8. Noted that in figure 1.8b, only the highlighted branches are valid $\mathcal{MA}$, all the others should be trimmed. This instability had also been recognized as an serious issue of $\mathcal{MA}$ and would also be discussed in chapter 2.

Besides these main-stream approaches, there exist other algorithms which computes an approximate $\mathcal{MA}$. These method take advantages of its individual approximate method to trade the accuracy of the resulting $\mathcal{MA}$ with efficiency. Illustrating these approaches had beyond the scope of this study, readers

who is interested in this area can referred to Yang[37] and Foskey's[16] work for more details.



(a) Original shape and its $\mathcal{MA}$ (b) Sampled set and its Voronoi diagram (c) Final results

Figure 1.8: [1] Medial axis computation via Voronoi diagram.

1.2.2 Boundary reconstruction using $\mathcal{MA}$

The medial representation encodes the geometric information of the generating domain, and hence it is able to be used to reconstruct the domain itself.

There exist only a few literatures which explored the reconstruction via medial representation mathematically. The most well known are Giblin et al[18, 17, 19, 22] and Damon et al[24, 8, 9, 10, 11, 12, 13]. They discussed the mathematical details of the relationships between $\mathcal{MA}$ and its representing domain's boundary which can be applied to achieve desired boundary continuity in the process of boundary reconstruction.

In order to mathematically evaluate and explore the relationship between $\mathcal{MA}$ and its representing boundary, Giblin and Kimia[19] put forward the expression of the $\mathcal{MA}$ points by solving an expression for the center of the maximum ball osculating a locally parametrized boundary and the solution of the balls constructed the local forms of the osculating types(for example, a junction point where multiple $\mathcal{MA}$ branches meet is a different local form compare to each individual smooth branch). They then claimed[18, 17] that this categorization of $\mathcal{MA}$ points benefited the reconstruction paradigm in 2D that one may interested, because the criterion of obtaining a smooth boundary for different local forms of $\mathcal{MA}$ were very different. Giblin and Kimia provided the criterion regulating the distance function to obtain a differentiable boundary for a smooth branch of $\mathcal{MA}$(*On the intrinsic reconstruction of shape from its symmetries*/[18] lemma) and also provided a set of equations regulating the distance function at a non-differentiable point on the $\mathcal{MA}$ which corresponding to a differentiable point on the boundary([18] section). They also provided a reconstruction process which the original boundary can be constructed by the envelope of circles whose radius is associated with distance function and centered at the corresponding $\mathcal{MA}$ point locations.

Damon et al.[8, 13] extended the boundary reconstruction process that the distance function can be altered to an arbitrary function that served the similar purpose as distance function resulting altered

shapes produced by the same $\mathcal{MA}$. Their work focused on exploring the necessary and sufficient conditions for obtaining an differentiable boundary from an arbitrary $\mathcal{MA}$ and arbitrary adjustable distance function. Three important conditions regulating the arbitrary function for obtaining a differentiable boundary were given as: radial curvature condition, edge condition and compatibility condition. They claimed and proved that for any combination of a set of $\mathcal{MA}$ and a 'vectorized' distance function, if the three conditions were met, the recovered boundary would be at least have C^1 connectivity everywhere.

1.2.3 Medial zone

Besides aiming at recovering the original boundary via a medial representation, researchers started to notice the intermediate shapes which one can get either during the process of extracting $\mathcal{MA}$ or recovering an object's boundary. As during the grassfire process the intermediate boundary is in some ways 'similar' to the domain(We shall explain this similarity in a more rigorous way). The work of James Damon[9, 10, 11] constructed a radial map which allows ones to define the intermediate levels of the flow from the $\mathcal{MA}$ to the actual boundary. Each level of the radial map is defined by a parameter t and the boundary of each level is given by $\mathbf{B}_t = x + t\mathbf{U}(x) : x \in \mathbf{M}, 0 \leq t \leq 1$ where $t = 0$ was the initial starting point of the radial flow, namely the $\mathcal{MA}$ and $t = 1$ was the actual boundary of the object. The shapes which was defined by different t form a family of intermediate shapes which is in fact one particular family of $\mathcal{MZ}$.

Turkiyyah[32, 26, 30] put forward a shape morphing process based on the $\mathcal{MA}$ for shape design and optimization. his process of shape morphing computed the $\mathcal{MA}$ of the combination of the two initial shapes as the intermediate surface. The intermediate shapes were generated by different weights of the two shapes that contributed in computing the $\mathcal{MA}$. If we take the $\mathcal{MA}$ and its representing boundary as the two shape of the morphing process, the intermediate shapes would most likely form a family of $\mathcal{MZ}$.

The word 'similar' is used to describe the relationship between the $\mathcal{MZ}$ and the domain, which in fact only imply a considerably loose constraint that gives little information about how the $\mathcal{MZ}$ should be. As a matter of fact, the '$\mathcal{MZ}$' which is generated by different approaches(such as grassfire procedure, radial flow or a evolved boundary introduced by Turkiyyah using his medial representation) can have very different families of the $\mathcal{MZ}$ even for the same domain. Depending what application which this $\mathcal{MZ}$ family is applied to, it is hard to be judgmental about which family is more effective than any other ones.

Aiming at providing an initial solution for applications like shape optimization and path planning, a definition of the $\mathcal{MZ}$ was put forward by Eftekharian and Ilieş[15]. In that work, the 'similarity' between the $\mathcal{MZ}$ and its representing $\mathcal{MA}$ or domain was defined to be homotopically equivalent. Besides the definition, a novel procedure of obtaining the $\mathcal{MZ}$ family was also proposed. This procedure used

an approximate distance function on which the family of $\mathcal{MZ}$ was able to be captured. The original formulation of the approximate distance function used a continuous differentiable function defined over the entire domain which was believed to capable of ensuring the topological relationships as well as the geometrical ones.

1.3 Approach

Although a fast evaluation of the $\mathcal{MZ}$ boundary via a differentiable distance function is very attractive, previous implementation of the $\mathcal{MZ}$ computation failed to preserve homeomorphism which was considered to be the fundamental property which $\mathcal{MZ}$ should hold. Thus, in this work, an approach was presented which possessed the ability to generate an effective family of $\mathcal{MZ}$ based on $\mathcal{MA}$. This approach ensured homeomorphism and provided intuitive control means for generating desired members of the $\mathcal{MZ}$ family within the two extremes, namely the $\mathcal{MA}$ and the original domain. This thesis proposed to replaced the previous mechanism of identifying $\mathcal{MZ}$ points with a Gaussian Kernels based point selection mechanism. The boundary of the $\mathcal{MZ}$ was then the envelope of all the Kernels where the properties of such envelope was discussed in Giblin's and Kimia's work[22, 18, 19]. Although, in their work, only the recovery of the original boundary was discussed, the conditions which they derived can be applied to examine the boundary property of the computed $\mathcal{MZ}$ in this study. Besides the conditions given by Giblin, J.Damon also proposed sufficient and necessary conditions to recover a smooth boundary from an arbitrary $\mathcal{MA}$. Generally speaking, the approach of computing the $\mathcal{MZ}$ in this study contained the following basic steps which had already been illustrated in figure 1.7: from a to d shows a basic procedure of extracting $\mathcal{MA}$ of a geometry via constructing distance function and perform a boundary recovery based on the computed $\mathcal{MA}$. After importing the shape, a distance function can be constructed based on the shape shown in (b). Then the ridges and ravines(highlighted in Red lines in (b)) can be extracted and be projected back to the original domain(shown in (c)). Based on the $\mathcal{MA}$ as well as a distance function associated with it, the original boundary(shown in dash line in (d)) can be reconstructed via the envelope of all the maximum balls related to the $\mathcal{MA}$(in this case: all the Gaussian kernels).

1. Compute the exact distance function of the domain.
2. Extract the $\mathcal{MA}$ and a new approximate distance function on the $\mathcal{MA}$ based on the computed distance function.
3. Construct Gaussian Kernels over the $\mathcal{MA}$'s domain with its individual kernel width controlled by the new approximate distance function.
4. Obtain the $\mathcal{MZ}$ boundary by tracing the envelope of all the kernels.

1.4 Contributions of this approach

- Our approach is able to construct $\mathcal{MZ}$ for 2D or 3D geometries which the distance function of the geometry over the domain is applicable.
- The transition through the $\mathcal{MZ}$ family for a given geometry can be intuitively controlled and the homeomorphism is naturally preserved by the formulation of the $\mathcal{MZ}$ computation.
- The envelop of all the Gaussian kernels are controlled by an adjustable approximate distance function so that theoretically the boundary smoothness for a corresponding singular point on the $\mathcal{MA}$ can be enforced by adjusting the distance function.

Chapter 2

Preliminaries

In this chapter, concepts, which were used to understand and analyze the medial zone($\mathcal{MZ}$) construction, were discussed and explained.

2.1 Medial axis properties

For any given shape, there exist a unique medial axis($\mathcal{MA}$) which is invariant with respect to coordinate systems or computational method. Hence, the properties of $\mathcal{MA}$ hold for any shape and is always true no matter what kind of Medial Axis Transformation was used to compute it. These properties were illustrated as the follows.

2.1.1 Homotopy equivalence

As had already been introduced in previous chapter, $\mathcal{MA}$ has the following properties: Homotopy equivalence, finiteness, instability and semi-continuity.

Out of these known properties, the most significant and contributive one is homotopy equivalence. Homotopy equivalence describe the similarity of two set in topological point of view. The essence of this property states that the branches of $\mathcal{MA}$ are connected in the same way as their representing domain which means that there exist a continuous deformation from the domain to its $\mathcal{MA}$.

2.1.2 Instability and Semi-continuity

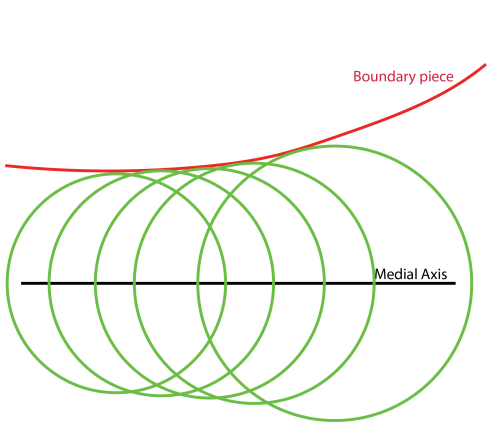
The instability and semi-continuity were recognized by evaluating the hausdroff distance[2] between the different sets of $\mathcal{MA}$ obtained by a family of shapes which differs by only a small changes in an arbitrary local part of the boundary. Despite the hausdroff distance can be relatively small between the family of the shapes geometrically, the hausdroff distance of the $\mathcal{MA}$ of the shapes are usually much greater which implies that the $\mathcal{MA}$ is not stable with small boundary changes. This issue limits the effectiveness of

$\mathcal{MA}$ in areas such as design and optimization. It is especially severe in the computation of $\mathcal{MA}$ involve a discretized domain, because the computed $\mathcal{MA}$ is highly sensitive to even the sampling density.

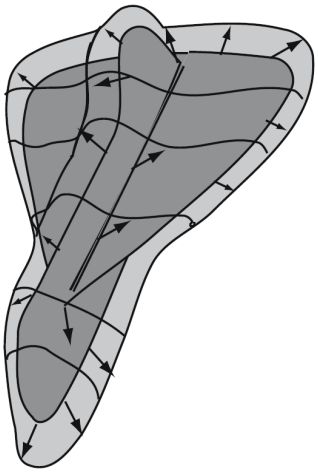
2.2 Boundary reconstruction from medial representation

Boundary reconstruction served an important part of computing $\mathcal{MZ}$. Although the study was not about reconstructing the actual boundary of the original domain, the procedure of computing $\mathcal{MZ}$ and reconstructing the boundary was very much alike. Thus the boundary properties of the computed $\mathcal{MZ}$ can also be analyzed by applying the knowledges of reconstructing the original boundary.

As introduced in the previous chapter, there exist only a few literatures that analyzed the properties of the reconstructed boundary. Out of them, Giblin[18] and Damon[13] provided marvelous works in analyzing the conditions for imposing certain continuity to the reconstructed boundary. In the actual reconstruction paradigm, Giblin computed the recovered boundary as the envelope of the maximum balls that centered on the $\mathcal{MA}$ while Damon computed the boundary as a combination of a $\mathcal{MA}$ and a vectorized distance function($\mathbf{V}$)[1]. The boundary of the recovered shape under Damon's structure was obtained by substituting all the points on $\mathcal{MA}$ into $\mathbf{V}$. Although these two paradigms were different in the formalism, the recovered boundary was unique meaning that the vector formed from the point on $\mathcal{MA}$ to its contributing point on the envelope was exactly the same as $\mathbf{V}$. Thus, it was possible and reasonable to combine the work of both Giblin and Damon to obtain a method of checking and adjusting boundary properties of the computed $\mathcal{MZ}$.



(a) Giblin's recover method



(b) [13]Damon's recover method

Figure 2.1: Boundary recovery by Giblin's and Damon's method

In order to discuss the property of the reconstructed boundary, the set of $\mathcal{MA}$ needs to be separated

[1]Normally, the distance function($\mathcal{DF}$) only gives a scalar value providing the Euclidean distance from a interior point to the closest boundary of a given shape(see figure 2.1). Here in Damon's work[8], the 'distance function' he used was a vector function which for a given point on $\mathcal{MA}$, it provided the vector pointing from the point on $\mathcal{MA}$ to the corresponding boundary point.

into proper categories. Generally speaking, the conditions for maintaining a certain connectivity on the reconstructed boundary varies by the different types of $\mathcal{MA}$ point. Following Giblin's work, we distinguish the different $\mathcal{MA}$ points by its different local forms. The local forms were defined by the different combinations of contact orders which related the osculating ball's radius to the local curvature of the boundary.

It is not only important but also necessary to identify these local forms because in the boundary reconstruction process(would be explained later in details), different local forms would lead to different conditions for imposing a same boundary property.

2.2.1 Contact order

Contact orders was computed by solving the center of the osculating ball(circle in 2D, sphere in 3D) by a locally parameterized boundary it touched. The contact order was defined in the form as A_k, which k represent the order of such contact. The term "order" comes from the derivation of the contact. By parameterizing the surface into polynomials, the different order of contact is exactly the order of polynomials which solve the center of the contact ball. This method was first brought out for computing symmetry sets, but as the set of $\mathcal{MA}$ was known to be a subset of the symmetry set, it can be also used to define the set of $\mathcal{MA}$ with restriction which only odd order of contact is considered to be in the set of $\mathcal{MA}$(details of the derivation can be found in[19]). These orders of contact are A_1, A_3, A_5..., but in most cases, only A_1, A_3 and the combinations of these two contact types can be seen. For a basic contact type, figure 2.2 shows an intuitive example in 2D of such contact types.

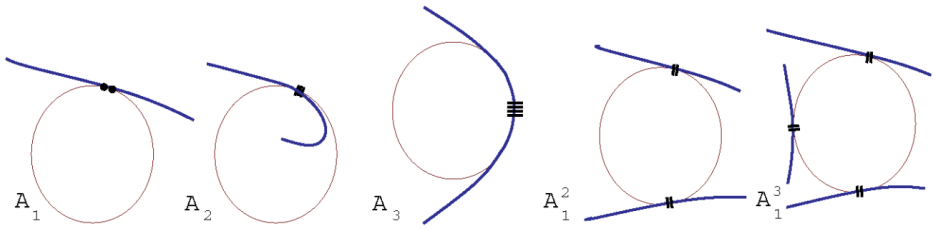
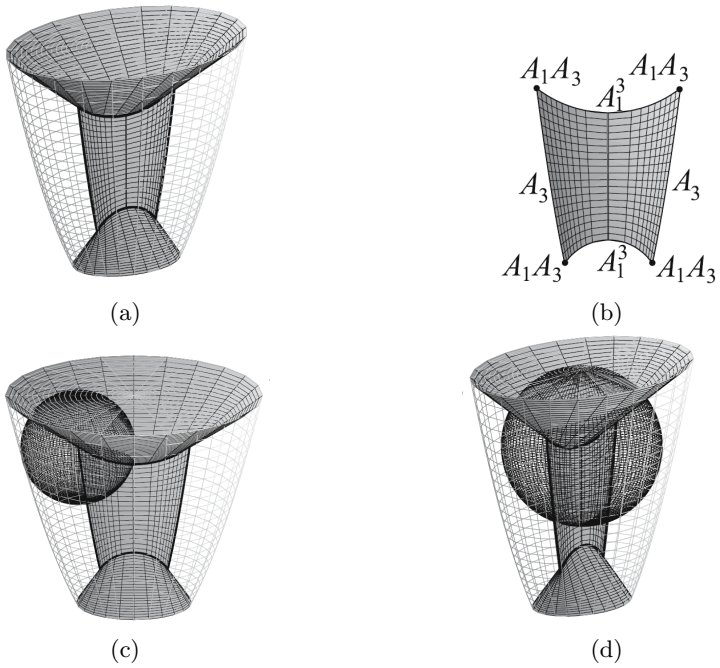


Figure 2.2: Basic Contact Types in 2D [18]

The contact types shown can be categorized into A_1, A_2, A_3. They are representing respectively as **ordinary contact**(or **2-point contact**), **osculating contact**(or **3-point contact**) and **4-point contact**. As for the set of $\mathcal{MA}$, only the ordinary contact points and the 4-point contact is concerned. The ordinary contact stands for the contact which the osculating ball's center locates at the normal line to the boundary at the contact point. The 4-point contact means that in addition to osculating contact, the contact point is a vertex or a local curvature extremum. With such categorization, the set of $\mathcal{MA}$ can now be mathematically expressed as the combination of these contact orders, called local forms.



(a) (b)

(c) (d)

Figure 2.3: [19] Example of Local form of $\mathcal{MA}$.

2.2.2 Local form of $\mathcal{MA}$

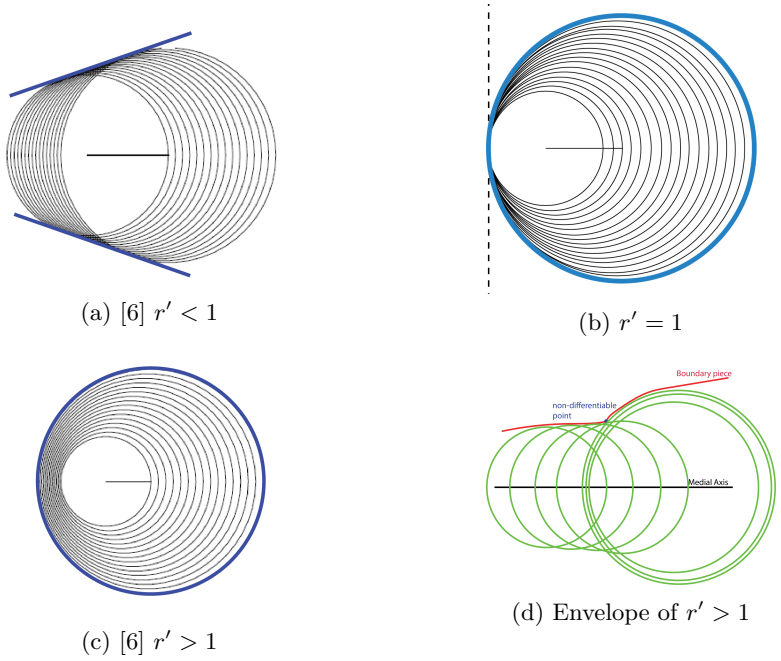
As stated above, the set of osculating balls, which their centers are the set of $\mathcal{MA}$, can be expressed locally by the combination of the contact orders. Giblin et al[17] provided in their work a full sets of $\mathcal{MA}$ as the following

- A_1^2: a smooth sheets of $\mathcal{MA}$.
- A_1^3: the local representing medial as is three half-sheets meeting along a smooth curve(a Y-junction curve).
- A_3: indicates a boundary point on a smooth sheet of the $\mathcal{MA}$.
- A_1A_3: indicates a fin-point.
- A_1^4: indicates a 6-junction point.

Under this local formation of the $\mathcal{MA}$, the whole set of $\mathcal{MA}$ is represented by a *hypergraph* where it is consisted of **nodes**(A_1A_3 and A_1^4 points),**links**(A_3 and A_1^3 curves) and **hyperlinks** (A_1^2 sheets), an example can be seen in figure 2.3a shows the $\mathcal{MZ}$ of a non-polyhedral object. The center piece of the $\mathcal{MA}$ branch is labeled by its different contact order. A_1A_3 and A_1^3 were shown in figure 2.3c and 2.3d.

2.2.3 Giblin's conditions

Giblin[18] discussed the boundary reconstruction in 2D and derived two conditions which were sufficient for hyperlinks to be differentiable.



(a) [6] $r' < 1$

(b) $r' = 1$

(c) [6] $r' > 1$

(d) Envelope of $r' > 1$

Figure 2.4: Envelope of different r' and non-differentiable boundary from $r' > 1$.

For a 2D case, consider a $\mathcal{MA}$ patches $\mathcal{MA}$ and we are aiming at recover the representing boundary $\mathcal{B}$. Let $\mathcal{MA}$ be parameterized by $\gamma(s)$ where s is the arc-length. Furthermore, as the boundary is recovered by envelopes, we defined a function $r(s)$ which defines the radius of the circles centered at $\gamma(s)$. Let prime sign(eg: $r'(s)$) denotes d/ds. Following his work in[18], we can achieve the following basic restriction for boundary reconstruction:

The first condition states as:

$$r'(s)^2 \leq 1 \tag{2.1}$$

This inequality relationship restrict the growth of the radius along s such that the growth of $r(s)$ cannot exceed the growth of s. Where $r'(s) = \pm 1$ means that the absolute growth of the radius function equals the growth of the arc-length s. This would often indicates an edge point of the $\mathcal{MA}$ which corresponds to an edge or crest point on the recovered boundary. Figure 2.4 shows the examples of different $r'(s)$ value and its contribution to the family of the circles. In the figure, the highlighted curves are the envelope of the circles. The cases shown in figure 2.4a and 2.4b are preferred because having $r' > 1$ would lead to a non-differentiable on the recovered boundary as shown in figure 2.4d.

Another condition was given to avoid critical points(see figure 2.5). Critical points were highlighted in red and the envelope of the right figure was highlighted in blue) which introduces by computing envelopes of the circles and can be stated as follows:

$$\left(1 - r'^2\right) - rr'' \pm r\kappa\sqrt{1 - r'^2} \geq 0 \tag{2.2}$$

where κ stands for the curvature of the local $\mathcal{MA}$ point.

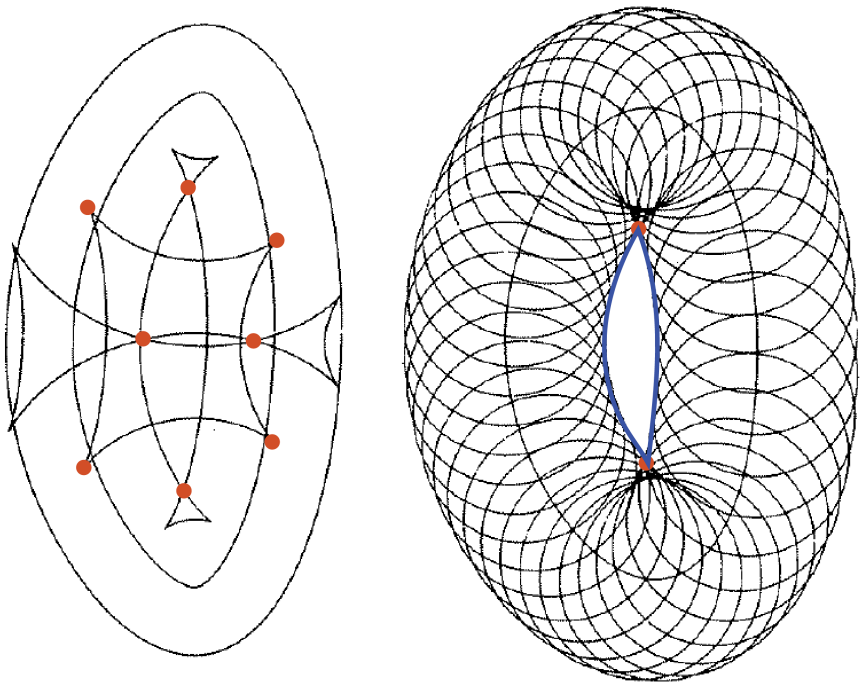


Figure 2.5: [6]Critical points on the envelope.

There were alternative forms which describe the same condition, along with its derivation can be found in [18] lemma 2. This condition would regulate the radius function by the local curvature of the $\mathcal{MA}$ branch so that it will not create critical points.

However, in the actual implementation, only the 1st condition was enforced because a critical point would not likely to produce topological artifacts such as discontinuous parts and holes. However, if C^1 continuity is concerned in the actual application, then the second condition should be also enforced.

Besides the basic conditions, only a relationship between A_1^3 point and its corresponding differentiable boundary was briefly discussed. The discussion only states that for a smooth boundary at an A_1^3 point(usually it would be a Y-junction) the radius functions along the three meeting branches and its differential properties are not independent. There were 6 variables describing the 3 pieces of boundary around a A_1^3 point and if 3 of them were fixed, the other 3 can be solved. However, this condition was hard to apply in practical cases, because even the non of the 3 of 6 variables were independent and this condition was certainly not able to be generalized into 3D cases. Despite this, the first condition mentioned above was very helpful and easy to apply, thus in the actual implementation, the first condition was used to regulates the envelope computed to behave as discussed.

2.2.4 Damon's conditions

Damon et al[8] provided much more complete and detail conditions for constructing a differentiable boundary via a generalized medial representation[2]. In their work[8, 9] the conditions had been proven to be necessary and sufficient to obtaining a overall smooth boundary with the weakest connectivity of C^1 occurs at the singularity points of the $\mathcal{MA}$. Under his structure, the set of $\mathcal{MA}$ points are separate into 3 categories:(1) Regular $\mathcal{MA}$ point M_{reg} which contains only the A_1^2 points. (2) Edge point $M_{sing,edge}$ which contains only the A_3 points. (3) Non-edge singularity points $M_{sing,non}$ which contains the rest of the contact types mentioned above. Based on this categorization, the conditions can be state as follows:

For a generalized medial representation $[\mathcal{MA}, \mathbf{V}]$, where $\mathcal{MA}$ stands for the set of $\mathcal{MA}$ and $\mathbf{V}$ is the vectorized distance function. To obtain an overall C^1 continuity boundary, the following three conditions were necessary and sufficient[8]:

- Radial curvature condition: for all regular points M_{reg}, $r < \min 1/(\kappa_{ri})$ for all positive principal radial curvatures κ_{ri}.
- Edge condition: for all the edge points $M_{sing,edge}$, $r < \min 1/(\kappa_{Ei})$ for all positive principal edge curvatures κ_{Ei}.
- Compatibility condition: $\eta_U \equiv 0$ for all singularity points M_{sing}.

The greatest benefits of Damon's conditions not only lies on the fact that this sufficient condition is developed for 3D boundary reconstruction but also the fact that they provided a clear criterion which can be directly used in examining the boundary smoothness of the computed $\mathcal{MZ}$. Thus if boundary smoothness is a concern, one can apply damon's conditions to examine and make adjustments to the distance function locally to obtain the desire boundary smoothness.

2.3 Medial Zone($\mathcal{MZ}$)

2.3.1 Definition of $\mathcal{MZ}$

Before getting into any details of the $\mathcal{MZ}$ formulation, the definition of $\mathcal{MZ}$ should be clearly proposed. In the previous work[15], the definition of $\mathcal{MZ}$ was made by listing all of its properties and relationship towards the original domain. For this study, we are not intended to alter the definition and we restricted ourselves to discuss and compute the $\mathcal{MZ}$ family which falls inside the domain. Knowing the purpose, the definition of the $\mathcal{MZ}$ was stated as follows:

[2]Usually, $\mathcal{MA}$ and $\mathcal{DF}$ needed to be computed from a certain geometric shape. By 'generalized' means that $\mathcal{MA}$ and $\mathcal{DF}$ do not need to be computed from the actual shape, they can be arbitrary sets as long as having the same format as usual $\mathcal{MA}$ and $\mathcal{DF}$.

For a closed, bounded, semi-analytic domain, the $\mathcal{MZ}$ is also a closed, bounded semi-analytic set which is homeomorphic to the domain and is also homotopy equivalent to and contains the $\mathcal{MA}$.

Generally, homeomorphism as the restriction on constructing the $\mathcal{MZ}$ provides no information about how the actual $\mathcal{MZ}$ should be, such as size and shape. These restrictions, however, is necessary for general geometric reasoning. As a result, the behavior of the $\mathcal{MZ}$ is regulated by its desire properties which would be illustrated in section 2.3.2.

2.3.2 Properties of $\mathcal{MZ}$

For a domain Ω, the families of medial zones, $\mathcal{MZ}(\Omega)$, was defined to have the following property:

1. Relationship to the domain: $\mathcal{MZ}(\Omega) \subseteq \Omega$.
2. Relationship to the $\mathcal{MA}$ of the domain: $\mathcal{MA}(\Omega) \subseteq \mathcal{MZ}(\Omega)$.
3. Dimension: the $\mathcal{MZ}$ $\mathcal{MZ}(\Omega)$ of Ω should be dimensionally homogeneous and have the same dimension as Ω itself, except when $\mathcal{MZ}(\Omega) = \mathcal{MA}(\Omega)$.
4. Homeomorphism: Ω and the family of all its medial zones $\mathcal{MZ}(\Omega)$ should be homeomorphic, except when $\mathcal{MZ}(\Omega) = \mathcal{MA}(\Omega)$.

Besides defining the properties of the desire $\mathcal{MZ}$, the concept 'family of $\mathcal{MZ}$' was also a key idea behind the $\mathcal{MZ}$ construction. As we know that $\mathcal{MZ}$ was constructed in a similar way as $\mathcal{MA}$. In the construction process, it was preferred to have control on the actual 'thickness' of the $\mathcal{MZ}$ so that it would be able to fit in different applications with a desire thickness. To quantify this idea,the computational paradigm was proposed to be able to construct a family of effective $\mathcal{MZ}$ which the largest set is the original domain and the smallest set is the $\mathcal{MA}$. If the construction process was treated as a shape morphing from the $\mathcal{MA}$ to the domain, then we would require the computation paradigm to have the ability of generating shapes that can converge to both extremes. This property was then defined as **Double Convergence** property of the family of $\mathcal{MZ}$.

2.3.3 Outline of the previous computation paradigm

The Previous computation paradigm can be summarized in the flowchart shown in figure 2.6. The Point Membership Classification(PMC) test was constructed by a constant Laplacian approximation value which points have Laplacian value smaller than the threshold would be consider inside $\mathcal{MZ}$; equals to the threshold means on the boundary of $\mathcal{MZ}$ and greater than the threshold indicates outside $\mathcal{MZ}$. The rendering method called 'Marching Cubes' traces along the boundary of $\mathcal{MZ}$ by tracing the points that their value equal to the threshold.

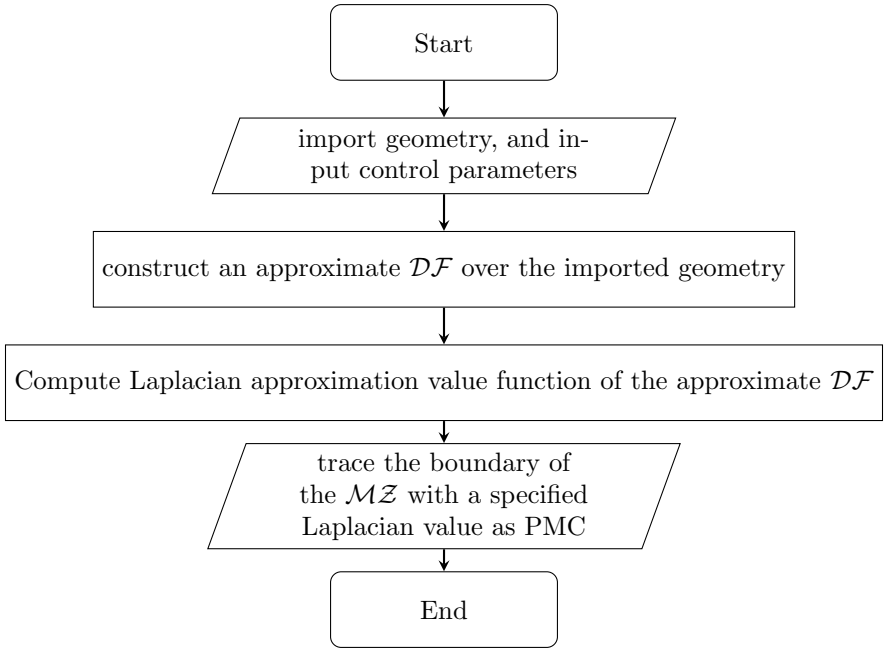


Figure 2.6: Previous procedure for $\mathcal{MZ}$ construction

2.3.4 Distance function construction

Originally, the paradigm illustrated above was used to compute $\mathcal{MA}$ for a semi-analytic domain. The distance function of the domain was constructed by labeling the shortest distance toward the boundary. In the actual implementation, this was achieved by combining the individual distance function, defined by each smooth piece of the boundary, using a MIN operation. In order to fulfill the task of taking the minimun value of all the individual distance functions, R-function was used in previous work providing means to construct the desire distance function. The basics of R-function would be explain later so that the idea behind using R-function would be self-proving.

As explained above, the method used in computing $\mathcal{MA}$ of this study is extracted $\mathcal{MA}$ from the distance function over the domain which its boundary representation allows to compute individual distance function for each closed face.

One can construct the distance function over the domain from taking the min value of all the individual distance functions at each and every point inside the domain. An alternative way of computing was using the **R-function**[28]. Invented by V.L. Rvachev, the R-function is meant to provide means to construct a $\mathbb{C}^n$ function over the domain which is defined by primitive half-spaces. For this study, the **R-function** provided a mean of operating the distance function via real valued function instead of a traditional one so that the distance function over the domain can be calculated by substitute all the individual distance function defined by each boundary piece.

There exist many systems of **R-function** with various combining rules. Out of them, the one which was used in computing the distance function in this study was called 'Principal R-function' denoted as

$R_\alpha(\triangle)$. The principal R-function was expressed as follows:

$$R_\alpha(\triangle) = \frac{1}{1+\alpha}(x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}\ ) \tag{2.3}$$

As stated in [28], previously, out of many systems of R-functions, the $R_1(\triangle)$ system was used in the computation of the $\mathcal{MA}$. By substituting $\alpha = 1$ into equation 2.3 the principal R-function system became the math expression for general Boolean operation[28][15] which is consisted of the following:

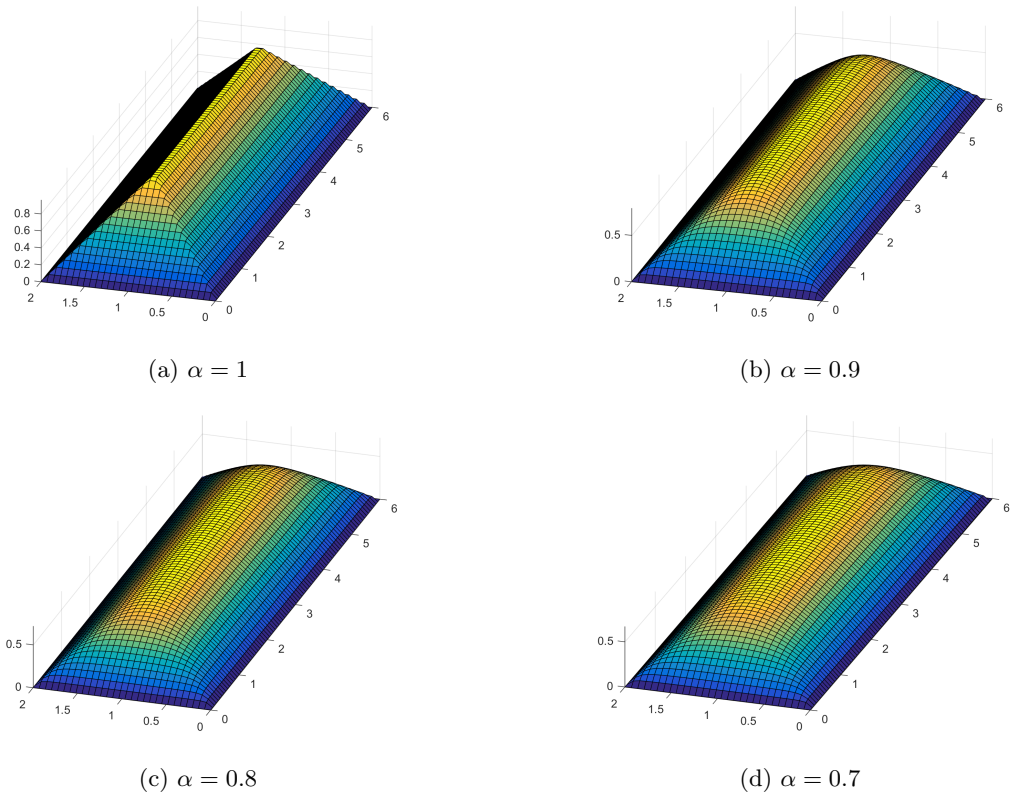
$$\begin{aligned} C &\equiv const && (logical\ 1) \\ \bar{x} &\equiv -x && (logical\ negation\ \neg) \\ x_1 \wedge_1 x_2 &\equiv min(x_1, x_2) && (logical\ conjunction\ \wedge) \\ x_1 \vee_1 x_2 &\equiv max(x_1, x_2) && (logical\ disjunction\ \vee) \end{aligned}$$

To generate the distance function over the domain, the conjunction function was used to combine individual distance functions. The resulting distance function would have local extremum which creates ridges and ravines on the function surface. Because $\mathcal{MA}$ was defined as the center of the domain and has at least two closest boundary point, these ridges and ravines, which were the natural separations of the Voronoi cell created by the individual distance function, would have at least two closet boundary points and the projection of them were then used as the set of $\mathcal{MA}$. An intuitive example had been illustrated in figure 1.7. The ridges and ravines of the distance function are the location of the $\mathcal{MA}$(highlighted in red lines).

As a result, with proper ridges and ravines extraction mechanism, one can easily capture the $\mathcal{MA}$ by locating the ridges and ravines on the distance function. Later on, with the power of R-function system, the computation paradigm can theoretically be extended to compute $\mathcal{MZ}$ of a given shape.

2.3.5 Why R-function

The power of R-function system lies on the fact that it can depart from a simple min or max function. With altering the α such that $0 \leq \alpha < 1$(as shown in figure 2.7), the non-differentiable edge points on the distance function become differentiable, and at the crest point, which originally was the ridge or ravine point, would naturally have the largest local curvature. And with the α value close to 0, the This allows a possible way of constructing the $\mathcal{MZ}$ if (1) a method of determine the local crest point is available. (2) a consistent procedure of capturing desire width of around the local crest point to form a $\mathcal{MZ}$. However, as introduced above, this paradigm can not be fulfilled by the approximated Laplacian(detailed in the **Extracting Ridges and Ravines** part). Preliminary work had shown that it is considerably difficult to obtain a reliable mechanism for computing $\mathcal{MZ}$ from the paradigm illustrated above. Thus, seeking an alternative way of constructing the $\mathcal{MZ}$ was preferred.



(a) $\alpha = 1$ (b) $\alpha = 0.9$

(c) $\alpha = 0.8$ (d) $\alpha = 0.7$

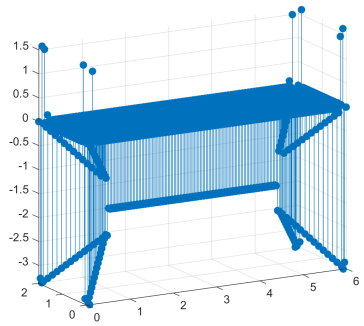
Figure 2.7: Approximated distance function of a rectangular domain computed by using R-function.

2.3.6 Extracting Ridges and Ravines

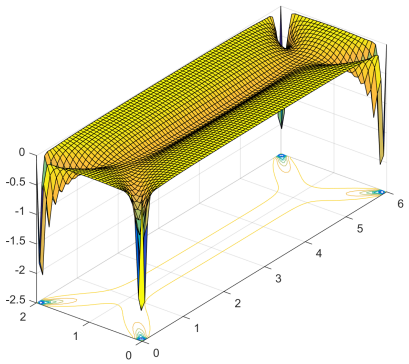
After obtaining the 'desired distance function'[3] over the domain, the problem becomes finding a method to extract the desire ridges and ravine so that they can be projected back as $\mathcal{MA}$(or $\mathcal{MZ}$ when altering α.).

For this study, there needed to be a fast and robust way of extracting $\mathcal{MA}$ from a 3D domain which its distance function is defined in the 4th dimension. Laplacian edge detection algorithm was then chosen to achieve both efficiency and effectiveness. Laplacian edge detection method is a commonly used method in the field of computer vision as a efficient method of capturing large derivative changes in functions or images' intensity. It looks for intensity changes in images which relying on the fact that changes in intensity can be captured by a large first directional derivative and resulting a zero-crossing in the second directional derivative. In this study,although these 4D ridges and ravines can not be visualized, they can be captured by the Laplacian operator applying on the distance function. A finite approximation of the Laplacian was used which was known as discrete Laplacian approximation that can perform a fast computation to construct a laplacian value over the domain based on the distance function. With such function, the place where $\mathcal{MA}$ located would have a greater absolute value compared to other interior locations of the domain so the $\mathcal{MA}$ can be easily extracted by selecting the 'highlighted' points.

[3]this 'desired distance function' can be the exact distance function obtained by having $\alpha = 1$ or an approximate distance function given by altering the α value



(a) Laplacian approximation for $\mathcal{DF}$.



(b) Laplacian approximation for approximate $\mathcal{DF}$ at $\alpha = 0.8$.

Figure 2.8: Discrete Laplacian approximation for $\mathcal{DF}$ as well as approximate $\mathcal{DF}$.

Originally, this Laplacian approximation was used in 2D image processing. In this study, it had been extended into 3D and its power seems to hold for 3D cases as well. The formulation is shown below:

$$\begin{aligned} \bigtriangledown^2 f(x,y,z) &\simeq l_{ijk} \\ &= f_{i+1,j,k} + f_{i-1,j,k} + f_{i,j+1,k} \\ &\quad + f_{i,j-1,k} + f_{i,j,k+1} + f_{i,j,k-1} - 6f_{i,j,k} \end{aligned} \tag{2.4}$$

Figure 2.8 shows a distance function of a rectangle and its laplacian value computed by discrete Laplacian. The $\mathcal{MA}$ is captured by setting the threshold to a small value below 0.

As discussed in previous section, if $0 \leq \alpha < 1$, the resulting differentiable is not the distance function(called approximate distance function) andthe ridges and ravines on the resulting function became differentiable local curvature extrema. Previous work intend to capture the regions near the local extrema of the resulting function in such a way that the geodesic distance from the local extrema can be controlled. However, the Laplacian approximation proposed above did not detect all the features needed to generate effective $\mathcal{MZ}$ family so a new method was needed to compute these points.

2.3.7 Discussion

Previous procedure take advantages of the 'nice' properties provided by the approximate distance function computed by using R-function as well as the relationship between the Laplacian value and the local curvature to construct the PMC test on the Laplacian approximation function over the domain. However, the topological correctness of this method heavily relied on the ability to capture the desired points on the approximate distance function. Although there is an un-denying fact that the crest lines, which indicates the location of $\mathcal{MA}$, has the local maximum curvature and the local maximum curvature would result in a local minimum Laplacian value computed from the approximate distance function. However, as one trace along the $\mathcal{MA}$, the value of this local minimum would also change accordingly.

In the actual implementation, the PMC was designed to select points that has a Laplacian value that is below the threshold. Because of this PMC, there can be cases where an adequate Laplacian threshold that is working for a particular location has a lower value than the local minimum value of another location. If that happens, the resulting $\mathcal{MZ}$ would exclude those points that has a higher Laplacian value even though they should, in principal, be captured. Excluding these points would create holes and disconnected surface patches and as a result, violate homeomorphism. However, in our investigation, we are not aware of a robust way to capture points around the crest lines of the approximate distance function. Instead, a new set of $\mathcal{MZ}$ was defined and a new computational procedure was also proposed. This new set of $\mathcal{MZ}$ shares the same theoretical properties as the previous one, but it can in principal preserve the topological properties while previous method can not.

Chapter 3

Medial Zone computation

3.1 Overall procedure

In this chapter, the actual computation paradigm was illustrated and the mechanism used in computing the valid medial zone($\mathcal{MZ}$) family was explained, not only in the formulation, but also in the actual implementation as well.

Generally speaking, the new paradigm for $\mathcal{MZ}$ computation was proposed as the following figure 3.1. However, to follow this paradigm, there needed to be an effective point membership classification(PMC)[1] method which can overcome the issue of introducing topological artifacts that previous computation had faced. To tackle this problem, a new PMC based on a special type of Gaussian kernel(RBF Gaussian kernel)[2] was introduced. For each point on $\mathcal{MA}$, This new PMC attached a kernel with desired radius with it. With this construction, the boundary of $\mathcal{MZ}$ was formed by the envelope of all the kernels. To begin with, the new PMC was introduced in the next section.

3.2 New PMC based on Gaussian kernel

Gaussian Kernel is constructed based on the well known Gaussian distribution. The Gaussian kernel had been widely used in a lot of the active research areas such as pattern recognition and machine learning providing a simple yet powerful filter for membership classification. For a Gaussian kernel defined on a certain point, only the neighborhood of that point would have a kernel value above zero so that not only the kernel is able to capture the neighborhood points of a query point but also the range of the neighborhood can be controlled by parameters when constructing the kernel. With such kernel, one was able to capture points which were in a desire close range to any given medial axis($\mathcal{MA}$) point.

[1] Point membership classification means a mechanism which can decide, for a geometry, whether a query point is IN, ON or OUT. IN means inside, ON means on the boundary of the geometry and OUT means outside the geometry

[2] RBF stands for Radial Bases Function, it is a 'regularized' form of Gaussian kernel that the kernel is always ranged from 0 to 1

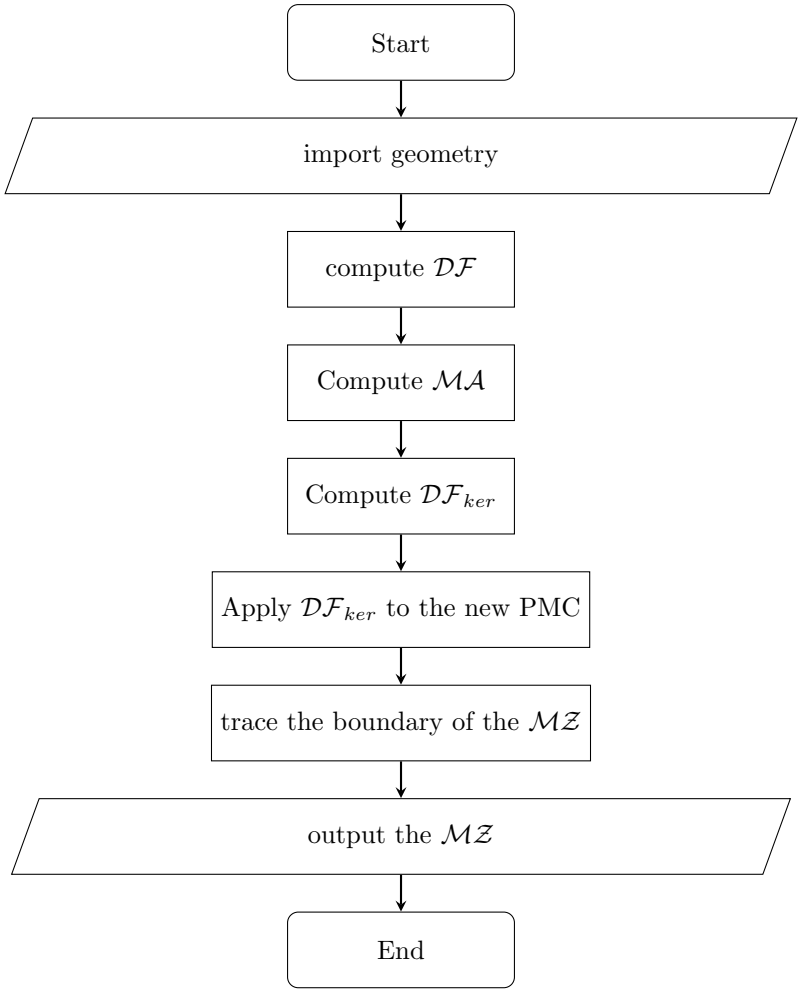


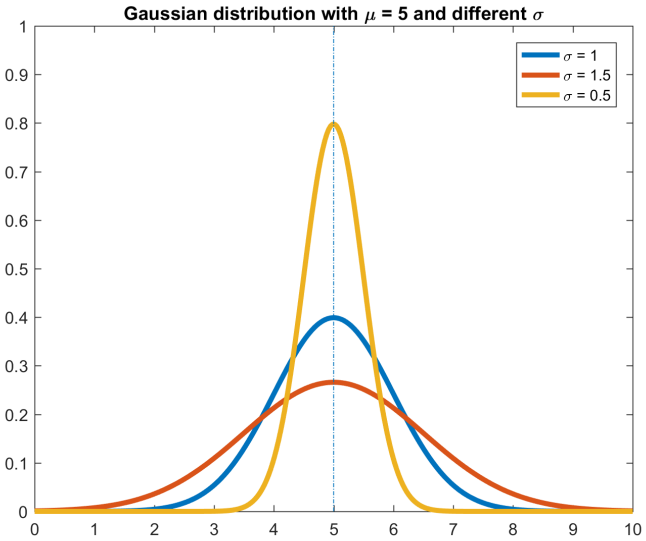
Figure 3.1: General paradigm for Medial Zone computation

Gaussian kernel was chosen in this study because it can be easily formulate to represent a closed ball(circles in 2D, spheres in 3D). the closed balls would inherit the distance information so that the kernels radius can be controlled as well as represent a distance function constructed from the $\mathcal{MA}$ computation. Moreover, there exist known conditions for a smooth boundary recovered by closed balls and distance function, thus choosing Gaussian kernel allows these conditions to be able to applied to this study.

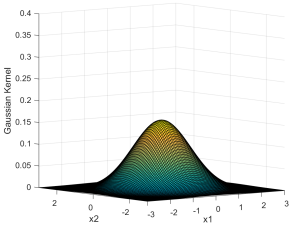
Originally, a 3D Gaussian kernel was defined as:

$$Gaussian(\mathbf{X_1}, \mathbf{X_2}) = \frac{1}{\sqrt{2}^{\frac{3}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} exp(-\frac{1}{2}(\mathbf{X_1} - \mathbf{X_2})^T \mathbf{\Sigma}^{-1}(\mathbf{X_1} - \mathbf{X_2})) \tag{3.1}$$

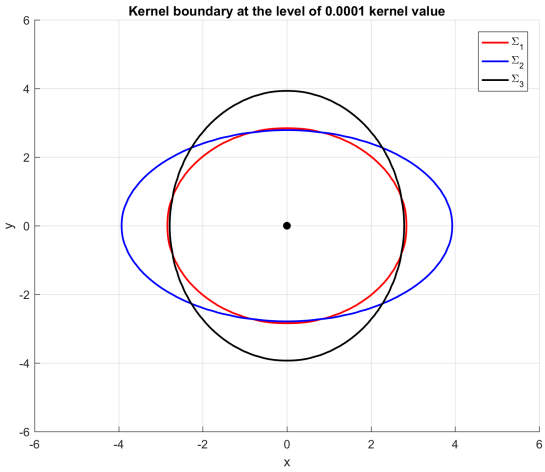
Where the integral of the kernel is 1 and the expand and, flatness and shape of the kernel is controlled by the coefficient matrix Σ. An intuitive example was shown in figure 3.2a, with different σ(a 2D version of the coefficient matrix), the expand of the kernel and the maximum value of it are altered. And also shown in figure 3.2b, the shape of the multivariate Gaussian kernel can be altered from a circle to an ellipse with different coefficient matrix Σ.



(a) 2D Gaussian distribution with different σ



(b) Surface plot of multivariate Gaussian distribution



(c) Multivariate Kernel range with different Σ

Figure 3.2: Gaussian distribution and multivariate Gaussian kernel.

In this study, Gaussian kernel was used to represent the maximum ball(Sphere in 3D) associated with a particular point on $\mathcal{MA}$. A homogeneous sphere(disk in 2D) kernel and the radius was preferred to be controlled by the desire distance function. Besides, the property of having the whole kernel's integral to 1 provided no actual benefits to the $\mathcal{MZ}$ formulation. So the RBF Gaussian Kernel was used in the $\mathcal{MZ}$ construction serving as a PMC tool for capturing the desire neighborhood around the $\mathcal{MA}$ and constructing the $\mathcal{MZ}$.

A RBF Gaussian Kernel means the following simplification:

Firstly, the normalized term was removed. Secondly, the 3x3 $\mathbf{\Sigma}$ matrix was set to represent a sphere.

The benefit of using the RBF lies on the fact that the real value function, which in this case the Gaussian kernel value, solely depends on the relative distance from $\mathbf{X_1}$ and $\mathbf{X_2}$. More specifically, the RBF Gaussian Kernel value is solely determined by a dimensionless matrix Σ. As we are building the

RBF Gaussian kernels for all the $\mathcal{MA}$ points, this set up would provide two major benefits: (1) This set up allows us to use an identical PMC threshold value to extract the valid points inside of all the Kernels. (2) Such identical PMC threshold would always representing a same relative distance to every $\mathcal{MA}$ point. To illustrate this, consider an example shown in figure 3.3. A constant kernel value would only be determined by the relative distance from the center of the kernel. In the example shown, Kernel value of 0.01 indicated 3σ away from the center and 0.606 indicated a σ width kernel.

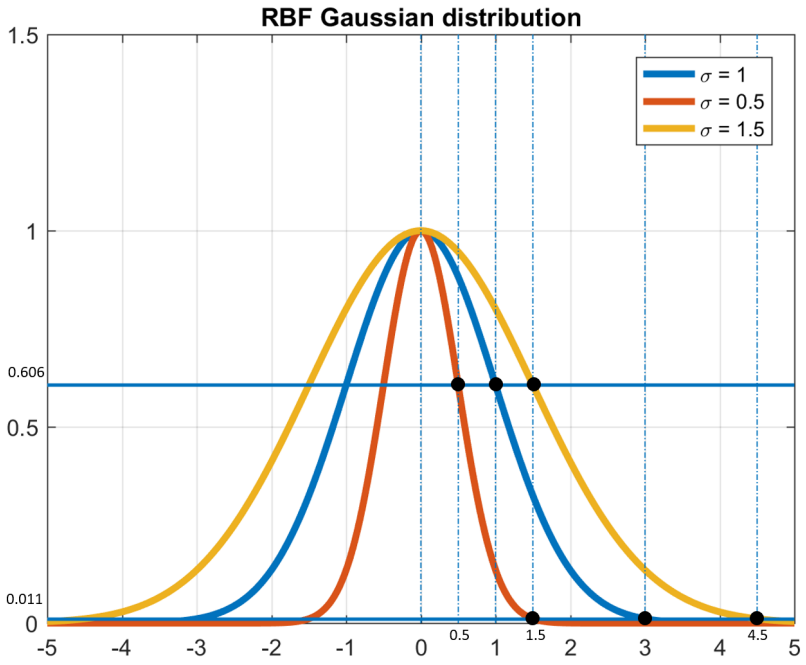


Figure 3.3: the RBF Gaussian kernels with different σ

To formulate this RBF Gaussian Kernel PMC function, we set

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix} \tag{3.2}$$

$$\mathbf{X_1} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \mathbf{X_2} = \begin{pmatrix} x_{ma} \\ y_{ma} \\ z_{ma} \end{pmatrix} = \mathbf{P_{ma}} \tag{3.3}$$

where $\mathbf{P_{ma}}$ is the $\mathcal{MA}$ point.

By substituting equation 3.2 and 3.3 into 3.1, we derived the expression of our RBF Gaussian Kernel

PMC function as follows:

$$Gaussian(\mathbf{P_{ma}}) = exp(\frac{-1((x - x_{ma})^2 + (y - y_{ma})^2 + (z - z_{ma})^2)}{2\sigma^2}) \tag{3.4}$$

This Gaussian kernel provided an important means to relate the distance function with the $\mathcal{MA}$. For each point on $\mathcal{MA}$, there was a kernel with its radius associated with the distance value of that point, the envelope of all the kernels defined by $\mathcal{MA}$ points would become the boundary of $\mathcal{MZ}$.

3.3 Individual kernel's width formulation

Theoretically, the kernel's width was controlled by the covariance matrix Σ. Following the discussion made in previous section, the kernel formulated in this study was controlled only by a single parameter σ. With a threshold value, the kernel was able to serve as a PMC test capturing the points within a desire range of the kernel's center by selecting the points whose kernel value is greater or equal to the threshold. Figure 3.3 illustrated the relationship between threshold values and the corresponding width of the kernel.

It is reasonable to choose a mechanism that the width represents the distance from the $\mathcal{MA}$ to the desire boundary since we intend to use an approximate distance function to control the width of the kernel and the boundary of the resulting $\mathcal{MZ}$ is traced as the envelope of all the kernels. Following this structure, the construction would be as follows:

Let the distance function of the medial points denoted as $\mathcal{DF}$, the formulation of an individual kernels width would be:

$$\sigma = \frac{\mathcal{DF}}{3}. \tag{3.5}$$

The numerator 3 is determined by the fact that all the points within 3σ radius would have a value greater than zero. Since we intent to use 0.01 as the threshold, this formulation would have the desire radius of the kernel equal to the distance function $\mathcal{DF}$.

3.4 Kernel's Width Function

It is important to seek an appropriate function which can control all the kernels in a way such that a complete family of valid $\mathcal{MZ}$ can be generated. Figure 3.4 shows conceptually the valid family of $\mathcal{MZ}$, the function which can generate such $\mathcal{MZ}$ family was what we seek in this study.

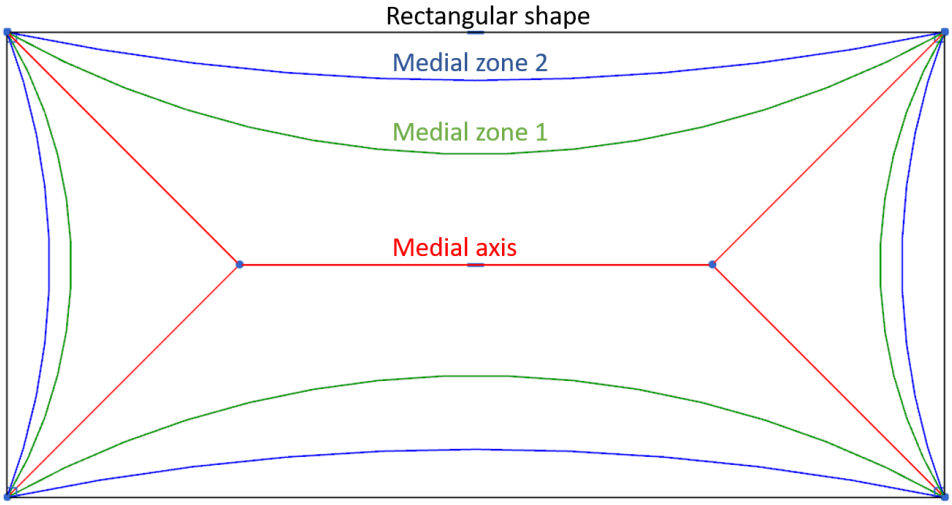


Figure 3.4: $\mathcal{MZ}$ family of a rectangular shape.

3.4.1 Requirements

To qualify this problem, the requirements of a valid $\mathcal{MZ}$ and the concept of complete family should be understood. The requirements of a valid $\mathcal{MZ}$ had been proposed in the properties of $\mathcal{MZ}$ in chapter 2. Despite having quite a few requirements, the most important and essential property was homeomorphic to the domain and homotopy equivalent to the $\mathcal{MA}$. To satisfy these basic requirements, the function should have the following properties;

1. Continuity: the function should be at least continuous in order to generate a continuous $\mathcal{MZ}$ boundary.

2. Bounded function: the function's value for each individual kernel should be bounded between 0 and the maximum allowed width(the distance function value).

Besides, as the goal included to generate a **family** of $\mathcal{MZ}$, there needed to be a mechanism that can produce a corresponding family of width function which satisfies the double convergence property proposed by defining the family of $\mathcal{MZ}$.

3.4.2 Distance Function

Even with the above requirements, there exist infinite many ways of constructing a family of continuous function to control the radius of the kernels that satisfied the requirements. One fundamental formulation of such function was the distance function acquired from the $\mathcal{MA}$ computation. The distance function is absolutely continuous if the boundary of the geometry is continuous and, as stated in the requirements,

it would certainly not exceeding the upper bound of the width function. In order to satisfied the double convergence requirements, a new parameter is introduced as $\gamma \in (0, 1)$. γ is meant to indicates how 'thick' the resulting $\mathcal{MZ}$ should be. $\gamma = 0$ means the kernel would have 0% of the distance function as width(resulting the $\mathcal{MA}$) and $\gamma = 1$ means the kernel's width equaled to the distance function value(resulting the original domain). This formulation allows one to compute the $\mathcal{MZ}$ with different thicknesses in a continuous matter and was stated as the following:

$$\mathcal{DF}_{ker} = \gamma \mathcal{DF} \tag{3.6}$$

Although applying distance function directly was very computational efficient, the resulting $\mathcal{MZ}$ boundary would be most likely piecewise linear. Although the smoothness does not affect homeomorphism or homotopy equivalence, previous work had shown the benefits of having a smooth yet curved $\mathcal{MZ}$ boundary. Thus, in the new formulation, we shall try to preserve the smooth and curved boundary as much as possible. As a result, the distance function alone can not provide such property and a new function needed.

3.4.3 Kernel's width Function Formulation

The kernel's width function used in this study was constructed by combining the exact distance function with a exponential function. To illustrate our formulation, we shall denote the exact distance function of $\mathcal{MA}$ as $\mathcal{DF}$ and the exponential function as $\mathcal{DF}_{exp}$(details would be discussed shortly) and the function used in controlling all the kernels as $\mathcal{DF}_{ker}$.

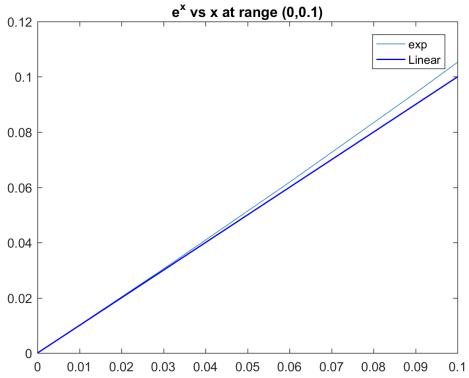
The formulation of $\mathcal{DF}_{ker}$ can be stated as follows:

$$\mathcal{DF}_{ker} = \gamma \left(\gamma^k \mathcal{DF} + (1 - \gamma)^k \mathcal{DF}_{exp} \right), \qquad k \geq 2 \tag{3.7}$$

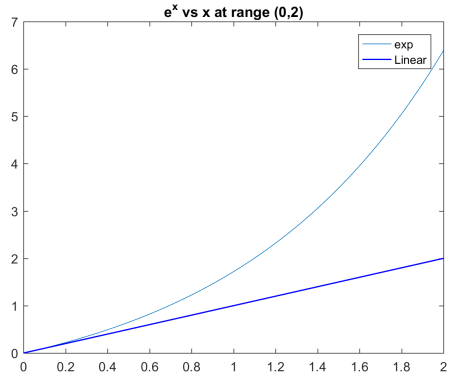
where γ is the same as the one used in controlling the individual kernel, and k is a user specified integer greater than 1(2 by default). In equation 3.7, the γ outside the parentheses served the same purpose as in equation 3.6 and the γ inside the parentheses controlled the interpolation of the exact distance function $\mathcal{DF}$ and the exponential function $\mathcal{DF}_{exp}$ so that when $\gamma \to 0$, $\mathcal{DF}_{ker} \to \mathcal{DF}_{exp}$ and when $\gamma \to 1$, $\mathcal{DF}_{ker} \to \mathcal{DF}$.

Besides the distance function $\mathcal{DF}$, the exponential function $\mathcal{DF}_{exp}$ also played an important role in constructing the width function for the kernels. Originally, the exponential function was constructed by substitute the exact distance function into an exponential function shown below:

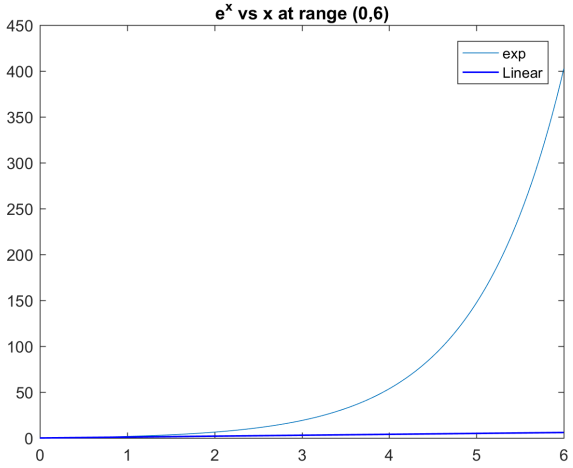
$$\mathcal{DF}_{exp,pre} = exp(\mathcal{DF}) - 1. \tag{3.8}$$



(a) $x \in (0, 0.1)$



(b) $x \in (0, 2)$



(c) $x \in (0, 6)$

Figure 3.5: e^x vs x under different scales.

Because the exponential function is able to change the linear distance function into a smooth function with a monotonously increases gradient, this exponential function can produce a curved $\mathcal{MZ}$ boundary along a $\mathcal{MA}$ branch. However, this formulation was very sensitive to the actual value of the distance function. As can be seen in figure 3.5, with different maximum value of the distance function, the resulting functions have great differences. The difference between the exponential function and its original distance function was so huge that it can not be used for formulating the kernel's width function. Thus, a normalized exponential function was put forward so that the exponential function would always behave as shown in figure 3.6 where the normalized function only touched the distance function at global extrema value point. The formulation was stated as the following:

$$\mathcal{DF}_{exp} = \beta_2(exp(\beta_1\mathcal{DF}) - 1) = \beta_2 exp(\beta_1\mathcal{DF}) - \beta_2 \tag{3.9}$$

where β_1, β_2 are two scaling factors.

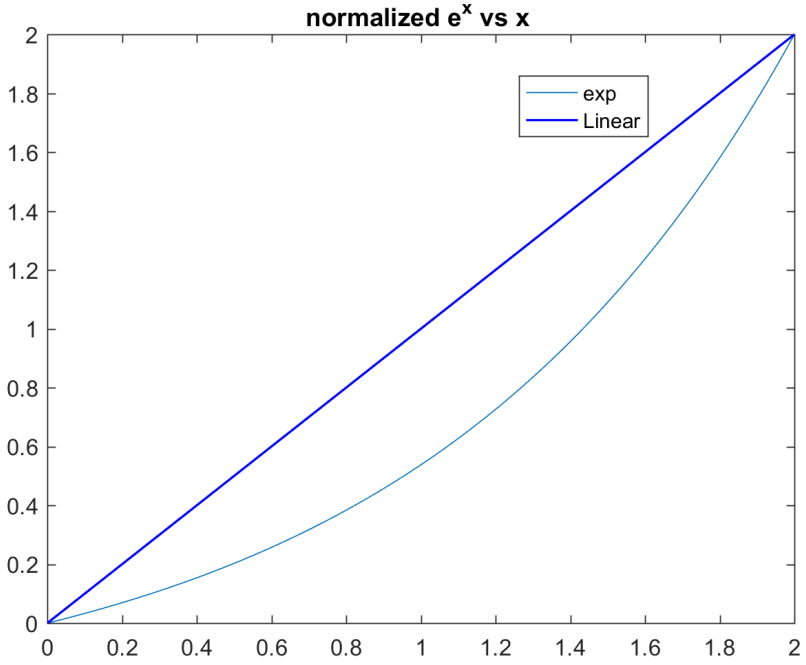


Figure 3.6: The normalized exponential function vs the distance function

The two scaling factors: β_1 and β_2 are used to normalized the exponential function. Specifically, β_1 is set to rescale the maximum value of the original distance function to 2(because in the actual experiment, the shape provided by this rescaling works reasonable well) and β_2 is set to rescale the maximum value of the resulting exponential function to match the maximum value of original distance function.

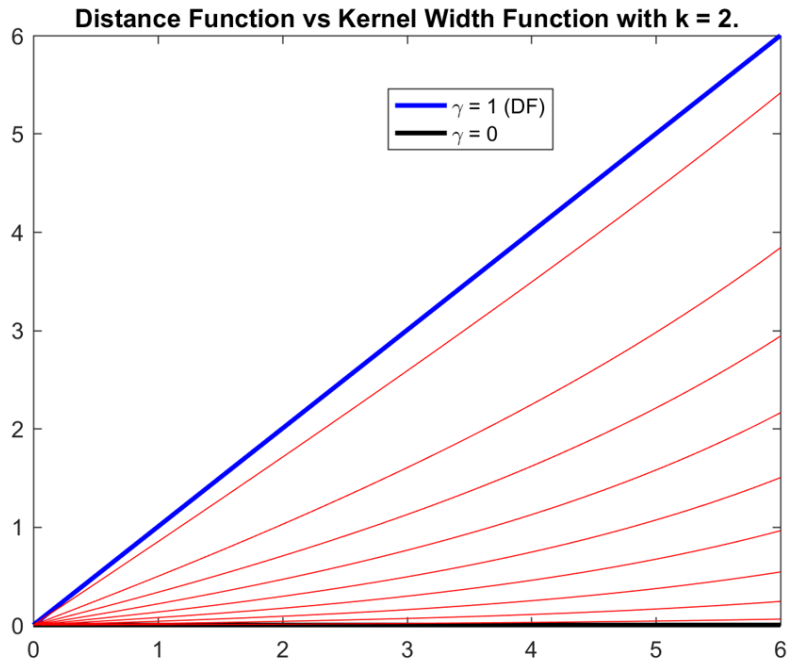


Figure 3.7: Kernel's width function vs distance function.

3.4.4 Properties of $\mathcal{DF}_{ker}$

As proposed in equation 3.7, the Kernel's width function is able to meet the requirements listed in section 3.4.1. First of all, this constructed function is continuous on $\mathcal{MA}$ branches as long as the distance function of them are continuous. Also, the constructed width function is bounded between zero and the maximum allowed distance value in fact, as shown in figure 3.7 , the only situation that the function's value reaches the maximum allowed value would be either on global extrema or when $\gamma = 1$. As γ increased from 0 to 1, the resulting $\mathcal{DF}_{ker}$ converged to the distance function in a continuous manner.

As for maintaining the differential properties of the $\mathcal{MZ}$ boundary, there exist several issues which are hard to take care of and there haven't seen a robust and efficient way to avoid it.

The major issue was the occurrence of singularity points either on a Regular point or a non-differentiable point on $\mathcal{MA}$. The steps that one has to make to get rid of the singularity points on the $\mathcal{MZ}$ boundary are different according to both Giblin's and Damon's work. As a result, the 'treatments' for both situations were discussed in a separate manner.

For imposing a differentiable boundary on regular $\mathcal{MA}$ points(M_{reg}), as illustrated in chapter 2, certain criteria should be sufficed. Giblin and Kimia, who worked closely on reconstructing boundary using envelopes, provided only the criteria for 2D cases. However, the main idea of the work stated that the growth of the 'distance function'(in our case the kernel's width function) needed to be limited not only according to the relative arc-length but also the local curvature. Although there exist no proof that supports Giblin's criteria in 3D cases, the experimental results showed that it still worked reasonably

well in 3D if the gradient direction of the kernel's width function satisfies the criteria.

For a non-differentiable point of $\mathcal{MA}(M_{sing})$ current formulation of the kernel's width function is not able to impose a differentiable boundary in its neighborhood due to the fact that it is hard and inefficient to construct a width function that impose such differentiability around M_{sing}. However, Damon's conditions(listed in chapter 2 Damon's work) can be used to check the boundary smoothness at these singularity points. With these conditions one can adjust the function locally at the non-differential points to achieve the desire differentiable boundary.

3.5 Implementation

The pseudo code is provided below in three sections illustrating the implementation: the overall process(so called Medial Zone Computations) and two separate illustrations of constructing the mixed kernel width function and the tracing of kernels envelope.

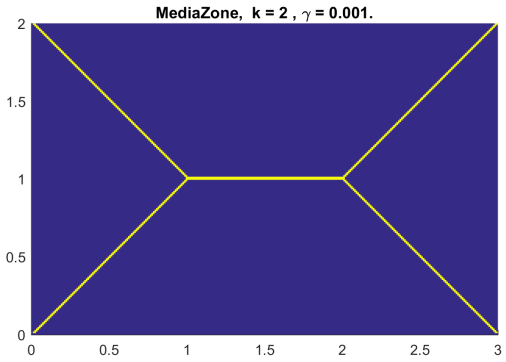
Algorithm 1 Overall Process

procedure MEDIAL ZONE COMPUTATION(detail of * and ** steps can be found in Alg. 2 and 3)
  input: Ω the Geometry model in Parasolid file
  Output: $\mathcal{MZ}(\Omega)$
  Process the imported geometry, generate grid and construct distance functions of the geometry and give each grid point the distance function value.
  Compute $\mathcal{MAMA}(\Omega)$ by using the laplacian approximation and select points which has a non zero value.
  Construct $\mathcal{DF}_{ker}$ the Kernel's width function for $\mathcal{MA}(\Omega)^*$
  for $i = 0$ to $sizeof(\Omega)$ **do**
    $list(P_i) \leftarrow$ the list of cloest neighbor points of set $\mathcal{MA}(\Omega)^{**}$
    $kernelval = max(Gausian(P_i, list(P_i)))$
    if $kernelval > threshold$ **then**
      $\mathcal{MZ}(\Omega).push(P_i)$

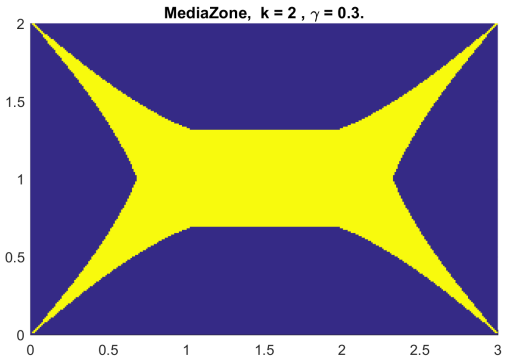
Algorithm 2 *Mixed Approximate Distance Computation

procedure $\mathcal{DF}_{ker}$ COMPUTATION
  input: $\mathcal{DF}$ the distance function, γ and k as control parameters for $\mathcal{DF}_{ker}$ calculation
  Output:$\mathcal{DF}_{ker}$ the kernel's width function
  for $i = 0$ to $sizeof(\mathcal{MA}(\Omega))$ **do**
    $m_i = \gamma^k \mathcal{DF}|_i + (1-\gamma)^k \mathcal{DF}_{exp}|_i$
    $P_{ma,i} = [x_i, y_i, z_i, m_i]$
    $\mathcal{DF}_{ker}|_i = P_{ma,i}$

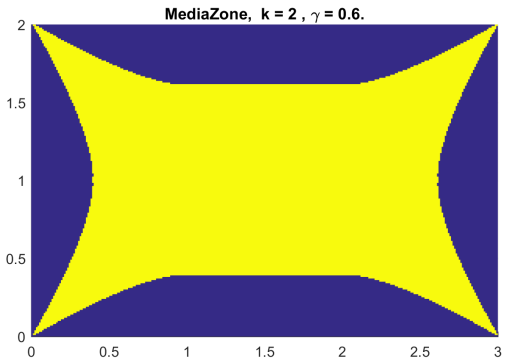
With the implementation shown, a 2D example of a rectangular with dimension 3×2 was presented to examine the paradigm proposed. From figure 3.8a to figure 3.8f, the $\mathcal{MZ}$ with respect to the desire expand ratio γ were represented by the yellow region. This new method showed the ability of constructing the homomorphic $\mathcal{MZ}$ family.



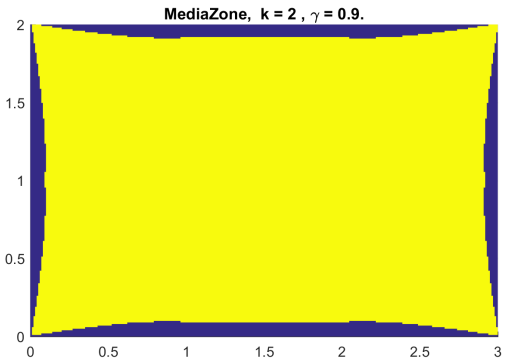
(a) Medial axis, $\gamma = 0.001$.



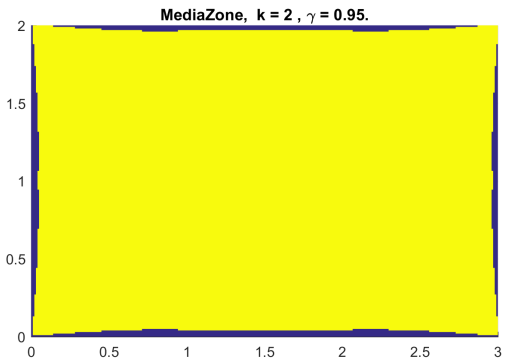
(b) Medial zone 1, $\gamma = 0.3$.



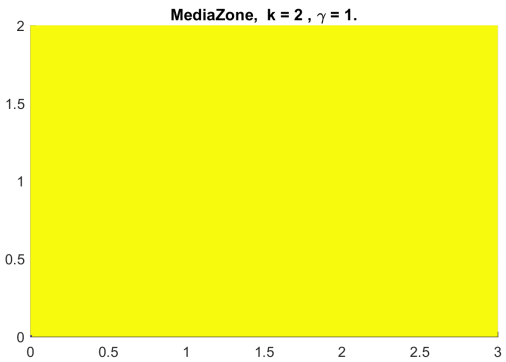
(c) Medial zone 2, $\gamma = 0.6$.



(d) Medial zone 3, $\gamma = 0.9$.



(e) Medial zone 4, $\gamma = 0.95$.



(f) Whole domain, $\gamma = 1$.

Figure 3.8: 2D example of $\mathcal{MZ}$ computed by new method.

Algorithm 3 **Closet Neighbor Search and Gaussian Kernel

1: **procedure** CLOSET NEIGHBOR SEARCH
2: input: Query point P_i, total grid points n.
3: output: $list(P_i)$ the list of nearest neighbors.
4: Build search tree $tree1$ structure on $\mathcal{MA}$ set $\mathcal{MA}(\Omega)$
5: $k = int(0.02\%n)$
6: $list(P_i) = tree1 \rightarrow annKsearch(P_i, k)$

1: **procedure** GAUSSIAN KERNEL
2: input: parameter γ(same as γ in $\mathcal{DF}_{ker}$ computation)query point P, kernel point P_{ma} and $\mathcal{DF}_{ker}$ value for the corresponding kernel point.
3: output:$Gaussian(\mathbf{P}_{ma})$ at $\mathbf{P}$
4: $Gaussian(\mathbf{P}_{ma}) = exp(\frac{-1((x-x_{ma})^2+(y-y_{ma})^2+(z-z_{ma})^2)}{2\sigma^2})|_{\mathbf{P}}$

3.6 Discussions on implementation

The $\mathcal{MZ}$ region computed in the 2D example shown above was a combined results of the new Gaussian kernel PMC and the specific Kernel's radius function $\mathcal{DF}_{ker}$ introduced in this chapter. Because there exist no 'cookbook' procedure for constructing a valid family of $\mathcal{MZ}$, the choice made in this study was just one possibility out of infinite many possible choices.

The essence of this proposed computational method is that the new PMC based on Gaussian kernel can, in principal, produce medial zones with no violations on homeomorphism. In the actual computation, it is up to the users to decide what $\mathcal{DF}_{ker}$ suits best for the specific application. For future reference, a list of all the parameters controlling the final shape of $\mathcal{MZ}$ under the new PMC as well as current construction of $\mathcal{DF}_{ker}$ were provided below.

- α(default value was set to 1) value of R_α system used in constructing $\mathcal{DF}$ and computing $\mathcal{MA}$;
- σ formulation in equation 3.5;
- $\mathcal{DF}_{exp,ma}$ in equation 3.9;
- k value in equation 3.7;
- γ value in equation 3.7

The parameter γ was used to control the 'thickness' of $\mathcal{MZ}$ in range $(0, 1)$, where 0 resulting a $\mathcal{MA}$ and 1 resulting the original shape.

α was used to control the outcome of combining individual distance function defined by individual boundary pieces of the given shape. Although it was true that one can choose an value between 0 to 1 for α to vary the out-coming $\mathcal{DF}$ to approximate $\mathcal{DF}$, however, for the paradigm proposed by this study, the α value is preferred to be 1 in order to perform a fast evaluation of the $\mathcal{DF}$ of the imported geometry and correctly compute the set of $\mathcal{MA}$ points.

σ formulation was chosen so that the threshold can be set to 0.01. In fact, this can be varied to any value between 0 to 1 as long as a fixed relationship between the threshold value and the relative

distance measured by σ was established. For example, 0.6065 can be set to the threshold value as long as $\sigma|_P = \mathcal{DF}_{ker}|_P$.

$\mathcal{DF}_{exp,ma}$ served as one important part in constructing the kernel's radius function. The natural property of an exponential function had contributed in shaping the out-coming boundary of $\mathcal{MZ}$.

k value controlled the interpolation of $\mathcal{DF}$ and $\mathcal{DF}_{exp}$ and a comprehensive example shown in figure illustrated the influence of k in the interpolation.

Out of these parameters listed above, only γ was not determined before the actual computation of one specific $\mathcal{MZ}$. All the other parameters, although being fixed, can actually be varied to obtain a different family of $\mathcal{MZ}$.

Chapter 4

Experimental Results

The implementation was tested on various of geometries from as simple as a convex polyhedron to as complex as a geometry consists of planar and non-planar boundary faces. Throughout the test cases, no violation on homeomorphism as well as homotopy equivalence were detected.

In these examples, we have used the implementation on a workstation with an Intel Xeon 2.3GHZ computer (dual 10 cores) with 64G RAM. The implementation, which was originally written in a 32bit platform, have been moved to a 64bit platform to make fully use of the RAM. Thus, the implementation can afford much denser grid which would enhance the ability of capturing small boundary features in a complex shape and compute a 'correct' medial zone($\mathcal{MZ}$) that can preserve homeomorphism we seek.

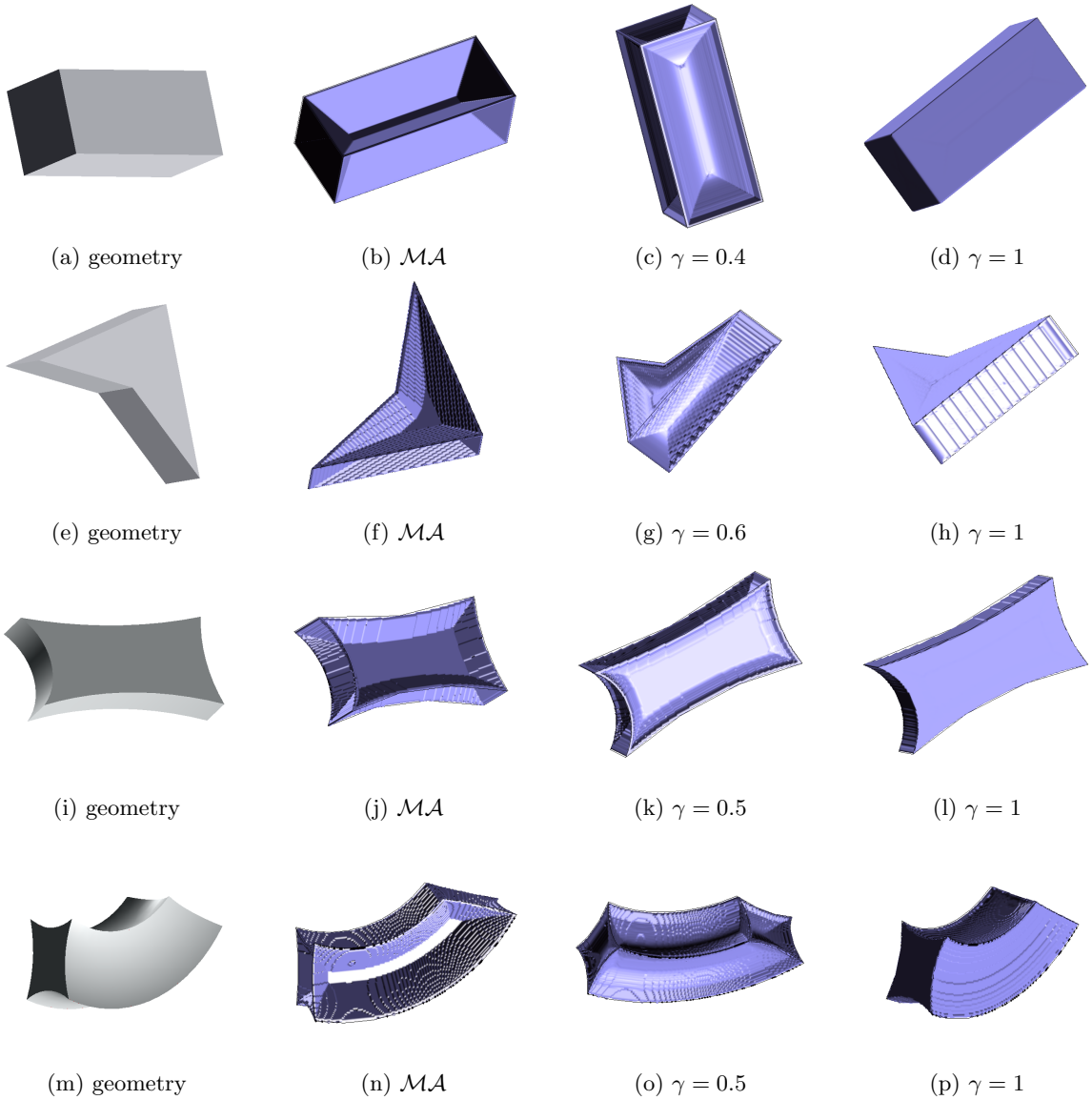
4.1 Examples

Ideally, the $\mathcal{MZ}$ of the geometries show be presented as 'growing' from the medial axis($\mathcal{MA}$) to the original domain in a continuous manner, but it is impossible to do so in this study. As a result, the results are presented in the following manner: original geometry, $\gamma = 0$, $0 < \gamma < 1$ and $\gamma = 1$. For simple geometries shown in figure 4.1, the width function is regulated so that the gradient direction of Kernel's width function would always satisfy the condition described in equation 2.1. Although there exist no proof on the validness of applying this condition in 3D cases, the results shows that the restriction put on the kernel's width function works considerably well with obtaining smooth curved boundary everywhere except the singularity points on the $\mathcal{MA}$ as discussed in chapter 3. Most importantly, homeomorphism was able to be preserved during the computation of all the members of the $\mathcal{MZ}$ family. The computation of complex shapes were conducted by using the same restrictions tested in simple cases. From the results shown in figure 4.2, the regulated kernel's width formulation works just as expected.

The validness of the implementation was first examined in simple models such as block, concave polyhedral and non-polyhedral shapes(figure 4.1). Then it was examined by more complex industrial

shapes such as gears and gasket(figure 4.2) as well as geometric shapes(4.3) which can be arbitrary non-polyhedral with holes. Throughout the complex shapes that were tested, the results show the ability of the implementation to preserve homeomorphism and boundary continuity. In the shown examples, the faces of the $\mathcal{MZ}$ are bumpy because of the rendering method used as well as the grid size chosen.

Besides providing a valid family of $\mathcal{MZ}$, the examples shown on non-polyhedral cases illustrated one of the contributions that this proposed method is not restricted to compute $\mathcal{MZ}$ for polyhedral. Theoretically, as long as a valid distance function and a set of $\mathcal{MZ}$ points can be obtained from the geometry, this implementation is able to provide a valid family of $\mathcal{MZ}$ of any imported shapes.



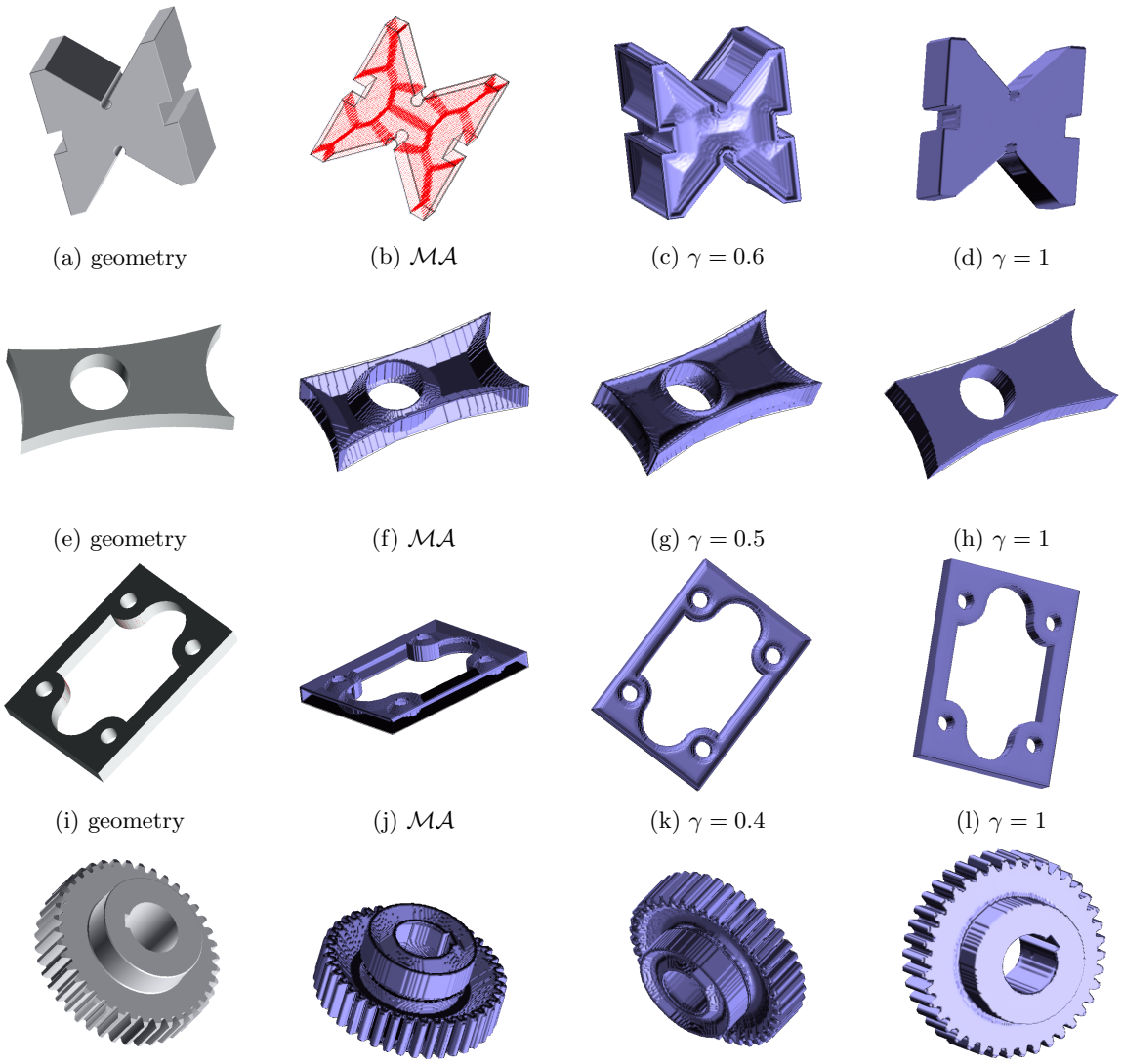
(a) geometry (b) $\mathcal{MA}$ (c) $\gamma = 0.4$ (d) $\gamma = 1$

(e) geometry (f) $\mathcal{MA}$ (g) $\gamma = 0.6$ (h) $\gamma = 1$

(i) geometry (j) $\mathcal{MA}$ (k) $\gamma = 0.5$ (l) $\gamma = 1$

(m) geometry (n) $\mathcal{MA}$ (o) $\gamma = 0.5$ (p) $\gamma = 1$

Figure 4.1: Experiments tests on block, concave polygon and non-polygonal geometries



(a) geometry (b) $\mathcal{MA}$ (c) $\gamma = 0.6$ (d) $\gamma = 1$

(e) geometry (f) $\mathcal{MA}$ (g) $\gamma = 0.5$ (h) $\gamma = 1$

(i) geometry (j) $\mathcal{MA}$ (k) $\gamma = 0.4$ (l) $\gamma = 1$

(m) geometry (n) $\mathcal{MA}$ (o) $\gamma = 0.4$ (p) $\gamma = 1$

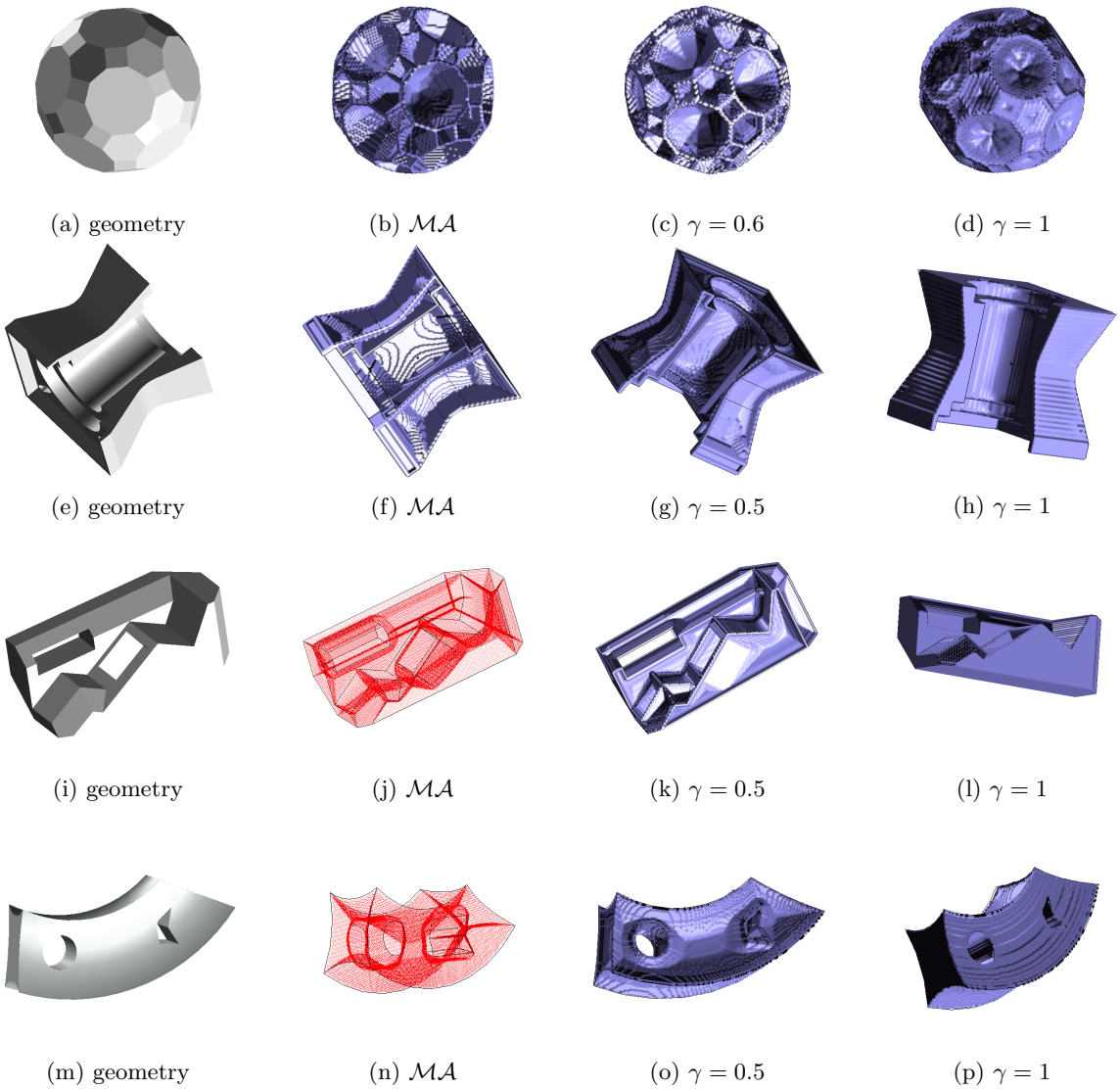
Figure 4.2: Results of complex geometries of polyhedral and non-polyhedral geometries.

4.2 Observations

Despite being able to preserve crucial properties, there exist some noticeable limitations of the proposed method. As can be seen throughout figure 4.1, 4.2 and 4.3, The resulting $\mathcal{MZ}$ has non-differentiable point corresponding to a smooth piece of the boundary. Also, there exist geometric artifacts in figure 4.1g, 4.2j and 4.2o. Besides, there had been detected a convergence issue shown in figure 4.3d that the $\mathcal{MZ}$ failed to converge to the original domain.

Through observations, these were mainly geometric artifacts and were introduced by generally 2 reasons:(1) the specific choice of the $\mathcal{DF}_{ker}$ used in the computation. (2) the limitation of current implementation.

As had been discussed in chapter 2. $\mathcal{DF}_{ker}$ should satisfy Damon's conditions locally To obtain a locally smooth boundary on the computed $\mathcal{MZ}$, However, the specific $\mathcal{DF}_{ker}$ that was chosen in this



(a) geometry (b) $\mathcal{MA}$ (c) $\gamma = 0.6$ (d) $\gamma = 1$

(e) geometry (f) $\mathcal{MA}$ (g) $\gamma = 0.5$ (h) $\gamma = 1$

(i) geometry (j) $\mathcal{MA}$ (k) $\gamma = 0.5$ (l) $\gamma = 1$

(m) geometry (n) $\mathcal{MA}$ (o) $\gamma = 0.5$ (p) $\gamma = 1$

Figure 4.3: More examples on complex geometries

study did not guarantee to satisfy Damon's conditions for each and every point. Therefore, at locations where Damon's conditions failed, would introduce non-differentiable on the $\mathcal{MZ}$ boundary. Since the efforts of this study was about delivering a new computational method that do not introduce topological artifacts, correcting these geometrical artifacts were beyond the scope of this study. However, in principal, one can construct a $\mathcal{DF}_{ker}$ that satisfies Damon's condition at the location of interest to get rid of the non-differentiable points. Or, one can also apply Damon's condition on current $\mathcal{DF}_{ker}$ and adjust the condition-violating points of $\mathcal{DF}_{ker}$ locally to achieve desired smoothness.

Current implementation can, in principal, produce a valid $\mathcal{MZ}$ family for any given shapes which can provide a proper distance function by Parasolid. However, in current implementation, There exist two limitations that would introduce artifact potentially, one was the use of ANN library[1], the other one

[1] Approximated Nearest Neighborhood Search Library

was related using Parasolid as an external library.

Although theoretically, the resulting $\mathcal{MZ}$ boundary was achieved by tracing the envelope of all the kernels constructed by the $\mathcal{MA}$, it was not efficient, or sometimes practical depending on the number of grid points, to compute the kernel value for each and every kernel to decide whether a point is in range of any one of the kernels in practical procedure. To tackle this issue, An external library ANN was used to limit the kernel value computation of the $\mathcal{MA}$ point that is geometrically near the query point. In most of the cases, the $\mathcal{MA}$ point that gives the largest kernel value would usually geometrically near the query point. However, there exist certain cases where the center of the largest kernel containing the query point, in principal, is not in the close neighborhood. Thus for these cases, the appropriate corresponding $\mathcal{MA}$ point can not be obtained by ANN. An example was shown in figure 4.3d, where the $\mathcal{MZ}$ of a soccer ball like shape can not converge to the entire domain because the point which would provide the largest kernel value for points in the neighborhood of each faces' center was given by the center point of the geometry. However, in the computational process, the center point is excluded on the list of nearest neighbors which resulting the issue shown.

The Parasolid library used in current implementations did not return a correct distance function of the cylindrical surface as shown in figure 4.2j and 4.2o. The inadequate distance function returned by the library had caused the implementation to return inadequate $\mathcal{MA}$ points which led to the incorrect $\mathcal{MZ}$ for cylindrical surfaces shown. The cause of this issue had not been fully understand due to the lack of accessibility of Parasolid library.

Chapter 5

Summary

It is an un-denying fact that medial axis($\mathcal{MA}$) and medial zone($\mathcal{MZ}$) has great potential in solving motion planning, path planning and geometric reasoning related problems. However, previous implementation of $\mathcal{MZ}$ computation causes topological artifacts which violates homeomorphism on the resulting $\mathcal{MZ}$. The issue is believed to be caused by a combination of the over-simple PMC proposed by previous method and the poor Laplacian Approximation used in the actual implementation. As a result, the PMC method chosen in previous method failed to appropriately capture the desire points for constructing a valid $\mathcal{MZ}$ family.

In this study, we provided an improved method for computing a valid $\mathcal{MZ}$ family of any geometry which its distance function can be obtained by using Parasolid. This method replaced previous Laplcian approximation PMC with a Gaussian Kernel based PMC. Besides, a kernel's radius function, which was constructed by a non-linear interpolation of distance function and a adjusted exponential function(see equation 3.7), is also proposed to serve the purpose of controlling all the individual kernels in a way such that the boundary of the $\mathcal{MZ}$ can be obtained by tracing the envelope of all the kernels whose boundary is represented by points that have the same kernel value. The new method was tested on both 2D and 3D cases as well as polyhedral and non-polyhedral shapes. The results supports our claim as expected.

This proposed method trade topological correctness for computation efficiency. Although introduces several geometrical artifact, this method can in principal prevent topological artifacts and provide an intuitive thickness control to compute a family of topologically correct $\mathcal{MZ}$.

Besides, as a follow-up work from previous project, this study investigated into the boundary of the computed $\mathcal{MZ}$ such as boundary continuity and geometrical artifacts. Existing known conditions of imposing a geometrically artifacts-free surface were also reviewed and these conditions can be applied to current $\mathcal{DF}_{ker}$ constructions to correct the artifacts.

Bibliography

[1] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, 2001.

[2] N. Aspert, D. Santa Cruz, and T. Ebrahimi. Mesh: measuring errors between surfaces using the hausdorff distance., 2002.

[3] D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes-a state-of-the-art report, 2009.

[4] H. Blum. A transformation for extracting new descriptions of shape, symp. models for the perception of speech and visual form, 1964.

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

[6] J. W. Bruce and P. J. Giblin. *Curves and Singularities: a geometrical introduction to singularity theory*. Cambridge university press, 1992.

[7] T. Culver, J. Keyser, and D. Manocha. Accurate computation of the medial axis of a polyhedron, 1999.

[8] J. Damon. Smoothness and geometry of boundaries associated to skeletal structures i: Sufficient conditions for smoothness, 2003.

[9] J. Damon. Smoothness and geometry of boundaries associated to skeletal structures, ii: Geometry in the blum case. *Compositio Mathematica*, 140(06):1657–1674, 2004.

[10] J. Damon. Determining the geometry of boundaries of objects from medial data. *International Journal of Computer Vision*, 63(1):45–64, 2005.

[11] J. Damon. The global medial structure of regions in r3. *Geometry & Topology*, 10(4):2385–2429, 2006.

[12] J. Damon. Global geometry of regions and boundaries via skeletal and medial integrals. *Communications in Analysis and Geometry*, 15(2):307, 2007.

[13] J. Damon. *Geometry and medial structure*. Springer, 2008.

[14] A. A. Eftekharian and H. T. Ilieş. Distance functions and skeletal representations of rigid and non-rigid planar shapes. *Computer-Aided Design*, 41(12):865–876, 2009.

[15] A. A. Eftekharian and H. T. Ilieş. Medial zones: Formulation and applications. *Computer-Aided Design*, 44(5):413–423, 2012.

[16] M. Foskey, M. C. Lin, and D. Manocha. Efficient computation of a simplified medial axis. *Journal of Computing and Information Science in Engineering*, 3(4):274–284, 2003.

[17] P. Giblin and B. 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.

[18] P. J. Giblin and B. B. Kimia. On the intrinsic reconstruction of shape from its symmetries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):895–911, 2003.

[19] P. J. Giblin and B. B. Kimia. *Local forms and transitions of the medial axis*. Springer, 2008.

[20] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in prm planners, 2000.

[21] L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 14(9):869–885, 1992.

[22] F. F. Leymarie and B. B. Kimia. The shock scaffold for representing 3d shape, 2001.

[23] N. J. Naccache and R. Shinghal. Spta: A proposed algorithm for thinning binary patterns. *IEEE Transactions on Systems, Man, & Cybernetics*, 1984.

[24] S. M. Pizer, K. Siddiqi, G. Szekely, J. N. Damon, and S. W. Zucker. Multiscale medial loci and their properties. *International Journal of Computer Vision*, 55(2-3):155–179, 2003.

[25] W. R. Quadros, B. Gurumoorthy, K. Ramaswami, and F. B. Prinz. Skeletons for representation and reasoning in engineering applications. *Engineering with Computers*, 17(2):186–198, 2001.

[26] J. M. Reddy and G. M. Turkiyyah. Computation of 3d skeletons using a generalized delaunay triangulation technique. *Computer-Aided Design*, 27(9):677–694, 1995.

[27] M. Samozino, M. Alexa, P. Alliez, and M. Yvinec. Reconstruction with voronoi centered radial basis functions, 2006.

[28] V. Shapiro. Semi-analytic geometry with r-functions. *ACTA numerica*, 16:239, 2007.

[29] D. J. Sheehy, C. G. Armstrong, and D. J. Robinson. Computing the medial surface of a solid from a domain delaunay triangulation, 1995.

[30] D. W. Storti, G. M. Turkiyyah, M. A. Ganter, C. T. Lim, and D. M. Stal. Skeleton-based modeling operations on solids, 1997.

[31] H. Tamura. A comparison of line thinning algorithms from digital geometry viewpoint, 1978.

[32] G. M. Turkiyyah, D. W. Storti, M. Ganter, H. Chen, and M. Vimawala. An accelerated triangulation method for computing the skeletons of free-form solid models. *Computer-Aided Design*, 29(1):5–19, 1997.

[33] M. S. Vimawala and G. M. Turkiyyah. Computational procedures for topological shape design. *Computer methods in applied mechanics and engineering*, 125(1):257–285, 1995.

[34] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space, 1999.

[35] H. Xia and P. G. Tucker. Finite volume distance field and its application to medial axis transforms. *International journal for numerical methods in engineering*, 82(1):114, 2010.

[36] Y. Xia. Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes. *IEEE transactions on pattern analysis and machine intelligence*, 11(10):1076–1086, 1989.

[37] Y. Yang, O. Brock, and R. N. Moll. Efficient and robust computation of an approximated medial axis, 2004.